INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

TECNOLÓGICO
DE MONTERREY
BIBLIOTECA
Campus Ciudad de México

HAIR PAINT: SISTEMA INTEGRADO PARA MODELADO INTERACTIVO, ANIMACIÓN Y VISUALIZACIÓN DE CABELLO

TESIS QUE PARA OPTAR EL GRADO DE
MAESTRO EN CIENCIAS COMPUTACIONALES
PRESENTA

BENJAMÍN HERNÁNDEZ ARREGUÍN

Asesor:	Dr. ISAAC JUAN RUDOMIN GOLDBERG	
Comité de tesis:	Dra. MARÍA DE LOS ANGELES JUNCO REY	
	Dr. ISAAC JUAN RUDOMIN GOLDBERG	
	Dr. BEDRICH BENES	
Jurado:	Dra. MA. DE LOS ANGELES JUNCO REY	Presidente
	Dr. BEDRICH BENES	Secretario
	Dr. ISAAC JUAN RUDOMIN GOLDBERG	Vocal

Atizapán de Zaragoza, Edo. Méx., Enero de 2004

RESUMEN

Una característica intrínseca en los seres humanos es el cabello. Cuando se requiere generar humanos virtuales el cabello es uno de los elementos que permiten añadir realismo a su simulación. Este proceso requiere de tres fases modelado, animación y visualización o "render". Las técnicas existentes se basan en uno o varios de estos aspectos. De esta forma algunas técnicas centran su atención en la apariencia del cabello y dejan en segundo plano la animación, lo cual da como resultado que el cabello luzca muy realista pero dichas técnicas no son adecuadas para aplicaciones en tiempo real. Por otra parte, existen otros métodos que optimizan el rubro de la animación y dejan en segundo lugar la apariencia lo cual resulta en cabello con aspecto poco realista. Se propone un sistema fácil de usar para modelado interactivo, animación y visualización de cabello en tiempo real que permite especificar características del cabello mediante imágenes a colores.

CONTENIDO

1. Antecedentes **9**

2. Planteamiento del Problema **11**

3. Estado del Arte **13**
3.1 Modelado de cabello 13
3.2 Animación del cabello 16
3.3 Visualización y apariencia del cabello 17
3.4 Propuestas comerciales 19

4. Objetivos **22**
4.1 Objetivo Principal 22
4.2 Objetivos Específicos 22

5. Modelado **23**
5.1 Imágenes que definen las características geométricas y de estilo para el cabello 23
5.1.1 Obtención de la intensidad de la imagen 24
5.2 Características geométricas 25
5.2.1 Forma de las hebras 25
5.2.2 Distribución y posición de las hebras 26
5.2.3 Longitud de las hebras 27
5.3 Características de estilo 29
5.3.1 Rizos 29
5.3.2 Trenzas 31
5.3.3 Optimización de senos y cosenos 32
5.3.4 Color 34

6. Animación de cabello **35**
6.1 Sistema de partículas 35
6.1.1 Movimiento de las partículas 35
6.1.2 Sistema de partículas básico 36
6.1.3 Fuerzas 37
6.2 Métodos numéricos 37
6.2.1 Problema de valor inicial 37
6.2.2 Soluciones numéricas 38
6.2.2.1 Método de Euler 38
6.2.2.2 Método del punto medio y Runge-Kutta 40
6.2.2.3 Método de Verlet 41

6.3 Desarrollo 42
6.3.1 Resortes y amortiguadores 43
6.3.2 Restricciones de distancia 45
6.3.3 Métodos de integración seleccionados 46
6.3.4 Detección y respuesta a las colisiones 46
6.3.5 Simplificación de cálculos 48
6.3.6 Nivel de detalle físico implícito 50

7. Visualización Y Apariencia 52
7.1 La Naturaleza anisotrópica de las superficies 52
7.2 Modelo de Phong 53
7.3 Iluminación de líneas en R^3 54
7.4 Generación visual de las líneas iluminadas 55
7.4.1 Mapeo y creación de la textura anisotrópica 56
7.4.2 Componente difusa 56
7.4.3 Componente Especular 57
7.5 Aliasing 58
7.5.1 Mejoras en la apariencia del cabello 58
7.5.2 Quincunx antialiasing 60

8. Interfaz 61
8.1 Diseño de la interfaz 62
8.1.1 Área de herramientas 62
8.1.2 Área de trabajo 65
8.1.3 Área 3D 67

9. Resultados 69
9.1 Modelado 69
9.2 Animación 71
9.3 Visualización y Apariencia 73

10. Conclusiones 75

11. Trabajo a futuro 76

12. Referencias 77

13. Apéndice A. La naturaleza del cabello 81

14. Apéndice B. Modelado de cabello en Maya 83

15. Apéndice C. Antialiasing 87

LISTA DE FIGURAS

1.1 Actores Virtuales Realistas de "Final Fantasy: Spirits Within" 9

3.1 Modelado de cabello Explícito 13
3.2 Modelado de Cabello mediante el uso de Cilindros Generalizados 14
3.3 Modelo Multiresolución de Kim *et al* 15
3.4 Modelado de Cabello mediante rastro de fluido 16
3.5 a) Distribución de Partículas convencional b) Distribución propuesta por Bando *et al* 16

5.1 Modelo de una Hebra de cabello 25
5.2 Mapa de distribución y posición de las hebras del cabello 26
5.3 Mapa de Longitudes 28
5.4 a) Rizo individual enmarcado en el círculo, b) Cabello rizado 29
5.5 Suma de señales 30
5.6 Distintos niveles de rizados. De izquierda a derecha el nivel de rizado aumenta 30
5.7 Estilos de trenzas. a) Trenza Triple, b) Trenza de Corona, c) Trenza Arwen 31
5.8 Trenza simplificada 32
5.9 Decimación. x[n] es la secuencia original, $x_p[n]$ ilustra las muestras de x[n] que integran a $x_b[n]$. Note que $x_b[n]=x_p[nN]$ 33
5.10 Cabello Colorido 34

