HAL
open science

Adaptive Wisp Tree - a multiresolution control structure for simulating dynamic clustering in hair motion

Florence Bertails, Tae-Yong Kim, Marie-Paule Cani, Ulrich Neumann

To cite this version:

Florence Bertails, Tae-Yong Kim, Marie-Paule Cani, Ulrich Neumann. Adaptive Wisp Tree - a multiresolution control structure for simulating dynamic clustering in hair motion. Symposium on Computer Animation, Jul 2003, San Diego, United States. inria-00519007

HAL Id: inria-00519007

https://inria.hal.science/inria-00519007v1

Submitted on 17 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Eurographics/SIGGRAPH Symposium on Computer Animation (2003)
D. Breen, M. Lin (Editors)

Adaptive Wisp Tree - a multiresolution control structure for simulating dynamic clustering in hair motion

F. Bertails[1], T-Y. Kim[2†], M-P. Cani[1] and U. Neumann[2]

[1] GRAVIR-IMAG, joint lab of CNRS, INRIA, INPG and UJF
[2] Integrated Media System Center, University of Southern California

Abstract

Realistic animation of long human hair is difficult due to the number of hair strands and to the complexity of their interactions. Existing methods remain limited to smooth, uniform, and relatively simple hair motion. We present a powerful adaptive approach to modeling dynamic clustering behavior that characterizes complex long-hair motion. The Adaptive Wisp Tree (AWT) is a novel control structure that approximates the large-scale coherent motion of hair clusters as well as small-scaled variation of individual hair strands. The AWT also aids computation efficiency by identifying regions where visible hair motions are likely to occur. The AWT is coupled with a multiresolution geometry used to define the initial hair model. This combined system produces stable animations that exhibit the natural effects of clustering and mutual hair interaction. Our results show that the method is applicable to a wide variety of hair styles.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

Figure 1: *Comparisons between real hair and simulation results.*

1. Introduction

Despite recent advances in the animation of virtual humans, simulating the complex motion of long hair remains an area of active research. In contrast to short hair and fur that has been successfully used in feature films such as Stuart Little [2], Monsters Inc. [9], simulating reasonably long, human-like hair poses additional challenges since both hair interaction with the character's body and mutual hair interaction need to be modeled. Without these interactions, the resulting hair animation would lose realism.

This paper describes an effective adaptive method that models the dynamic clustering behavior often observed in long hair motion (see Figure 1). To model such splitting and merging behavior, we employ multiresolution representations for both modeling (geometric level of detail) and animation (control level of detail), where the latter is extracted from the former. The adaptive wisp tree (AWT), a control structure built upon the multiresolution model, approximates the large-scale coherent motion of clusters as well as small-scale variations in hair motion. During animation, the AWT automatically adapts itself based on local wisp fusion and separation. It also helps to focus computations where visible hair motions are likely to occur. The time-varying AWT segments control the motion of underlying multiresolution hair model, including the individual hair strands used at the rendering stage. The resulting animations are efficiently computed and stable.

† Currently working at Rhythm & Hues Studio.

© The Eurographics Association 2003.

1.1. Previous work

This section focuses on animation methods that can be used for animating long hair†. Hair modeling and rendering are not addressed. A complete state of the art can be found elsewhere [20, 23, 4, 17].

Most methods for animating hair rely on a physically-based dynamics model. A brute-force dynamics simulation would consume unacceptable computational resources, due to the large number of hair strands ($\sim$ 100,000) and the complexity of their interactions. Early work on hair animation does not consider the problem of mutual hair interaction and mostly focuses on the motion of individual hair strand [24, 1, 26]. For efficient processing of mutual hair interactions, an auxiliary mechanism is often employed. Examples include the cloth-like patches used for the production of Final Fantasy [14] and a layered shell to add volume around the head model [19], but these are limited to modeling smooth lateral interactions (as if hairs were layers of patches), rather than the realistic complex motions and interactions we seek to model.

