An Analysis of Real-time Rendering and Simulation of Hair and Fur
By Matthew Sacchitella
November 4th, 2010
I. Table of Contents

A. Research Analysis Paper ......................................................................................p. 3-9

B. Abstract #1: “Real-time Fur Simulation and Rendering.” .........................................................p. 10-11

C. Abstract #2: “Real-time Hair Simulation and Rendering on the GPU.” ............................p. 11-12


F. Abstract #5: “Interactive Virtual Hair Salon.” .................................................................p. 14-15


H. Abstract#7: “Real-time Fur with Precomputed Radiance Transfer.” .................................p. 16-17

I. Abstract #8: “Practical Real-time Hair Rendering and Shading.” ........................................p. 17

J. Other Sources ........................................................................................................p. 18
In Animation today, I would argue that most if not all of the major production practices done in 3D contain a single commonality, the quest to find ways to further inverse the relationship between time and product quality. That is, to say, we are today, as we always have been, seeking ways to yield higher and higher results with less and less time devoted to obtaining these results. It can be seen in nearly every aspect of animation production. Innovations across the board; Motion Capture, allows us to generate character animation and gestures more quickly than key frame animation, and often times with tweaking will yield a more “natural” result (where as keyframe animation has a tendency to be pseudo-natural, in it’s attempt to replicate true human motion). Advanced Rigging with Mel and Python, allow us to generate more and more complex rigs, in less and less time. Every year, more complex algorithms and more advanced GPUs and hardware, make way for faster rendering times. And somewhere in the middle of all this lies hair and fur. Like all other aspects, they stand at a fork. The crossroads of realism vs. render times. Down the right path lies a goal of lifelike hair, an attempt to mimic some of the difficult qualities found in real human hair or animal fur (i.e. anisotropic reflection, self-shadowing, semi-transparency), and to the left lies render speed, a goal focused on pumping out a product, and doing so quickly. For hair and fur, however, the fork has quickly narrowed into a single road, with an initial real-time fur rendering done by Microsoft Research and professors at Princeton in 2001 (S1), and realistic real-time hair rendering accomplished as early as 2003 in the Nalu Demo at NVIDIA’s GeForce 6800 Launch (P10).

The first of the two aspects I researched was real-time rendering and simulation of realistic hair, which due to its (hair’s) very nature could be looked at as a more difficult
task to accomplish than fur. The bulk of the research I reviewed, stressed the difficulty in recreating realistic hair due to hair’s “several specific characteristics,” (P6). Said characteristics can be summarized as anisotropic reflections (P4,P8), a self-shadowing property (P4,P5, P7), complicated geometry (P4,P6, P7,P8,P10), sheer volume (P4,P6,P7, and hair strand clustering (P2, P3, P5, P8).

The various studies, at parts, tend to overlap in their methodology towards achieving these attributes, but at times can differ. The latter is the case for anisotropic reflections, as most of the work done tends to choose one of two base models. The first “proposed by Heidrich et al.” is used primarily for it’s “simplicity and its capability of exploiting graphics hardware,” (P4). The second uses diffuse and specular algorithms proposed by Kajiya and Kay to create “fake dual specular highlights, moving the primary to the tip of the hair and the secondary towards the root,” (S2). Both of these methods are capable of creating fairly realistic results in real-time, however, based on images of rendered results that I have seen, it is my opinion that many of the models developed from Kajiya and Kay’s algorithms seem to produce products truer to photorealism, especially recent work done by Tariq and Bavoil of NVIDIA, presented at SIGGRAPH ’08.

The second aspect of realistic hair is it’s self-shadowing property, which according to many of the papers is one of the most computationally expensive aspects of rendering photo real hair; fortunately today this burden is mainly placed on the GPU, using self-shadowing algorithms developed by a host of people involved in computer graphics. However, most of the articles I read seemed to use the base model from Kajiya and Kay, and build upon it using various others insights (i.e. Interactive Virtual Hair
Salon” uses Kajiya and Kay’s base model, as well as Heidrich and Seidel’s “shifting tangents”, and Kim & Neumann’s “opacity shadow maps,”).