6.1 a) Estructura de una partícula b) Sistema de Partículas 36
6.2 Inexactitud del Método de Euler. Distintos pasos de tiempo son representados 39
6.3 Representación de una hebra de cabello con resortes y amortiguadores 44
6.4 a) Fuerza de Gravedad sin fuerza del viento b) Fuerza del Viento con la misma dirección que la Fuerza de Gravedad. Se observa que en b) el cabello se estiró haciéndolo parecer más largo que en a) aunque es la misma longitud 44
6.5 Restricción de distancia. a) Separando las partículas b)Juntando las partículas 45
6.6 Uso del radio vector para transformar una esfera en elipsoide 47
6.7 Hebras Fundamentales y Hebras Interpoladas 49
6.8 Comparación entre a) Cabello sin interpolar (5000 hebras fundamentales) y b) Cabello interpolado (5000 hebras en total, 500 hebras fundamentales y 4500 hebras interpoladas 49
6.9 Comparación entre a) Cabello con 5000 hebras base y b) Cabello con 5000 hebras en total, 500 hebras base y 4500 hebras asociadas 50

7.1 Ejemplo de un disco matizado con reflexión isotrópica (izquierda), y reflexión anisotrópica (derecha) 53
7.2 Vectores para el Modelo de Phong 53

7.3 Para las líneas, hay una infinidad de posibles vectores ***R*** que yacen en el cono alrededor de ***T***. Para calcular la iluminación se elige aquel que está en el plano ***LT*** 54
7.4 Proyección del vector ***L*** 55
7.5 Mapa de Textura Bidimensional para simular la reflexión anisotrópica del cabello con k_a=0.1, k_d=0.3, k_s=0.6 y n=40 58
7.6 Línea con aliasing 58
7.7 Peinado dibujado con polilíneas de un píxel de ancho 59
7.8 Método para plasmar el grosor característico de las hebras de cabello 59

8.1 Área de Herramientas 63
8.2 Área de Trabajo 65
8.3 Área de Visualización 3D 67
8.4 Modelador de cabello 68

9.1 Peinado Típico con cabello corto 70
9.2 a) Mapa de densidad, b) Mapa de longitud, c) Mapa de color, d) Mapa de nivel de rizado 70
9.3 Modelo con cejas, bigote y barba 70
9.4 Variedad de peinados 71
9.5 Trenza 71
9.6 a) 5000 hebras base, 0 hebras asociadas. b) 500 hebras base, 4500 hebras asociadas. c) 14000 hebras base, 0 hebras asociadas. d) 1000 hebras base, 13000 hebras asociadas 73
9.7 Cabello a) sin reflexión anisotrópica. b) con reflexión anisotrópica 74
9.8 Aliasing y antialiasing. Cabello a) sin aliasing. b) con componente alpha. c) con Quincunx Antialiasing d) con componente alpha y Quincunx antialiasing 74

14.1 Modelo a utilizar 83
14.2 El cráneo conformado con NURBS 83
14.3 Base de la cola de caballo 84
14.4 Curvas guía (en verde) para el cabello 84
14.5 Resultado de dibujar sobre la parte superior de la cabeza 85
14.6 Cabello siguiendo la dirección de las curvas de control 85
14.7 Curvas de control para la cola de caballo 86
14.8 La cola de caballo creada con curvas de PaintFX 86
14.9 Peinado final con cola de caballo 86

15.1 Patrones de muestreo 88

NOTA. La figura 7.2 Vectores para el Modelo de Phong, es contribución del Dr. Bedrich Benes. Las figuras 7.1, 7.3, 7.4 y 7.5 fueron extraídas de la referencia [21][22].

LISTA DE TABLAS

5.1 Asignación de colores 23
5.2 Decimación 33

9.1 Rendimiento entre amortiguadores – resortes y restricción de distancia 72
9.2 Rendimiento de la Animación en Sistema 1 72
9.3 Rendimiento de la Animación en Sistema 2 73

1. ANTECEDENTES

Las aplicaciones en tiempo real como juegos de video, tutoriales virtuales y cualquier simulador en el que el usuario pueda participar interactivamente, se han convertido en un campo importante dentro del cómputo debido a tres aspectos fundamentales, primeramente, permiten la interacción con el usuario, además proporcionan resultados en forma inmediata y finalmente hoy en día, dichas aplicaciones pueden ser ejecutadas en computadoras personales convencionales.

Este tipo de aplicaciones requieren de tres elementos que mejoran la interacción con el usuario: ambientes gráficos en los que el usuario se pueda desplazar, personajes (humanos virtuales) con los que el usuario pueda interactuar y cierto nivel de realismo que sea plasmado en los dos aspectos anteriores para mejorar la sensación de inmersión.

Uno de los logros más significativos que incorporan en forma destacada estos tres elementos, aunque es necesario mencionar que no son aplicaciones en tiempo real, lo podemos encontrar en la película *Final Fantasy: Spirits Within* de Square Pictures (Figura 1.1) y en un cortometraje de *The Animatrix, Final Flight of the Osiris*, los cuales muestran las características propias del ser humano como la piel, los gestos, los ojos, los dientes, la ropa y el cabello a tal nivel de realismo que en algunas ocasiones sería difícil diferenciarlas de sus contrapartes reales. Dentro de las características del ser humano mencionadas anteriormente, aquella que necesitó más tiempo de cómputo, en la película *Final Fantasy: Spirits Within*, fue la técnica encargada de generar el cabello de los personajes haciendo imposible adaptarla a aplicaciones en tiempo real, por tal motivo es interesante y necesario el desarrollo de otra técnica que sea ideal para aplicaciones en tiempo real; no olvidemos que el desarrollo de una técnica para la simulación de cabello tiene bastante reto, ya que estamos muy familiarizados con su aspecto y la forma en que se mueve y por tanto podemos inferir fácilmente si la simulación es convincente o no.

Fig. 1.1 Actores Virtuales Realistas de
"Final Fantasy: Spirits Within"

Para la simulación de cabello es necesario tener en cuenta tres aspectos fundamentales en el área de los gráficos por computadora, el modelado, la animación y la visualización o "render", cada uno de estos aspectos permiten incorporar las características naturales del cabello a la simulación, de esta forma el modelado permite representar la forma, distribución y estilo (peinados) del cabello, mediante la animación se representa su movimiento así como las interacciones resultantes de este movimiento y la visualización permite representar el cabello lo más realista posible.

2. PLANTEAMIENTO DEL PROBLEMA

La adición de cabello en humanos virtuales presenta diversos retos derivados a su naturaleza y al momento de ser modelado, animado y generado visualmente.