Hadap and Magnenat-Thalmann propose a novel paradigm for handling mutual hair interaction by considering hair as a continuum and coupling a fluid dynamics model with an elaborate model for stiff hair dynamics [12]. The smoothed particle approach is used to model both hair-hair and hair-air interactions. The method gives high quality results for straight, smooth hair motions and interactions. However, it is computationally expensive since every hair strand needs to be processed, and results do not illustrate the dynamic clustering effects observed in real hair motion. Our goal is to mimic such complex dynamic clustering behavior, which we find essential when dealing with most hair styles.

For efficient computation of hair motion, neighboring hairs are often grouped and approximated by compacter representations. Among the two main approaches to hair grouping, the first approach uses *guide hair* or *key hair* to model a sparse set of hair strands, from which dense hair is interpolated at the rendering stage [6, 4]. These methods are useful for modeling and animating smooth or short hairstyles and animal fur [2, 9]. The second approach uses *wisps* to group nearby hairs into representative geometric primitives (hair wisps), inside which individual hair strands are rendered. This approach is employed as a few large wisps (e.g., in Shrek [8]), or surface wisps [18], or volumetric wisps [27, 23], that are deformable or rigid.

In the efforts to expedite hair animation by reducing the number of animated primitives, processing mutual hair interaction remains a challenge. Chang et al. [4] models the hair medium as a continuum, approximated by polygonal strips connecting the guide hairs. Limiting the interactions to only those between the polygons and guide hairs greatly reduces the computation needed for modeling mutual hair interaction. Additional guide hairs are adaptively inserted to ensure that guide hairs remain evenly distributed during hair motion. However, the underlying continuum assumption limits the method to relatively simple and smooth motion, since general hair motion does not maintain the strand-neighbor relationships assumed by guide hair approaches.

The geometry and control provided in the wisp approaches can model the complex hair geometry and clustering effects of long, thick hair. Plante et al. [23] models hair interaction by allowing individual wisp to deform. While each wisp models the coherent motion of neighboring hair strands, the interaction between wisps are handled with an anisotropic collision model- i.e., damped collisions occur when two wisps of different orientations collide, while wisps with similar orientation are allowed to interpenetrate with a high friction coefficient. The method proves useful for simulating complex clustering effects in hair motion, but remains inefficient (3 hours for a few seconds of animation) due to the complex shape of hair wisps and the number of penetration tests. More importantly, this approach can be unstable at the rest state due to the high number of mutual interactions between wisps. Our mechanism for processing interactions is related to this method, but our adaptive approach improves upon both efficiency and stability of the animations.

The use of adaptive level of detail is a well known way to improve efficiency during simulations. Previous adaptive deformable models such as [10] were devoted to the simulation of continuous medium. Chang et al.'s approach is also based on a continuum assumption as interpolation is used to add extra guide strands. Our claim is that hair *is not a continuous medium*. Strong discontinuities in geometry can be observed during motion, especially at the tip of hairs. A multiresolution hair representation was proposed to model such characteristics of hair geometry [17], which proved effective for modeling a variety of complex hairstyles. Yet this work was not exploited for hair animation. Very recently, Ward et al. [25] proposed a novel approach for modeling hair using three geometric levels of detail : hair strips, hair clusters and individual hair strands. Transitions between these LODs are dynamically generated during motion, according to criteria such as camera position. This method yields an increased efficiency, especially at the rendering stage. Although promising, the results are shown only for gentle and smooth motions such as hair floating in the wind. In contrast, our approach focuses on the animation level of details, i.e. dynamically adapting the control structure for more complex hair motion. Our criteria for driving transitions between animation is different from the work of Ward et al. in that our method is driven by the local complexity of hair motion instead of the importance of the object on the screen for rendering [25]. Therefore, our approach is independent of any adaptation of such rendering LOD and could be coupled with Ward's method to increase the rendering efficiency as well.

† In the remainder of this paper, the term "hair" will be used for human-like hair as opposed to fur. We believe that such hair poses unsolved problems due to the complexity of hair interactions.

© The Eurographics Association 2003.