As previously stated, complicated geometry, sheer volume, and hair strand clustering are other difficulties that must be faced to obtain realistic real-time hair simulation and rendering. Geometry and volume are typically dealt with through interpolation. The two main types of interpolation I read about are “Multi-strand Interpolation” and “Clump-based interpolation” (also referred to as wisps). (P4, S2). Clump based interpolation creates clumps of hair as the hair moves, where multi-strand creates smooth flowing single strands. A combination of the two methods, produces some the closest to photo real results possible, and is also still possible in real-time. (S2). Hair strand clustering, the natural tendency of strands of hair to group into clusters when moving, is accomplished through this addition of clump-based interpolation.

It is clear from the many aspects that define photo real hair that it is no easy task. How hard then is working towards real-time fur simulation? After reading research on real-time fur rendering and simulation, I can say for certain, just as hard. Both hair and fur simulations must deal with some of the same overlapping issues. The first and most obvious is, as in hair, the sheer volume and geometric complexity required to create realistic fur. According to Kloetzli in “Real-Time Fur with Precomputed Radiance Transfer,” “rendering microsurfaces is a difficult task in computer graphics. Because microsurfaces are by definition very high frequency geometry, traditional rasterization or ray tracing techniques bog down to the point of uselessness or are plagued with terrible aliasing artifacts,” (P7). Both fur and hair have to deal with one hundred thousand or more individual hairs/strands, and thus incorporate some of the same methodology to
render them. Similar to hair, most realistic and real-time fur simulation and rendering makes use of Kayija and Kay’s raytrace algorithms to overcome these limitations as well as to provide self-shadowing.

While many of the methods used to create real-time fur are similar to those used to make hair, fur also has some methods unique to it. One of these deals with the creation of the fur geometry and is called the shell and fin method, outlined in “Real-time Fur Over Abitrary Surfaces,” and used and built upon in most of the realistic fur simulations done to date, (P1, S1, P3,P7). This method, essentially creates an offset shell (or plane) a given distance from the original mesh, and extrudes planes from the original mesh to the offset shell, which is then combined with lapped texturing to produce a realistic fur result renderable and simulated in real-time. This difference in geometry creation provides the main difference between generating hair and fur (hundreds of thousands of tiny hairs vs one hundred thousand longer strands) in 3d computer graphics.

After examining a plethora of research on the topic of real-time rendering and simulation of hair and fur, I now believe I have the insight to make a few observations and predictions about what the future holds for this aspect of 3D animation. The first of which is gathered from reading of my sources presented by NVIDIA (P2,S2,P9,P10). That is, the call has come for the incorporation of real-time rendering of realistic hair and fur into the 3D worlds created by video games. NVIDIA made the call at SIGGRAPH 2010, stating in lecture notes that, “With the advancement of the graphic hardware, we believe that the time has come to handle hair rendering properly in real-time graphics applications.” With the world leader in visual computing technologies backing it, it is
safe to say we will likely see the incorporation of these technologies into video games sooner rather than later.

Another of my predictions relates to a specific simulation I read about, done by Kelly Ward of Walt Disney Animation et al. The group developed an interactive real-time simulation of a virtual salon, complete with barber’s tools and a full head of hair. In my opinion this was the most interesting of all the research I read. The user, using haptics technology and a 3D interface, controls the simulation and is given the role of hair stylist. The tools available to the user include: scissors, water, hairspray, mousse, and a hair dryer.