Los retos más interesantes que se presentan derivados a la naturaleza del cabello son:

- Un gran número de hebras que lo conforman. Típicamente un cuero cabelludo saludable tiene de 100 000 a 150 000 hebras, por tal motivo aunque este número se simplifique, en la simulación se requerirá procesar una gran cantidad de cálculos.
- Color. Hebras de cabello de diferentes intensidades reflejan, refractan y absorben los rayos de luz en diferentes grados.
- Propiedades físicas como su forma (rizado, lacio, chino, quebrado) y su rigidez varían su movimiento.
- Estamos tan familiarizados a la forma en como debe lucir y moverse el cabello que es muy fácil para una persona común establecer que tan buena o mala es la simulación de cabello sin importar que tan compleja sea.

En relación al modelado, los principales retos que se presentan se deben a:

- Su representación. Esto es, seleccionar algún método adecuado para representar el cabello sin alterar su apariencia final en la computadora mediante primitivas de dibujo.
- Diseño de la herramienta de modelado, para permitir al usuario recrear diversos estilos de peinados y definir propiedades físicas del cabello de una forma fácil, rápida e intuitiva.

Los retos en relación a la animación se deben especialmente a:

- Su movimiento. Se debe seleccionar una técnica adecuada de animación que nos permita simular este comportamiento lo más natural posible, además se debe simplificar la técnica seleccionada para reducir el cálculo de las operaciones matemáticas y poder incorporar la simulación a aplicaciones en tiempo real.
- Interacción con otros cuerpos. Esta interacción permite que el cabello no traspase los límites establecidos por partes del personaje virtual como cabeza y hombros, de la misma forma que en el punto anterior, se deben realizar simplificaciones para reducir el número de cálculos.
- Respuesta a fuerzas externas. En la vida real las fuerzas como la gravedad y el viento complementan el aspecto general del cabello por tal motivo es necesario incorporarlas en la animación.

Con respecto a la visualización y apariencia los retos que surgen son los siguientes:

- La gran cantidad de hebras de cabello. Aunque la animación se haya optimizado, es necesario simplificar el dibujado de las hebras sin alterar en gran medida la apariencia final del cabello.
- Interacciones de cada hebra de cabello con la luz. Estas interacciones se deben plasmar en la simulación para poder representar la naturaleza anisotrópica del cabello además de generar sombras entre hebras de cabello.
- En general la visualización del cabello sufre del problema de *aliasing* lo cual provoca que se obtengan representaciones dentadas del cabello reduciendo en gran medida su realismo.

3. ESTADO DEL ARTE

Inicialmente los métodos o técnicas desarrolladas para la simulación de cabello se basaban específicamente en el modelado, en la animación o en la visualización del cabello. Algunas de estas técnicas centran su atención en la apariencia del cabello y dejan en segundo plano la animación, lo cual da como resultado que el cabello luzca muy realista, pero dichas técnicas no son adecuadas para aplicaciones en tiempo real. Por otra parte, existen otros métodos que optimizan el rubro de la animación y toman en cuenta el modelado pero dejan el rubro de la apariencia en segundo plano, lo cual produce cabello con aspecto poco realista.

A continuación se presenta una descripción por área de interés de las técnicas desarrolladas para la simulación de cabello.

3.1 MODELADO DE CABELLO

Gracias a las propiedades físicas del cabello así como a interacciones del tipo cabello – objeto (como la cabeza, peines, aerosoles, etc.) e interacciones entre hebras del cabello, el ser humano es capaz de realizar peinados intrincados. Para modelar estos comportamientos se recurre a *modelos de cabello explícitos.* En estos modelos, cada hebra es modelada con base a su forma y su movimiento reales. Daldegan *et al* [1] propone una técnica en la cual el usuario define en forma interactiva en tres dimensiones, unas cuantas características de cada hebra y después completa el peinado basado en dichas características. Los parámetros que se especifican para el cabello son la densidad, la distribución y la orientación (Figura 3.1).

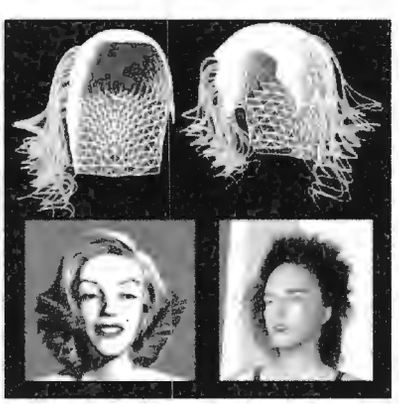

Fig. 3.1 Modelado de cabello Explícito

Los modelos explícitos del cabello son muy intuitivos y realistas. Desafortunadamente la forma de generar el peinado es muy tediosa y presenta dos dificultades, primeramente, toma de cinco a diez horas modelar un peinado complejo usando el método descrito por Daldegan *et al* [1], en segundo, resulta muy costoso aplicar movimiento a estas hebras debido a la gran cantidad de ellas. Estas dificultades pueden ser parcialmente reducidas si se agrupan las hebras. Naturalmente esta idea es válida ya que las fuerzas de adhesión y de cohesión unen a las hebras en mechones.

Existen técnicas que explotan la característica de las hebras del cabello a agruparse en mechones debido a la electricidad estática. Watanabe [2, 3] y Yan *et al* [4] desarrollaron varias técnicas para el modelado de mechones. La técnica que propuso Yan consistía en modelar cada mechón mediante el uso de *cilindros generalizados* (los cilindros generalizados "CG" son un grupo especial de cilindros los cuales pueden adquirir cualquier forma curveada) además usaba texturas volumétricas o texturas tridimensionales para que dichos cilindros lucieran como lucen los mechones reales (Figura 3.2).

Fig. 3.2 Modelado de Cabello mediante el uso de CG

El método de Csuri *et al* [5], consistía en generar volúmenes que se parecieran al pelaje mediante polígonos. Cada hebra era modelada como una serie individual de triángulos colocada sobre la superficie y visualizada usando el algoritmo para la eliminación de superficies no visibles. Miller [6] produjo mejores resultados al modelar las hebras como pirámides formadas de triángulos. Estas técnicas tenían serios problemas debido a que se requerían un número considerable de primitivas para representar el cabello y la longitud de sus hebras.