1.2. Overview

Our goal is to simulate the dynamic clustering behavior of hair motion and the mutual hair interactions. We strive for both efficiency and stability. The key idea is to dynamically adapt the local resolution of the control structure during animation, thereby concentrating computations where it is most needed, and reducing the number of interacting elements to maintain stability.
We employ a multiresolution hair wisp representation similar to that used for hair modeling by Kim and Neumann [17]. At each time step, interactions between active hair wisps are detected and processed, similarly to Plante et al.'s work [23].
The contributions of our method are :

- Adaptive wisp tree, a novel adaptive control structure for hair animation that simulates dynamic splitting and merging of hair clusters;
- Rules for detecting areas where refinement and merging should occur;
- Improving efficiency and stability of interaction processing in wisp-based approaches.

Sections 2 to 4 respectively describe these contributions. The last two sections discuss implementation and results before concluding.

2. Adaptive wisp tree

During real hair motion, clusters form and split due to frictions, static charges, and the features of the initial hairstyles. The number, location and size of the observed clusters vary over time. The adaptive wisp tree (AWT) is motivated by the fact that the clustering behaviors are observed moving from hair tips to roots, as if hair was progressively cut by a slider. We attribute the phenomena to the excessive collision between hairs near the scalp and to the fact that hair motions are constrained at the root, while hair is free to move apart at the end. Physical simulation of such behavior would require considerable amount of computation and pose stability issues in numerical integration. We rather aim at a simple data structure that visually mimics the clustering behavior, leading to a tree-like control structure that approximates hair cluster splitting and merging. See Figure 2.

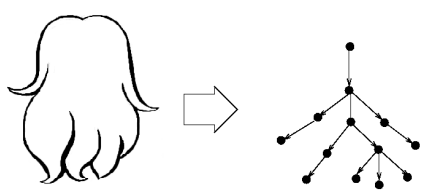

Figure 2: *The AWT is an acyclic oriented graph of wisp segments where nodes approximate the mass of nearby hairs and edges represent the control links between nodes.*

The geometric complexity and motion complexity are not always correlated : for instance, curly hair modeled as a large number of spiral-looking small wisps may move as coherently as a single large hair cluster for slight head movement, whereas seemingly smooth hair wisps often suddenly split into small clusters under drastic head motion. To model this behavior, we employ a multiresolution hair model to preserve the geometric detail, coupled with the AWT that controls the dynamic evolution of hair motion. The multiresolution geometric clusters are also used to guide potential splitting in a particular hairstyle during subsequent motion processing.

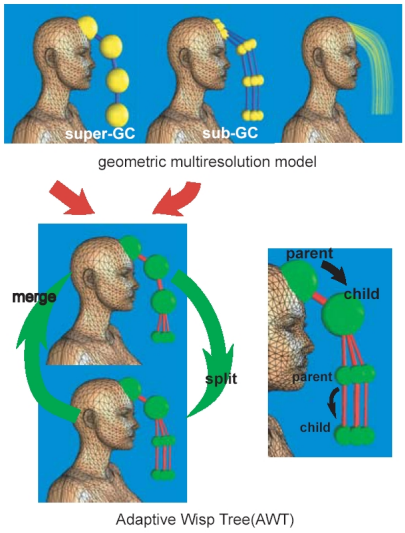

Figure 3: *Relationship between geometric multiresolution model and the AWT*

2.1. Multiresolution hair geometry

We represent a hair model with a hierarchy of generalized cylinders (GC), adapting from [17]. A hairstyle is progressively constructed by creating GCs and refining them into smaller ones. In the remainder of the paper, we use the terms *super-GC*, *sub-GC* when we refer to the modeling hierarchy. We reserve *parent* and *child* notations for denoting relationship between control links in the AWT (Figure 3).
The following modifications are made to [17] :

- The number of segments in sub-GCs should be multiples of that of the super-GC so that corresponding nodes can be easily identified.
- GCs have only circular cross-sections. Complex shapes emerge during animation as the result of splitting procedure. The condition greatly simplifies collision processing since a single radius per cross-section suffices, in contrast to complex deformable boundary used by [23].
- The framework of [17] does not enforce that the boundary of a super-GC encloses all its sub-GCs. The boundary of a super-GC is modified to fit the shape of its sub-GCs, in a post-processing step. Given a set of sub-GCs, we first compute the center-of-mass curve, and then modify the GC contours around this curve - a mere radius change in our case.