The first, scissors, is able to cut off hair geometry in real-time. The method uses a “triangle formed by the space between the open blades of scissors” to determine the clipping plane, (P5). Upon closing of the scissors, the hair skeletons are cut, splitting into two separate skeletons, and allowing the skeleton not still connected to the scalp to fall. The user can also apply water to the hair, causing “the mass points of the global skeleton [to] become heavier with the mass of water. The overall motion of the hair is limited due to the extra weight and if the hair is curly, the global skeleton will stretch under the extra weight and the curls will lengthen as expected.” Hairspray and mousse are also available, which create “dynamic bonds…between sections of hair that are in contact when applied,” and “grow the radii of the hair sections affected” respectively. Lastly the user can utilize a hair-drying tool. For this tool, “when the stylus button is pressed, a strong constant force is applied in the direction of its orientation…[and] any control points that fall within the cone of influence receive the force. Moreover, if a wet control point is influenced by the hair dryer, the control point will ‘dry’,” (P5).
In my mind, this is another clear candidate for research that has great potential in the future. I believe further developed versions of this “interactive virtual salon” could easily be incorporated into learning/training at beauty schools for hair stylists. I would imagine for those starting out in the beauty industry, it is not easy to find test subjects to practice on. Nobody wants a haircut from someone with no experience. The incorporation of this tool, I believe would be more than capable of providing most of the training needed for a hair stylist to have confidence in their abilities before ever having to cut actual hair. It may be up to debate, how well the learning on a haptic interface, would transfer into real life cutting of hair, but it certainly couldn’t hurt to test my theory. While the research done on the “interactive virtual hair salon,” was very impressive. Like any research, there is room for improvements. Further versions could seek to incorporate a wider range of tools. The tools developed thus far are very impressive, but tools could also be developed for “hair wear” objects like bobby pins, scrunchies, etc. to give the user greater control over the hairstyle. Seamless changing of the hairs color is another aspect that could be incorporated.

My final prediction is for the future of real-time fur. Based on the information I’ve gathered, I now think of the developments in fur as a very close shadow to hair. After all, as I stated earlier, they both must attempt to overcome some of the very same hardships in order to yield realistic results. However, it seems, as though, real-time fur is lagging behind the innovations of real-time hair, which leads me to my final conclusion. As is common in many areas of computer graphics, we will always seek to apply knowledge across multiple aspects of 3D if they can be used to our advantage. We have seen some of the same principles brought over from hair to fur, and we’ve seen fur
develop some of it’s own. Whether fur continues to follow the same guidelines of hair, remains to be seen, but one thing is certain, when realistic real-time rendering of hair makes the leap into real-time graphics applications, realistic real-time fur will not be far off.
**Research Abstracts & Sources**

**Primary Source 1 (P1):**


**Secondary Source 1 (S1):**


**ABSTRACT #1**

“Real-time simulation of fur is a key element in many virtual reality applications. However, It is still difficult to simulate effects of fur animation in real-time...[the authors’ are proposing] an interactive fur simulation method, to calculate effects of external forces...and direct manipulation for real-time interactive animation. [The Method, their solution to the difficulties surrounding effects simulation on fur in real time.] “consists of two layered textures for rendering...[The first] represents volumes of fur...[The second] covers and laps the edge [of the first]...” “Each layer is based on the Shell and Fin method.” “Shell is a structure to visualize the volume of fur...[consisting of] several lapped textures with controlled opacity. A fin structure...[fills] gaps among lapped textures...[filling in] holes in the Shell structures... [And] providing more precise representation of fur because its size is small and used for detailed representation.” “[The authors’] approach unifies these two structures using a shared vertex array to enhance rendering performance.” “The proposed system creates the mesh data of Shell and Fin based on a base mesh...[then uses an algorithm and 6-step method] to generate the shared vertices of fur with Shell and Fin structure.” “After the generation of the shared vertices, the proposed system creates faces of Fins and Shells that are connected through shared vertices...in the process of creating Shell faces, the system recreates the same number of faces in the base mesh...Faces of Fins are created by connecting neighbor vertices...[and] all of [the] faces are created after the iteration of these operations in all vertices.” “After the creation of Shell and Fin faces, a texture is generated to represent the shape of furs on Shells. The proposed method creates random Shell textures using a seed value...we utilize the seed value as a growing vector to represent the random direction of each fur strand...in addition, the proposed system can be customized, as it allows a user to manipulate different fur patterns during the creation process...if an artist requests the use of a special pattern, we can configure a desired texture pattern before the creation of a Shell texture. The given input texture is applied to generate the seed value for a random vector for texture generation. If a fur strand has several irregularities of colors, a color height map is used for natural fur representation.” “The method uses] “rotation angles to simulate [external and internal force simulation, by creating] “a force field, which stores a force per vertex in a base mesh...[and simulating] a rotation angle and a rotation axis using the inner and cross products between a force vector and a growing vector...[which enables] real-time simulation of various...forces.” [The authors’ method] “reduces about 25% of the extra costs of the redundancy of memory space using the shared
vertices architecture… [and compared to] straightforward dynamic fur simulation using 30 faces of Fin, [the method] can reduce around 85% of the extra memory space.”