Kim y Neumann [7] desarrollaron una técnica que reducía notablemente el número de primitivas para representar el cabello mediante un sistema multiresolución que permitía modelar el cabello. Este sistema permite modelar cada mechón jerárquicamente. De esta manera el usuario modela un peinado general mediante mechones (representado por cilindros generalizados) y después realiza cambios en forma local en cada mechón (los cuales son subdivididos por el sistema) para incrementar la complejidad del peinado (Figura 3.3).

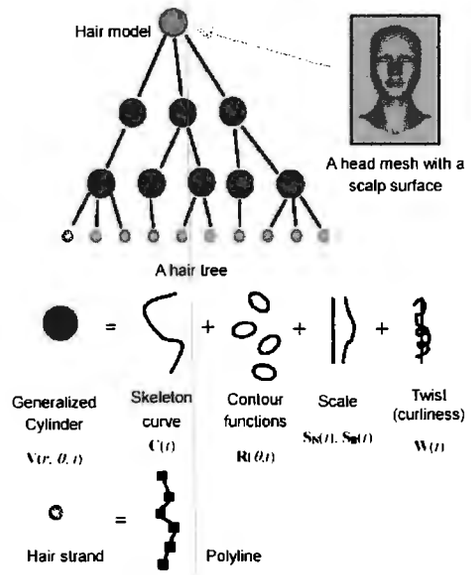

Fig. 3.3 Modelo Multiresolución de Kim *et al*

Lograr una definición geométrica precisa del cabello es bastante difícil, en muchas ocasiones no se puede inferir dónde empieza y dónde termina una hebra o dónde empieza y dónde termina un mechón, algo similar ocurre con su color. Para plasmar este comportamiento, Perlin *et al* [8] desarrolló las *hipertexturas* o texturas volumétricas, las cuales se usan para modelar pelaje. En este esquema, el pelaje es modelado como una intricada variación de densidades en un espacio tridimensional sin necesidad de definirlo geométricamente. Básicamente el modelo es una extensión de texturas 2D definidas analíticamente con la diferencia de que la hipertextura es evaluada en una región en vez de una superficie. Perlin demostró que mediante funciones analíticas simples se podía definir pelaje sobre un toroide o sobre una esfera, sin embargo las hipertexturas sólo pueden ser aplicadas a figuras geometrías definidas analíticamente. Kajiya *et al* [9] extendió este esquema para aplicar las hipertexturas a figuras geométricas complejas. Para lograrlo diseñó lo que se denomina *texel*. Un *texel* (abreviatura de texel element o elemento de textura) es un bloque sólido de hipertextura que es colocado sobre las figuras geométricas de forma similar a como se colocan tejas en una construcción, de esta forma se pueden recrear animales cuyo pelaje sea abundante.

Un método propuesto por Guang y Zhiyong [10] consiste en modelar cabello corto. En esta técnica el usuario define la zona donde crecerá el cabello en una malla que representa la cabeza, posteriormente los vértices del peinado son obtenidos automáticamente, un conjunto de nuevos vértices son generados sumando un vector *offset* a los vértices generados automáticamente. La longitud y la dirección del cabello es controlada con este vector *offset* el cual será diferente para cada vértice del peinado. Una serie de triángulos son generados conectando los vértices generados automáticamente con los nuevos vértices. Finalmente a los triángulos se les aplica una textura 2D con canal alpha para mejorar el efecto visual.

Como se mencionó anteriormente, una de las ventajas que tiene el modelo explícito del cabello es que es intuitivo y el agrupar las hebras en mechones permite controlar

globalmente la forma del cabello. Por el contrario, con las texturas volumétricas o texturas tridimensionales se puede recrear la naturaleza compleja del cabello y mostrar muchos detalles visuales. Thalmann [11] propone una técnica la cual explota las ventajas de los esquemas anteriores. En este método el cabello se modela como el rastro que deja un fluido. Para simplificar dicha representación, el rastro o flujo no presenta fuerzas de fricción. Para diseñar un peinado, el usuario establece algunas características, relacionados al fluido, alrededor del cuerpo geométrico (Figura 3.4).

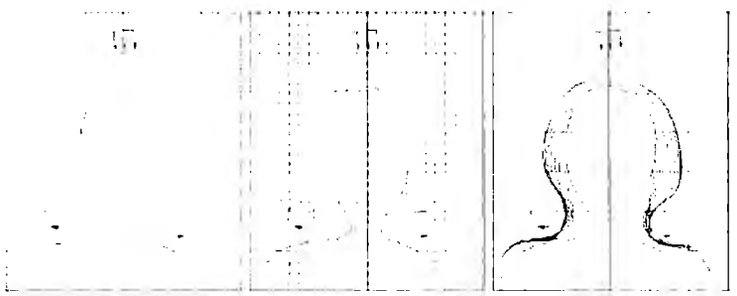

Fig. 3.4 Modelado de Cabello mediante rastro de fluido

3.2 ANIMACIÓN DEL CABELLO

Anjyo *et al* [12], Rosenblum *et al* [13] y Kurihara *et al* [14] diseñaron modelos dinámicos para la animación del cabello. Cada hebra es modelada como un segmento de conexiones rígidas que se pueden doblar en cada articulación. De este modo las ecuaciones relacionadas con el movimiento de la hebra son resueltas para obtener la animación y las colisiones con el cuerpo. Estos esquemas permiten la animación del cabello en tiempo real, pero si se desea plasmar las interacciones hebra – hebra resulta computacionalmente costoso.

Bando *et al* [15] presentó un método para animar cabello usando partículas con la diferencia de que estas, no están conectadas en forma estructurada, si no que están esparcidas a lo largo del modelo del cabello, para obtener una versión discretizada del volumen que ocupa el cabello. De esta forma las fuerzas externas son calculadas en una partícula y sus respectivos vecinos que están dentro de un radio específico (Figura 3.5).