© The Eurographics Association 2003.

2.2. Multiresolution mechanical structure

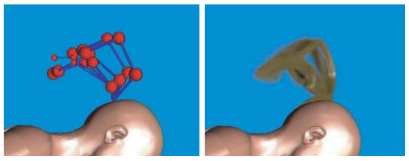

Figure 4: *Left image illustrates the AWT. Active nodes are drawn as spheres and segments are drawn as cylinders. Right image shows the corresponding hair rendering.*

The motion of each GC is approximated by a discrete set of nodes (Figure 4). A *node* corresponds to the center of the cross-section of the GC in the model hierarchy, the super-GC being refined at the end of the modeling stage, so that the number of cross-sections is the same at each hierarchy level. With each node, we store pointers to the corresponding nodes in the super-GC and sub-GC. We call such nodes *super-nodes* and *sub-nodes*, respectively. We also store the radius of the associated GC cross-section and the mass, computed from the number of hair strands that it simulates.
A set of *active* nodes partitions each GC so that each section of the finest level GC is governed by one and only one of them. The level of active nodes determines whether a GC section around the node is either animated as a coherent cluster or as a set of refined sub-GCs. We call *segment* the links that connect the set of active nodes, constructing a tree structure, the AWT (see Figure 5).
The AWT evolves during animation through nodes separation (a node splits into sub-nodes) or fusion (sub-nodes merge into their super-node). To ensure that the structure remains as a tree, merging is only allowed from root to tip, and splitting is allowed from tip to root[‡].
Dynamic splitting and merging occur only in the control model (AWT), not in the geometric model. Before animation, each hair strand is attached to one of the finest GCs in the model hierarchy and thus controlled by the set of active segments at any time during animation. We thus ensure that the visual motion of geometric model remains consistent despite the discrete changes in the LOD of the control model.

2.3. Animation

In practice, an AWT is built for each top-level GC in the model hierarchy; the entire hair being partitioned as a forest of such AWTs. At the beginning of the animation, often at rest, each AWT is initialized with only the coarsest nodes active. During animation, the tree evolves over time based on the fusion and separation processes.
Segments in the AWT are represented by either rigid links or viscous springs, depending on the desired behavior. Since a curly wisp should elongate during animation, soft springs are preferred for curly hair, whereas straight hairs are realistically modeled with rigid links. Among all the available simulation methods for this kind of mechanical models (see for instance [12, 9, 23, 4]), we have currently adopted a fast, approximate iterative method for simulating rigid links [21] as well as an implicit integration method for spring links [9]. Using Pai's strand model [22] may be a good alternate solution.
External conditions such as head movement, collisions, wind force fields, gravity, etc. affect the motion of the AWTs, constrained by the dynamic model.

‡ We ignore rare cases where the tip of hair remains coherent while other parts move apart. In theory, this situation can be handled by turning the tree into a graph with cycles, at the cost of a more complex data-structure than our current implementation.

3. Adapting the AWT

The AWT adapts over time by dual processes - splitting and merging. The splitting process locally refines the hair motion when a single wisp segment is not sufficient for the motion and deformation. The merging process simplifies the AWT when the motion becomes coherent again. This adaptation greatly simplifies the interaction processing as detailed in section 4.

3.1. Splitting (separation)

Our splitting criteria is based on the observation that motion becomes more complex in regions where hair accelerates. A node of the AWT splits if :

1. Its acceleration times radius is over a threshold value; and
2. Its child node in the AWT has already been split.

The first condition models the observed behavior that a larger (radius) or faster (accelerating) wisp is more likely to split. The second condition enforces the rule that splitting occurs from tip to root. Figure 5 shows the splitting process. When a node splits into a set of sub-nodes :

- The position of each sub-node is updated with an offset vector computed initially and whenever merging occurs (see 3.3).
- The velocity of sub-nodes are set to the velocity of the split node.