Primary Source 2 (P2):


Secondary Source 2 (S2):


Secondary Source 3 (S3):


ABSTRACT #2

“[The authors] present a method for simulating and rendering realistic hair in real time using the power and programmability of modern GPUs…Our method utilizes new features of graphics hardware (like Stream Output, Geometry Shader and Texture Buffers) that make it possible for all simulation and rendering to be processed on the GPU in an intuitive manner, with no need for CPU intervention or read back. In addition, we propose fast new algorithms for inter-hair collision, and collision detection and resolution of interpolate hair.”

“We simulate the hair based on a particle constraint method [discussed in “Position Based Dynamics”], which is extremely parallelizable and well suited to be implemented on the GPU.”

“We simulate the guide hair, we create on long Vertex Buffer of positions of all the guide hairs, inserting them back to back...to simulate the movement of the hair we render this VB to another VB using the vertex shader [VS] and the Stream Output pipeline stage.”

[With regard to Inter-hair collisions, the authors] “are particularly interested in the volume preserving nature of [these collisions]…[thus] we create a voxelized representation of all the interpolated hair and then apply repulsive forces to hair vertices in high density areas….unlike [Bertail’s] approach...we formulate our forces to point in the direction of the negative gradient of the blurred density. These forces aim to push hair where we would intuitively want…towards areas of low density…we also voxelize collision obstacles into this density grid…[preventing] inter-hair forces from pushing hair into solid objects.

“We use a combination of two methods to generate the additional hair: [for rendering] clump based interpolation and barycentric interpolation. The clump based method creates additional hairs following a single guide. Barycentric interpolation adds new hairs within a ‘scalp triangle’ by interpolating from the hairs rooted at the triangle’s vertices.” [By rendering]…a set of dummy
lines, and [using] the VS to read and interpolate the appropriate simulated guide vertices. [the authors’ can] “render the interpolated hair.”

“The GS [expands] interpolated lines into camera facing triangle strips, and then finally [allows the authors’ to] render the hair with shading, shadows and Alpha to Coverage.”

[One difficulty] “with interpolating new hair from multiple guide hair occurs when the interpolated hairs go through the collision obstacle (for example because the guide hair ended up on different sides of an obstacle).”

“To avoid this we detect when any interpolated strand would penetrate an obstacle, and switch the interpolation mode of such strands to single-strand interpolation…to identify hair vertices below other object-penetrating vertices [the authors’]…render all the interpolated hair vertices to a texture, such that all the vertices in one interpolated strand are rendered to the same pixel. For each vertex we output to its offset index from the root if that vertex collides with an object or a large constant…[by using] minimum blending in this pass…[so that the texture generates] the first vertex that intersects an object. [Finally] “We use this texture to decide the interpolation mode to use one each vertex.”

Primary Source 3 (P3):


ABSTRACT #3

“It is important to model and render fur both realistically and quickly. When the objective is real-time performance, fur is usually represented by texture layers (or 3D textures), which limits the dynamic characteristics of fur when compared with methods that use an explicit representation for each fur strand.”