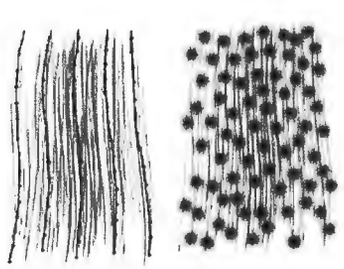

Fig. 3.5 a) Distribución de Partículas convencional
b) Distribución propuesta por Bando *et al*

Bertails *et al* [16] adaptó el método propuesto por Kim y Neuman para animar cabello. Este método consiste en animar cabello mediante una jerarquía adaptable que controla el movimiento de los mechones. Esta técnica permite representar fenómenos relacionados al movimiento del cabello, como por ejemplo la disociación de los nodos que forman las hebras que componen un mechón de acuerdo a su posición, es decir, aquellos nodos que estén en la punta tendrán mayor tendencia a separarse entre si que aquellos que están más cerca de la raíz.

En el caso de pelaje, comúnmente modelado con texturas tridimensionales, no se pueden incorporar los esquemas anteriores para representar su movimiento. Generalmente para animar el pelaje se recurre a funciones de densidad de volumen que varían a través del tiempo. De esta forma se pueden simular efectos del aire usando perturbación estocástica del espacio como ruido, movimiento Browniano, turbulencia, etc.

3.3 VISUALIZACIÓN Y APARIENCIA DEL CABELLO

El cabello presenta problemas relacionados a su visualización y apariencia. Las razones de estos problemas se deben a lo siguiente: gran cantidad de hebras, necesidad de una geometría detallada e interacciones complejas de las hebras con la luz. Agregado a lo anterior, la visualización del cabello sufre del problema de *aliasing* lo cual provoca que se obtengan representaciones dentadas del cabello reduciendo en gran medida su realismo.

Durante la década pasada, la apariencia del cabello ha sido estudiada por varios investigadores. En algunos casos se han logrado resultados exitosos. Sin embargo, estos resultados son adecuados en condiciones particulares y ofrecen capacidades muy limitadas o ninguna al momento de animar el cabello.

Reeves *et al* [17] propuso el uso de sistemas de partículas para modelar objetos borrosos o difusos como el fuego. A pesar del tamaño reducido de las partículas, que en algunas ocasiones pueden ser más pequeñas que un píxel, estas se manifiestan por sí mismas por la forma en que reflejan la luz, provocan sombras y obstruyen objetos. Por lo tanto, la estructura de la partícula requiere ser representada solo por un modelo que pueda recrear estas propiedades. Un sistema de partículas es visualizado mediante el dibujado de cada una de ellas en el *frame buffer*, de esta forma para obtener el color final del píxel se calcula la contribución de cada partícula. Esta técnica ha sido usada exitosamente para recrear objetos borrosos o difusos como nubes, fuego, humo, cabello, etc. además de haber sido incorporada en muchos sistemas de animación comerciales. Sin embargo, esta técnica tiene limitaciones relacionadas con el cálculo de sombras debido a que modelos estocásticos básicos no son adecuados para representar el tipo de orden y orientación del cabello. También requiere de un modelo apropiado para controlar la longitud y orientación del cabello.

Con relación a la visualización y apariencia del pelaje y retomando el esquema de Kajiya, un *texel* es un mapa de textura tridimensional donde los parámetros relacionados a la superficie y al modelo de iluminación están incorporados a un volumen. Los *texeles* son un tipo de modelo intermedio entre una textura y una geometría, sin embargo no están totalmente ligados a la geometría de una superficie en particular lo cual permite que el tiempo que se requiere para la visualización sea independiente de la complejidad geométrica de la superficie. Los *texeles* son generalmente recreados mediante la técnica denominada emisión de rayos (*ray casting*).

Goldman [18] propone un esquema en el cual se visualiza el cabello como si fuera pelaje en situaciones especificas como aquellas en las que la geometría del modelo del cabello no se aprecia apropiadamente debido a la resolución de la imagen. Goldman propone un algoritmo probabilístico, también conocido como algoritmo de pelaje falso. En este modelo, la luz reflejada por las hebras y por la piel que esta debajo de estas es mezclada mediante el uso de la probabilidad de que un rayo golpee una hebra en cierta área lo cual permite calcular el factor de opacidad.

Aunque las texturas tridimensionales son ideales para recrear pelaje y cabello corto, no lo son tanto si se trata de cabello largo.

Un método de fuerza bruta para la visualización y apariencia del cabello consiste en modelar cada hebra como un cilindro curveado y visualizar cada cilindro (primitiva). El problema radica por la gran cantidad de cilindros que se requieren para una representación adecuada del cabello. Sin embargo, este método se ha llevado a cabo pero con diferentes primitivas.

En otro esquema, se usó el algoritmo para la eliminación de superficies no visibles en hardware con matizado Gouraud para la visualización del cabello que era modelado mediante segmentos conectados de prismas triangulares. Sin embargo el modelo de iluminación era muy simple y no solucionaba el problema del aliasing. LeBlanc *et al* [19] propuso otro esquema hacía uso de búferes sombra y mezclado de píxeles. Esta técnica ha sido una de las más efectivas en términos prácticos. A pesar de que podía ser aplicada sobre una variedad de objetos con cabello y con pelaje, uno de las primeras intenciones de LeBlanc fue recrear distintos estilos de cabello y peinados en forma realista. De este modo la visualización del cabello es realizada siguiendo los siguientes pasos [20]. Primero la sombra de la escena es calculada para cada fuente de luz. Después, el buffer sombra para el cabello es calculado para cada fuente de luz (dicho buffer depende del tipo de peinado), esto se realiza mediante el dibujado de cada segmento del cabello en el buffer Z o buffer de profundidad y posteriormente se extrae un mapa de profundidad. El mapa de profundidad del buffer sombra de la escena y del cabello son combinados para formar un buffer sombra para cada fuente de luz. La escena con su buffer Z respectivo es generada a partir del modelo de la escena y de los buffer sombra combinados. Finalmente los segmentos del cabello son dibujados en la escena como polilíneas iluminadas [21, 22] además para determinar la visibilidad de las polilíneas se usa el buffer Z de la escena y para determinar las sombras del cabello se usan los buffer sombra combinados.

También es interesante agregar efectos especiales al cabello, como por ejemplo, que el cabello luzca mojado. Para lograr este tipo de efectos es necesario cambiar el modelo de matizado. Bruderlin [23] mostró métodos sencillos para plasmar una apariencia mojada del cabello como por ejemplo variar la contribución especular de la luz, es decir, las hebras que pertenecen a un mechón que están enfrente de la fuente de luz serán más brillantes que aquellas que no lo están.