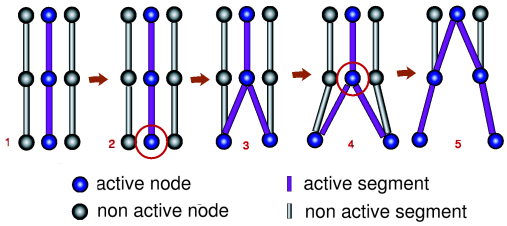

Figure 5: *The splitting process. 1. Initially, only coarse nodes are activated. 2. A node inside the circle meets the split criteria. 3. Its sub-nodes are activated 4. After a few time steps, the node inside the circle needs to split 5. Resulting AWT.*

3.2. Merging (fusion)

Nodes sharing *the same parent* merge if their super-node is capable of approximating their motion. The merging conditions are :

1. The merged nodes are contained within the radius of the super-node.

© The Eurographics Association 2003.

2. Their relative velocities are under a given threshold in magnitude, to avoid discontinuity in velocity when merging occurs.
3. The splitting condition 1 is not true (this test is added to prevent merging nodes from immediately splitting again).

Note that our merging test is local - it is only performed for the nodes with the same parent node. This way, the AWT eliminates any need to perform tests between arbitrary wisp segments. The tree structure of animated hair is preserved since merging takes place from the root to the tip of hair.
When nodes merge into a super-node :

- The super-node is positioned at the center of mass of the merging nodes. Its velocity is set to the mean velocity, weighted by masses of the merging nodes. Its radius is modified to tightly fit all the merging nodes.
- The offset vector for each merged node is computed and their positions are frozen with respect to the super-node thereafter.

3.3. Positions of Non-Active Nodes

To update the embedded multiresolution geometry, the position of non-active nodes should be updated during motion. Initially or when merging is performed, for each non-active node, an offset vector is computed that stores the node's relative position with respect to its super-node. At each time step, the position of a (non-active) sub-node is updated with the offset vector, rotated by a matrix that transforms the tangent vector of the active segment at the time of merging or initially, to the new tangent vector.

4. Hair interaction

One of the key benefit of the AWT is that it implicitly models mutual hair interactions so that neighboring wisps with similar motions merge, mimicking the static friction in real hair. This avoids subsequent collision processing between these wisps, thus increasing efficiency as well as gaining stability from the reduced number of primitives. In addition, the splitting behavior models wisps deformation without the need of the complex deformable wisp geometry used in previous work [23]. For collision processing, active segments in the AWT are thus represented by cylinders, which greatly simplifies collision detection tests.

4.1. Collision detection

The character's body is surrounded by a few bounding spheres, and a space voxel grid is used to store the active wisp segments at each time step. Each time an AWT segment penetrates a bounding volume, or when two segments intersect the same voxel, the wisp segment geometry is used for collision detection. This geometry is simply defined as a cylinder that connects the associated nodes in the wisp tree (a parent N_p and a child node N_c). The cylinder radius is set to the child node's radius[§] since the parent node may belong to a coarser level of detail (see Figure 6). For instance, a collision between two wisp segments is detected if the distance $d = d(P_1, P_2)$ between the closest points on their axes is smaller than the sum of their radii $r = r_1 + r_2$.
Note that collisions do not need to be detected when neighboring elements already have the same velocity. Testing for collisions in this case would even introduce artifacts, due to our split/merge process. When a merging occurs (e.g. when hair has come to rest), the coarser segment geometry only approximates its sub-segments geometry. Suddenly using this coarse geometry for collision detection would produce new, undesired motion towards a slightly different rest state. To avoid this problem, collision test is only performed between pairs of elements with relative motion differences.