“This paper proposes a method for animating and shaping fur in real-time, adding curling and clumping effects to the existing real-time fur rendering methods on the GPU…fur bending [is achieved through]…a mass-spring strand model embedded in the fur texture. We add small scale displacements to layers to represent curls which are suitable for vertex shader implementation…[then we] use a fragment shader to compute intra-layer offsets to create fur clumps…Our method…[makes it] easy to dynamically add and remove fur curls and clumps [as seen in wet or dry fur].”

“In computer graphics, existing fur related researches concentrate either on realism or real-time rendering. In the latter, fur is usually represented by static 3D textures or texture layers, which limits the dynamic characteristics of fur when compared with methods that use an explicit representation for each fur strand.”

[The author’s] “propose a method for manipulating the fur shape while maintaining real-time performance. [With a focus on] (un)curling and (un)clumping effects visible when fur becomes wet or dries up, [the authors] add these effects to the existing real-time fur rendering methods on GPU.”
“[The authors] generate a 2D texture mask representing wet fur clumping areas, so that portions of the object surface can be selectively subject to wetness. We position a mass-spring fur strand model over the object surface, and use it to displace the fur texture layers to represent large scale deformation of the fur. This algorithm is aimed to be implemented in the GPU vertex shader.”

“[The authors then] use the same wetness mask to know if a strand is in a clump region, and if so they compute the displacement that the strand should suffer due to the clumping effect. This algorithm is suitable to be implemented in the GPU fragment shader, because different displacements need to be applied to each point within a layer, and they realize this using texture coordinate manipulation.”

“As a result, [their] method can dynamically add and remove fur curls and clumps, as can be seen in real fur when getting wet and drying up. ...to the best of [the authors’] knowledge, the effects possible with [their] method were not seen performed in real-time in the previous research.”

Primary Source 4 (P4):


ABSTRACT #4

“[The authors are presenting] a hair model together with rendering algorithms suitable for real-time rendering. In [their] approach, [they] take into account the major lighting factors contributing to a realistic appearance of human hair: anisotropic reflection and self-shadowing. To deal with the geometric complexity of human hair, we combine single hair fibers into hair wisps, which are represented by textured triangle strips. [The authors’] rendering algorithms use OpenGL extensions to achieve real-time performance on recent commodity graphics boards. “

“Despite the fact that GPU’s evolved very rapidly over the past years, it is still difficult to render about 100,000 single hair strands in real-time. Further rendering challenges arise from the specific material properties of hair [anisotropic reflection, self-shadowing, semi-transparent]...[the authors] propose a complete wisp model based on textured triangle strips...[complete with] the most important lighting effects for human hair...to achieve real-time rendering...[they propose] efficient implementations...[using] recent commodity graphics boards.”

[Since the author’s rendering system is designed as] “a combination of several plug-ins...the hair renderer does not depend on particular graphics hardware. With this plug-in infrastructure, it is possible [to] write different hair renderer that are optimally adjusted to different hardware platforms [like NVIDIA or ATI].”

“[The] hair wisp...[known as a] hair patch...reduces the geometric complexity of [their] hair model and thus accelerates the rendering process...[this process uses a variety of algorithms to compute anisotropic reflection and shadows]...[For] anisotropic
reflection…[they] decided to [build their] algorithm upon the anisotropic reflection model proposed by Heidrich et al. due to its simplicity and its capability of exploiting graphics hardware…For the classical Phong illumination model, we need to evaluate the well-known equation…\[Illumination[\text{out}] = Illumination[\text{ambient}] + Illumination[\text{diffuse}] + Illumination[\text{specular}]\]…[using the direction towards the light source, the surface normal, the direction towards the viewer, and the reflected light direction.]…using [this model]…we can efficiently evaluate the Phong and the Blinn-Phong model for any point in space.”

“For the shadowing of the hair, [the authors] have developed a modified version of the opacity shadow maps algorithm proposed by Kim and Neumann…to compute shadows on a per-pixel basis, allowing us also to cast shadows of the hair patches onto the head. To this end, the shadowing process is divided into two steps…compute all opacity maps, if an update is necessary…[and] compute shadow of fragments by back-projecting them into maps.”