Kong y Nakajima *et al* [24] presentó un método que usaba un buffer de volumen visible para reducir el tiempo que se requiere para la visualización. El buffer de volumen visible es un espacio cúbico definido por el usuario de acuerdo a la memoria disponible del sistema y de la resolución necesaria. Ellos consideraron el modelo del cabello como una combinación entre una representación tosca y una representación detallada del cabello, las transiciones entre estas representaciones dependen del valor de opacidad o del punto de vista del espectador. Aunque reduce considerablemente el tiempo que se requiere para la visualización, los resultados visuales no son muy realistas.

Yan *et al* [4] aplicaron texturas tridimensionales a cilindros generalizados que representaban el cabello. Se usó trazado de rayos (*ray tracing*) para obtener los límites de los cilindros generalizados para capturar las características de la función de densidad necesaria para generar las texturas tridimensionales.

Recientemente Marschner *et al* [25] extendieron el modelo de Kajiya y Kay haciéndolo mas preciso mediante la adición de nuevas mediciones relacionadas con la transmisión y reflexión interna de la luz en cabello que es modelado con cilindros elípticos transparentes.

3.4 PROPUESTAS COMERCIALES

En general con cualquier software de modelado por computadora se puede modelar, animar y visualizar cabello sin embargo este proceso es muy tedioso (Apéndice B) por otra parte ha sido tanto la importancia que se le ha dado a la incorporación de cabello o pelaje a personajes virtuales que empresas desarrolladoras de software han diseñado funciones especificas en sus programas o han creado extensiones para lograrlo. Estas adiciones permiten al diseñador agregar cabello o pelaje, modelar peinados, animarlo y dar una apariencia realista aunque el cabello, su comportamiento y apariencia resultante no se genera en tiempo real.

Dentro de las alternativas comerciales tenemos por ejemplo la extensión para Maya "Shave and Haircut" de Joseph Alter Inc. [26]. "Shave and Haircut" incorpora cuatro grandes funciones visualización y apariencia, peinado, dinámica y texturizado. Para la visualización y apariencia, "Shave and Haircut" usa el motor de generación y visualización de imágenes de Maya que está basado en trazado de rayos técnica que permite generar

reflexiones, refracciones y sombras en el cabello y/o en el pelaje. Además puede realizar post procesos para agregar efectos como campo de profundidad y neblina.

En cuanto al peinado, "Shave and Haircut" incorpora:

- Parámetros de desplazamiento para modificar la geometría de las hebras del cabello o pelaje para modelar estilos como rizado, lacio, obtener un aspecto húmedo, etc.
- Campo de guías que definen la forma general del cabello o pelaje.
- Ecualizador de densidad que permite modificar la distribución de las hebras del cabello o pelaje.
- Interfaz para peinar la cual permite realizar transformaciones sobre el campo de guías.
- Una herramienta para cepillar permite peinar el cabello.

En cuanto al apartado de dinámica, en "Shave and Haircut" las guías pueden ser seleccionadas, desplazadas o movidas sin necesidad de usar menús. Además la física de las guías se modifica de acuerdo al cuerpo en el que están colocadas y agrega detección y respuesta a las colisiones, también toma los campos de fuerza de Maya para poder adicionar fuerzas externas. Finalmente permite agregar restricciones de movimiento para poder modelar colas de caballo.

Otra alternativa es Maya "Fur" [27] de Alias WaveFront sin embargo esta sólo disponible en la versión Maya "Unlimited". Maya "Fur" permite al diseñador modelar y generar pelaje y sólo cabello corto. Maya "Fur" incluye:

- Interfaz de usuario con Maya "Artisan". Esta interfaz permite pintar sobre NURBS o superficies poligonales atributos del pelaje como calvicie, longitud y rizado. Crear mechones y variar su nivel de agrupamiento, frecuencia y forma pintando o mapeando estos atributos para lograr un pelaje con aspecto mojado, sin brillo o sucio.
- Tipos de Maya "Fur". Una superficie puede tener diferentes tipos de pelaje, por ejemplo la superficie de una cabeza puede tener bello corporal, cejas, bigote o barba. Atributos del pelaje como grosor, longitud, opacidad, rizado y dirección pueden ser configurados globalmente o especificados localmente. Los colores de la base y punta del pelaje pueden ser definidos mediante texturas procedurales o texturas almacenadas en archivos para recrear efectos como las manchas de un leopardo o rayas de una cebra.
- Preconfiguraciones. Existen 10 descripciones de pelaje preconfiguradas como el de ardilla, oso y oveja, además se pueden mezclar entre sí.
- Animación. Un sistema de atracción permite que el movimiento del pelaje este basado en cuadros llave o reaccione a fuerzas dinámicamente.

- Visualización y apariencia. Maya "Fur" está integrado con Maya "Renderer", además permite sombreado entre pelaje, "motion blur" y campo de profundidad.

Poser [28] de Curious Labs es un software especializado para modelado y animación de personajes. Incorpora herramientas que permiten modificar el aspecto físico de cada una de las partes del personaje, agregar ropa y para agregar cabello proporciona una herramienta llamada "Hair Room".

"Hair Room" de Poser permite especificar vértices objetivo del cabello para determinar como crece. Una vez que el cabello está colocado se peina y se le aplica movimiento con fuerzas como gravedad y viento. "Hair Room" Incorpora las siguientes herramientas:

- Controles de Crecimiento de Cabello. Estos controles permiten crear, editar y borrar grupos de crecimiento de cabello.
- Controles de Crecimiento. Estos controles permiten especificar la forma en que el cabello crece en los grupos.
- Controles de Peinado. Los controles de peinado permiten modificar la apariencia del cabello.
- Controles de Física. Estos permiten controlar la forma en que el cabello se comportará durante la animación, proporciona funciones para agregar colisiones entre el cabello y el cuerpo, fuerzas de gravedad, de viento, de resortes, entre otras.
- Propiedades del Cabello. Estas opciones permiten especificar las propiedades visuales para generar una apariencia realista del cabello.

La última alternativa que se presenta es la extensión para LightWave 3D llamada Sasquatch de Warley Laboratories [29]. Sasquatch integra su propio motor para generar cabello fotorealista. Adiciona controles para modificar estilo, color y matizado de cabello o pelaje y proporciona herramientas para peinar y animar cabello y pelaje con características similares a las alternativas anteriormente descritas.