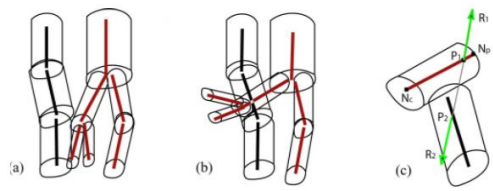

Figure 6: *Hair mutual interactions are detected efficiently since wisp segments are simple cylinders. (a) Friction forces are generated by the interaction of wisps segments of similar orientation; (b) soft response forces are added in other cases, (c) collision response being computed at closest points and then applied at mass nodes.*

4.2. Response to mutual hair interaction

Our processing of interactions is inspired from Plante et al.'s anisotropic collision response model [23]. The generated response forces allow wisp segments of similar orientation to interpenetrate, but high viscous friction is produced in this case. Response forces also prevent penetration between colliding wisp segments of different orientation, thus modeling the fact that a hair strand cannot cross each other strand (see Figure 6 (a) and (b)).
This anisotropic behavior can easily be modeled by computing viscous friction forces V_1 and V_2 at the closest points P_1 and P_2 between the segment axes :

$$V_1 = k_f(\dot{P}_2 - \dot{P}_1); \quad V_2 = -V_1 \tag{1}$$

and by adding repulsion forces R_1 and R_2 when the angle between the two segments is over a threshold :

$$R_1 = k_r(r-d)\frac{(P_1 - P_2)}{\|P_1 - P_2\|}; \quad R_2 = -R_1 \tag{2}$$

The resulting force $F = V + R$ at P (where P is P_1 or P_2) is then splitted into two forces F_p and F_c respectively, and added to the set of external actions applied at the segment

[§] Tapered cylinders computed from the parent and child nodes radii could be used instead. However, collision detection would be more intricate since closest points between two tapered cylinders do not necessarily correspond to the closest points between their axes. We did not observe any noticeable artifacts due to the use of standard cylinders.

© The Eurographics Association 2003.

nodes (see Figure 6, (c)) : if $P = uN_p + (1-u)N_c$,

$$F_p = uF; \quad F_c = (1-u)F \tag{3}$$

The friction coefficient k_f used in equation (1) should vary depending on the hierarchy level of the wisp segment. Friction forces should reduce the relative speed between nodes, but never change its orientation, which may occur if mass is too small with respect to the applied force. To maintain k_f at a reasonable value while ensuring the same friction behavior for all resolution levels, we use : $k_f = \alpha m$, where m is the mass of the child node N_c. We also check that the integration step is small enough to prevent the friction from reversing the direction of relative motion ($\alpha dt < 1$).

4.3. Hair interaction with obstacles

Hair is much lighter than the other objects in the scene, including the character's body. Hair position and velocity are modified to model response to collisions with obstacles as in [23]. An active wisp segment that intersects an obstacle is moved to remain in contact with it, by moving the child node N_c. The node's velocity is set to obstacle's velocity to model static friction, which we find more realistic than a bouncing behavior in the case of hair.

5. Results and discussion

In the examples shown, the mass of each coarse node was set so that the mass of the whole hair reaches between $500g$ and $2kg$ (that is, around $1g$ per coarse node), knowing that splitting an merging mechanisms obviouly ensure conservation of mass.
In our interactive interface, the splitting threshold can vary from 0.1 to $0.5N.m.kg^{-1}$. The value set was chosen according to the desirable importance of splitting during motion; for curly hair, we chose a high value (less splitting) so that hair remains grouped in wisps, whereas for smooth hair we chose a low value (more splitting) so that hair can visibily expand during motion. Only 2 levels of detail thus turned out to be sufficient for animating curly hair; for the simulation of smooth hair, we used 3 levels of detail to get a good compromise between realism and computational costs. More levels of detail could be used to increase realism.

5.1. Qualitative results

Snapshots taken from animation examples are shown in Figure 7 (see color plate Section). A variety of hairstyles were animated with varying character motions such as jumping, leaning, running, etc. Thanks to the merging process, our model stabilizes the hair motion even at high speed.
The animations shown in Figure 7 were rendered with about 10,000 individual hair strands. For rendering, we use a modified version of opacity shadow maps [16] for hair-hair shadow computation. Local hair reflectance is based on Kajiya-Kay hair reflectance model [13], implemented in nVidia's GC shader. In each frame, hair strands are drawn as OpenGL polylines, sorted by the visibility ordering algorithm in [17] for antialiasing. Rendering takes about a fraction of second to several seconds, depending on the desired features listed above.