“Rendering [the authors’] hair model with the real-time algorithms…yields a realistic appearance of human hair for a variety of hair styles…[the authors’ of the study have also] tested [their] hair rendering algorithms on several different graphics boards.”

Primary Source 5 (P5):

ABSTRACT #5

“User interaction with animated hair is useful for various applications…[but] due to the performance requirement, many interactive hair modeling algorithms tend to lack important, complex features of hair, including hair interactions, dynamic clustering of hair strands, and intricate self-shadowing effects…[as a result] realistic hair appearance and behavior are compromised for real-time interaction with hair.”

“Using simulation localization techniques, multi-resolution representations, and graphics hardware rendering acceleration…[the author’s] have developed a physically-based virtual hair salon system that simulates and renders hair at accelerated rates, enabling users to interactively style virtual hair. With a 3D haptic interface, users can directly manipulate and position hair strands, as well as employ real-world styling applications (cutting, blow drying, etc) to create hairstyles more intuitively than previous techniques.”

[To the best of the author’s knowledge] “there exists no method prior to [their] work that enables the user to interact and style virtual dynamic hair.” “A user…[of the author’s system] directly interact with hair through the 3D user interface and use operations commonly performed in hair salons. The operations…[supported in their system] include applying water, hair spray, and mousse to the hair, grabbing and moving sections of hair, using a hair dryer, and cutting the hair.”
“Cutting hair is crucial for changing a hairstyle…Lee and Ko (2001) cut hair with a cutting surface; hairs that intersect the surface are clipped to the surface shape. [The author’s] cutting method builds on the words of Lee and Ko (2001) to model cutting of hair performed with scissors as is used in a salon…the location for cutting is defined by a triangle formed by the space between the open blades of scissors.”

“When water is applied to the hair, the mass points of the global skeleton become heavier with the mass of the water. The overall motion of the hair is limited due to the extra weight and if the hair is curly, the global-skeleton will stretch under the extra weight [straightening the hair].”

“Hair spray is simulated on the hair by increasing the spring constraints of the global skeleton where it is applied…[In otherwords] dynamic bonds are added between sections of hair that are in contact when the hair spray is applied…[the mousse] adds volume to [the] hair…[by] growing the radii of the hair sections it affects.”

“Hair dryers are one of the most common tools in a hair salon. When the stylus button is pressed, a strong constant force is applied in the direction of its orientation. Any control points that fall within the cone of influence receive this strong force…[also] if a wet control point is influenced by the hair dryer, the point will “dry” [decreasing the amount of water depending on the length of exposure and power of the hair dryer force.]”

Primary Source 6 (P6):


ABSTRACT #6

“Realistically animating hair is no easy task…hair strands have a naturally anisostropic character; the length of a typical strand is several orders of magnitude larger than its diameter. Hair is also practically unstretchable and unshearable. At the same time it bends and twists easily, but resumes its rest shape when external strain is removed. These properties, combined with the fact that a typical human has over 100,000 individual hair strands, make accurate and fast physical simulation very difficult.”

“[The authors] concentrate on animating hair in real-time scenarios (like haptics), giving up strict realism in exchange for speed, but maintaining physical basis and plausibility. …[They] base their approach on…[what was] originally designed for larger, very flexible objects such as ropes…our main contribution lies in the development of a new method to handle twisting. Utilizing specific properties of hair strands, our method is both faster and more robust…Our twist method is fully capable of dealing with [the magnitude of processes required to simulate correct hair, and it is neither]…slowed down nor…[required to further shorten] the simulation time step… [the authors] method involves no matrix iterations, thus having a smaller memory footprint, and is easily parallelizable.”
“[The authors’] system simulates hair on a per-strand basis. The entire hair volume is viewed as a collection of individual leader strands, subject to physical simulation, and a greater number of follower strands, the state of which is just interpolated from leaders...this keeps the number of simulated strands at a manageable level, while still allowing non-uniform behavior in the hair volume.”