4. OBJETIVOS

OBJETIVO PRINCIPAL

Diseñar una herramienta fácil de usar e intuitiva para modelado interactivo, animación y visualización de cabello que permita realizar simulaciones de cabello para ser incorporadas a personajes realistas en aplicaciones en tiempo real que puedan ser ejecutadas en computadoras y hardware gráfico de uso común.

OBJETIVOS ESPECÍFICOS

Para que la herramienta sea la adecuada se debe considerar lo siguiente:

- Modelado sencillo, es decir, proporcionar una interfaz con opciones intuitivas para que el usuario pueda definir características geométricas y físicas del cabello así como una variedad de estilos y peinados. Además permita agregar movimiento a su modelo de forma automática.

- Animación natural, que permita al usuario apreciar la forma en que su modelo lucirá cuando se encuentre en movimiento.

- Práctico, en este aspecto se busca que la herramienta proporcione una interfaz genérica que pueda utilizarse para agregar cabello y pelaje a cualquier tipo de personajes y objetos.

5. MODELADO

Para modelar las formas que las hebras de cabello pueden adquirir se realizó una clasificación con base a su geometría y a su estilo. Con respecto a su geometría se determina posición, distribución y longitud. Con respecto a su estilo se define la forma ya sea lacia, rizada y trenzas, así como su color. Para recrear el peinado final éstas características son especificadas por el usuario por medio de imágenes a color en una interfaz de pintado.

5.1 IMÁGENES QUE DEFINEN LAS CARACTERÍSTICAS GEOMÉTRICAS Y DE ESTILO PARA EL CABELLO

La definición de las características geométricas y de estilo se realiza mediante imágenes a colores las cuales son generadas por el usuario con una interfaz de pintado integrada al sistema (capítulo 8). Esto permite que el usuario genere rápidamente peinados y debido a que es una interfaz de pintado común, el usuario tardará muy poco tiempo en aprender su manejo.

La forma en que son interpretadas estas imágenes es por medio de la intensidad de su color. Estas imágenes tienen componentes RGB (por su abreviatura en inglés Red, Green, Blue), por lo tanto se toma cada una de estas componentes y sus respectivas combinaciones para definir cada una de las características geométricas y de estilo del cabello, en la Tabla 5.1 se muestra esta relación.

Tabla 5.1. Asignación de colores

Color	Característica
Verde	Posición y densidad
Azul	Longitud
Rojo	Nivel de Rizado
Magenta	Trenzas
Combinación de verde, rojo y azul	Color

La intensidad de la componente es utilizada para representar valores máximos y valores mínimos en relación a cada una de las características ya sea geométrica o de estilo.

Por ejemplo cuando la imagen con componente roja está en su valor máximo se obtendrán hebras de cabello chinas pero en zonas guindas se obtendrán cabellos rizados.

Para relacionar estas imágenes con el modelo tridimensional al cual se le va a implantar cabello y con el peinado en sí, es necesario que el modelo tenga especificadas coordenadas de textura. Posteriormente con base a esas coordenadas de textura se obtiene la intensidad de color de la imagen correspondiente.

5.1.1 OBTENCIÓN DE LA INTENSIDAD DE LA IMAGEN

Antes de poder usar la imagen es necesario almacenarla en un arreglo, por otra parte aunque se está tratando con imágenes a color, sólo se usa una componente de color por cada característica del cabello, excepto para el color del cabello y para las trenzas (para las trenzas se está utilizando el color magenta el cual es una combinación con una misma contribución de color rojo y azul), por tal motivo y para ahorrar espacio en memoria se usan arreglos que almacenan los valores de la componente de color que varía, las otras componentes son descartadas. Estos arreglos almacenan valores numéricos entre 0 a 255 donde 0 indica la intensidad mínima y 255 la intensidad máxima, sin embargo se pueden especificar otros valores que varíen por ejemplo entre 0 y 10 haciendo la normalización correspondiente.

Como se había mencionado anteriormente, es necesario conocer las coordenadas de textura del modelo tridimensional para saber la intensidad de color que le corresponde a cada una de sus caras. Estas coordenadas de textura <u, v> varían entre 0 y 1, por otro lado el tamaño de las imágenes que son utilizadas es de 512 x 512 píxeles por tal motivo no se puede hacer una correspondencia directa entre las coordenadas de textura y las coordenadas de la imagen. Para lograr esta correspondencia es necesario hacer una transformación para obtener la intensidad correcta. Dicha transformación se presenta en el siguiente algoritmo.

```
1. Obtener intensidad de color (decimal u, v){
2.     decimal mapCoord[2], fracMapCoords[2], intensidad
3.     entero coords[2][2]
4.     mapCoords[0] = u * (imagen.ancho-1)
5.     mapCoords[1] = v * (imagen.largo-1)
6.     coords[0][0]= mapCoords[1] MOD height * width + mapCoords[0] MOD width
7.     coords[0][1]= mapCoords[1] MOD height * width + (mapCoords[0]+1) MOD width
8.     coords[1][0]= (mapCoords[1]+1) MOD height * width + mapCoords[0] MOD width
9.     coords[1][1]= (mapCoords[1]+1) MOD height * width + (mapCoords[0]+1) MOD width
10.    i = map[coords[0][0]]+map[coords[0][1]]+map[coords[1][0]]+map[coords[1][1]]
11.    i = result * maxIntensidad / 255
12.    regresa i;
13.    }
```

El algoritmo tiene como entrada los parámetros <u, v> los cuales son las coordenadas de textura del modelo, en las líneas 4 y 5 se calculan las coordenadas de la imagen que corresponden a las coordenadas <u, v>.

Es preciso recordar que la imagen se almacena en un arreglo unidimensional cuyo tamaño se obtiene al multiplicar el ancho de la imagen por su largo. Por ejemplo si tenemos una imagen de 32 píxeles de ancho por 32 píxeles de largo el tamaño del arreglo será de 1024 casillas, de esta forma en la casilla cero del arreglo se almacena el valor del primer píxel de la imagen cuya coordenada es (1, 1) y en la casilla 1024 se almacena el valor del último píxel cuya coordenada es (32, 32). Esta relación entre las coordenadas de los píxeles de la imagen y su casilla correspondiente en el arreglo se determina en las líneas 6, 7, 8 y 9. En la línea 10 se obtiene la intensidad de color final, y en 11, que puede ser opcional, se modifica dicha intensidad para especificarla en otra escala.