5.2. Performance

Hair model (N, L)	single-level time (sec)	AWT mean time (sec)
Short (5149, 2)	3.1	0.32
Short curly (4909, 2)	2.7	0.27
Long (6065, 3)	3.8	0.09
Long curly (9853, 3)	7.9	0.29

The above table compares our method with the performance of a single-level animation method where all the finest GC are simulated. With each hair model, we show the number of nodes (N) at the finest level and the number of geometric levels of detail (L). The single-level method was used only to measure how much efficiency we get with the adaptive method. The time for the AWT is a mean value, reflecting split / merge frequency during animation. The table shows time (in seconds) to compute each animation step for 4 different hair models with same (leaning) motion of the character body. These values were measured on a PC with an 1.7 GHz Pentium CPU. We used 4 integration steps per each frame ($dt = 10ms$). The maximum time to compute 10 seconds-long animation was about 5 minutes.

5.3. Discussion

In terms of time performance, our method scales favorably; it runs orders of magnitude faster than the work by Plante [23] which similarly used volumetric wisps for animation but without exploiting multiresolution. The time measurements are comparable to those of Ward et al. [25], obtained with a level of detail representation for hair.
The AWT is best suited for fast and drastic head motions, in contrast to other methods where only slow and smooth hear motions were illustrated. Our approach can yield strong clustered appearance since we do not allow hair wisps to deform after they are split. Such appearance is essential for many kinds of hair (e.g. curly hair, thick hair, etc.), but may not be always desirable (e.g. for very smooth hair).
Our current choices for wisp splitting criteria could be further extended. We are currently investigating into an unified set of criteria that would determine whether a wisp should split or not. Possible extensions could include the body action on the hair (including collision and friction), interaction between air and wisps, and interaction inside hair. Lastly, the recent work by Ward et al.'s [25] on geometric LODs could be combined with our animation LOD algorithms, for increased efficiency at the rendering stage.

6. Conclusion and Future Work

We presented an effective adaptive method for hair animation. Our adaptive wisp tree mimics the dynamic clustering behavior of hair motion often observed in long, thick hair as

© The Eurographics Association 2003.

well as in curly hair. The reduced number of wisp segments and their simple shapes result in efficient processing of complex mutual hair interaction, as well as increased stability of the simulation.

Our model is best suited for large-scale to mid-scale behavior of long, thick hair, such as splitting and merging of clusters. As hair wisp refines, collision detection takes place between finer and finer cylinders, necessitating smaller time steps to handle mutual hair interaction during fast motion. For such small-scale interactions, an alternative approach such as Hadap and Thalmann's smoothed particle model [12] could be better suited. To guarantee continuity of hair representation, it could be also interesting to investigate a hybrid approach that combines a very smooth hair representation (e.g. using the adaptive guide hair method [4]) with dynamic clustering effects provided by the adaptive wisp tree.

References

1. K. Anjyo, Y. Usami, and T. Kurihara. A simple method for extracting the natural beauty of hair. In *Proceedings of ACM SIGGRAPH 1992*, Computer Graphics Proceedings, Annual Conference Series, pages 111–120, August 1992.

2. J. Berney and J. K. Redd. A tale of fur, costumes, performance, and integration. *SIGGRAPH Course 14*, 2000.

3. S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović. A multiresolution framework for dynamic deformations. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 41–48, July 2002.

4. J. T. Chang, J. Jin, and Y. Yu. A practical model for hair mutual interactions. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 73–80, July 2002.

5. L. Chen, S. Saeyor, H. Dohi, and M. Ishizuka. A system of 3d hairstyle synthesis based on the wisp model. *The Visual Computer*, 15(4):159–170, 1999.