[The hair simulation for the authors’ outline is as follows] “pre compute rest-state values...while simulation running do...compute forces...integrate equations of motion...detect hair-head collisions...while constraints or collisions unsolved do...perform one constraint enforcement step...if position changed then update velocities...update Bishop frame...compute twist.” [The algorithm uses a do while loop to loop on itself and generate the simulated hair.]

Primary Source 7 (P7):


ABSTRACT #7

[The author] “introduces Precomputed Radiance Transfer (PRT) to shell textures in the context of real-time fur rendering. PRT is a method which allows static objects to have global illumination effects such as self shadowing and soft shadows while being rendered in real-time. This is done by precomputing radiance on the surface in a special basis that is chosen to allow reconstruction of correct illumination in arbitrary lighting environments...[the author uses Shell textures, or] 3D rings...around a model...[which] are transparent everywhere except at the intersection of the shell and the microgeometry that is being rendered.”

“Rendering microsurfaces is a difficult task in computer graphics. Because microsurfaces are by definition very high frequency geometry, traditional rasterization or ray tracing techniques bog down to the point of uselessness or are plagued with terrible aliasing artifacts...Fur is a perfect example of a microsurface, containing hundreds or thousands of hairs that are very small but certainly individually visible. Still, we need to render fur if we intend to have realistic animals in computer graphics. In addition, we need a very fast way to render it if we intend said animals to be in an interactive application such as a game.”

“Many different variations of PRT have been explored. It has been used to recreate diffuse...and specular...lighting, as well as model subsurface scattering...in real time. Multiple scales of PRT...can be used to create high-frequency local effects in addition to complete inter-model lighting. The ZH [Zonal Harmonics] basis functions can be used to represent easily rotated lighting, leading to PRT for deformable objects [otherwise impossible].”
“[The author aims to] create a realistic fur texture…compute lighting of the fur using an off-line technique…convert the lighting into a PRT basis…reconstruct the lighting in real-time…[using a] calculation…[he restricts] the width of hairs to be at most 1 pixel, limiting the number of adjacent pixels that a hair can intersect to four. Each of these pixels is numbered from the upper left clockwise…[this] allow[s] us to approximate the area of the pixel covered by the fur.”

Primary Source 8 (P8):


ABSTRACT #8:

“[The author presents] a real-time algorithm for hair rendering using a polygon model, which was used in the real-time animation Ruby: The Double Cross…the hair shading model is based on the Kajiya-Kay model, and adds a real-time approximation of realistic specular highlights as observed by Marschner et al. [The authors] also describe a simple technique to render the semi-transparent hair model in approximate back to front order. Instead of executing a special sorting step on the CPU at run-time, we render the opaque and transparent hair regions in separate passes to resolve visibility.”

“The hair model…is built with layered 2D polygon patches which provide a simple approximation of the volumetric qualities of hair. The polygonal model…[reduces the] load on the vertex processor, and simplifies…sorting the geometry from back to front.”

“Recently, Marschner et al. reported that hair has two distinguishable highlights. The first one is from light reflecting off the surface, and is shifted towards the tip of the hair. The second highlight is from light that is transmitted into the hair strand and reflected back out towards the viewer. The color of this highlight is modulated by the hair’s pigment color, and is shifted towards the hair root…to approximate [these] observations…we compute two separate specular terms per light source. The two terms have different specular colors…exponents and are shifted in opposite directions along the hair strand…[adding] a noise texture…[achieves an] inexpensive approximation of the sparkling appearance…in real hair.”

“[The authors’ model for hair rendering and shading is as follows] During pre-processing [they] sort the hair patches by their distance from the head and store the draw order in a static index buffer…During the first pass, [they] prime the Z buffer for the opaque regions of the hair model…[using] an alpha test to mask out the transparent parts…In the following passes, [they] use the full hair pixel shader…second pass…same opaque pixels as in the first pass get shaded. For the third…[they] draw all back-facing transparent regions. Finally…front facing transparent regions are rendered.”
Primary Source 9 (P9):


Primary Source 10 (P10):


Primary Source 11(P11):


Primary Source 12(P12):