6. A. Daldegan, N. M. Thalmann, T. Kurihara, and D. Thalmann. An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum*, 12(3):211–221, 1993.

7. G. Debunne, M. Desbrun, M-P. Cani, and A. H. Barr. Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 31–36, August 2001.

8. R. Falk and L. R. Sande. Shrek: The story behind the screen. *SIGGRAPH Course Note 19*, 2001.

9. M. Fong. Animating monster fur. *SIGGRAPH Course 36: From Ivory Tower to Silver Screen*, 2001.

10. E. Grinspun, P. Krysl, and P. Schröder. Charms: A simple framework for adaptive simulation. *ACM Transactions on Graphics*, 21(3):281–290, July 2002. Proceedings of ACM SIGGRAPH 2002.

11. S. Hadap and N. Magnenat-Thalmann. Interactive hair styler based on fluid flow. In *Computer Animation and Simulation '00*, pages 87–100, August 2000.

12. S. Hadap and N. Magnenat-Thalmann. Modeling dynamic hair as a continuum. *Computer Graphics Forum*, 20(3):329–338, 2001. Proceedings of Eurographics'01.

13. J. Kajiya and T. Kay. Rendering fur with three dimensional textures. In *Proceedings of ACM SIGGRAPH 89*, Computer Graphics Proceedings, Annual Conference Series, pages 271–280, 1989.

14. S. Kent. *The Making of Final Fantasy. The Sprits Within*. Brady Publishing, Indiana, 2001.

15. T-Y. Kim and U. Neumann. A thin shell volume for modeling human hair. In *Computer Animation 2000*, IEEE Computer Society, pages 121–128, 2000.

16. T-Y. Kim and U. Neumann. Opacity shadow maps. In *Rendering Techniques 2001*, Springer, pages 177–182, July 2001.

17. T-Y. Kim and U. Neumann. Interactive multiresolution hair modeling and editing. *ACM Transactions on Graphics*, 21(3):620–629, July 2002. Proceedings of ACM SIGGRAPH 2002.

18. C. Koh and Z. Huang. A simple physics model to animate human hair modeled in 2D strips in real time. In *Computer Animation and Simulation '01*, pages 127–138, September 2001.

19. D-W. Lee and H-S. Ko. Natural hairstyle modeling and animation. *Graphical Models*, 63(2):67–85, March 2001.

20. N. Magnenat-Thalmann and S. Hadap. State of the art in hair simulation. In *International Workshop on Human Modeling and Animation*, Korea Computer Graphics Society, pages 3–9, 2000.

21. C. Van Overveld. An iterative approach to dynamic simulation of 3-D rigid-body motions for real-time interactive computer animation. *The Visual Computer*, 7:29–38, 1991.

22. D. Pai. Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum*, 21(3):347–352, 2002. Proceedings of Eurographics'02.

23. E. Plante, M-P. Cani, and P. Poulin. Capturing the complexity of hair motion. *GMOD*, 64(1), january 2002.

24. R. Rosenblum, W. Carlson, and E. Tripp. Simulating the structure and dynamics of human hair: Modeling, rendering, and animation. *The Journal of Visualization and Computer Animation*, 2(4):141–148, 1991.

25. K. Ward, M. C. Lin, J. Lee, S. Fisher, and D. Macri. Modeling hair using level-of-detail representations. In *International Conference on Computer Animation and Social Agents*, May 2003.

26. Y. Watanabe and Y. Suenaga. A trigonal prism-based method for hair image generation. *IEEE Computer Graphics and Applications*, 12(1):47–53, 1992.

27. Z. Xu and X. D. Yang. V-hairstudio: an interactive tool for hair design. *IEEE Computer Graphics & Applications*, 21(3):36–42, May / June 2001.

28. X. D. Yang, Z. Xu, J. Yang, and T. Wang. The cluster hair model. *GMOD*, 62(2), 2000.

© The Eurographics Association 2003.

Figure 7: *From top to bottom : short, smooth hair in running motion; long, thick hair in leaning motion; long, curly hair in leaning motion; short, curly hair in leaning motion*

© The Eurographics Association 2003.