Este algoritmo es utilizado en todas las fases como un proceso previo para obtener las intensidades adecuadas de color ya sea para especificar las características geométricas o las de estilo.

5.2 CARACTERÍSTICAS GEOMÉTRICAS

5.2.1 FORMA DE LAS HEBRAS

Un cabello humano es, geométricamente, un cilindro largo, delgado y curveado. Típicamente un humano tiene alrededor de 100 000 a 150 000 cabellos; cada cabello puede tener un grosor que varía entre 0.03 mm a 0.09 mm, además de ser más quebrado en unas zonas que en otras, es decir, puede ser perfectamente lacio en unas o un tanto rizado en otras.

Por lo anterior para modelar las hebras de cabello se ha optado por representarlas de una forma explícita, es decir una representación en la cual cada hebra de cabello es modelada con base a su forma y movimiento reales, ya que representaciones volumétricas como el uso de texturas tridimensionales [9] no son adecuadas para modelar cabello largo o representaciones utilizadas en [7] terminan por utilizar líneas para generar el peinado final. Por otra parte para modelar explícitamente la forma del cabello se pueden usar curvas NURBS, superficies NURBS o polilíneas. El uso de curvas NURBS permite obtener líneas curvas suaves sin embargo el tiempo de procesamiento es mayor que usar polilíneas [30], lo anterior reduciría el rendimiento de la animación, por otra parte se puede suavizar una polilínea usando algún método de interpolación entre sus nodos. Por lo anterior se eligieron las polilíneas para modelar las hebras del cabello.

El modelo final de la hebra de cabello que se propone consta de dos tipos de polilínea una polilínea esqueleto y una polilínea gráfica (Figura 5.1).

Fig. 5.1 Modelo de una Hebra de cabello

Las polilíneas esqueleto tienen las siguientes propiedades:

1. Están conformadas por 4 o 5 nodos de acuerdo al nivel detalle en la animación que se desee obtener.
2. Son la representación física de las hebras del cabello, es decir, las ecuaciones de movimiento son calculadas sobre sus nodos.
3. Sirve como trayectoria guía a las polilíneas gráficas.
4. No son dibujadas al momento de generar el modelo final del peinado.

Las propiedades de las polilíneas gráficas son las siguientes:

1. Tienen entre 10 a 30 nodos, lo cual permite obtener curvas suaves para representar diferentes niveles de rizado del cabello
2. Son la representación visual del cabello, es decir, estas polilíneas son dibujadas al momento de generar el peinado final.

El motivo principal por la que se optó por esta representación dual es que permite la reducción de cálculos en relación a la animación y evita usar espacio extra en memoria ya que la información de las polilíneas gráficas no es almacenada debido a que es derivada de la información de las polilíneas esqueleto para ser inmediatamente dibujada.

5.2.2 DISTRIBUCIÓN Y POSICIÓN DE LAS HEBRAS

Los folículos de cada persona están posicionados en forma aleatoria además existen zonas en el cuero cabelludo más pobladas de hebras de cabello que otras, por tal motivo tienen una distribución distinta entre persona y persona. Para determinar la posición de los folículos se usa como punto de partida una imagen, esta imagen representa las zonas en las cuales crecerá el cabello. Distintas intensidades de color verde representan distintas distribuciones, es decir, si la imagen tiene zonas totalmente verdes habrá una alta densidad de folículos, por otro lado si otra zona de la imagen tiende a ser negra, habrá menos densidad (Figura 5.2). Posteriormente con el uso de coordenadas baricéntricas [30] se obtienen las posiciones finales de los folículos.

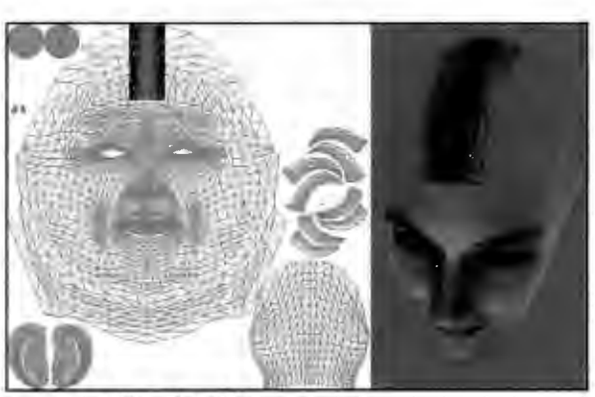

Fig. 5.2 Mapa de distribución y posición de las hebras del cabello

El algoritmo que distribuye los folículos y los posiciona en el modelo con base a lo anterior es el siguiente.

```
1. distribucionPosicionHebras {
2.      hebrasUsadas=0
3.      asignaIntensidadMax (mapaDensidad, MaxInt)
4.      numCaras = determina cuantas caras del modelo caen en zonas con color de la imagen
5.      hebrasPorCara = NUM_HEBRAS/numCaras
6.      Para i = 0 hasta numCaras-1{
7.              Para n=1 hasta MaxInt {
8.                      si  la cara i ocupa una zona a color con intensidad == n {
9.                              hebrasPorCara2 = hebrasPorCara*n
10.                             Para j=1 hasta hebrasPorCara2 {
11.                                     colorHebra = colorCara ( cara i)
12.                                     PosHebra = coordBaricéntricas( cara i )
13.                                     dirHebra = normal ( cara i)
14.                                     hebrasUsadas = hebrasUsadas +1
15.                                     j = j+1
16.                             }
17.                     }
18.             n = n+1}
19.     hebrasPorCara = (NUM_HEBRAS - hebrasUsadas)/(numCaras)
20.     i=i+1}
20. }
```

5.2.3 LONGITUD DE LAS HEBRAS

Una vez que se ha establecido la posición de las raíces del cabello es necesario especificar la dirección de crecimiento de las hebras y que tanto van a crecer, es decir su longitud. Por motivos de simplificación y para aprovechar el hecho de que se conocen las normales de cada una de las caras del modelo tridimensional se determinó que la dirección de crecimiento fuera la misma que la dirección de la normal en las caras correspondientes del modelo. En el siguiente algoritmo y con base a la intensidad de azul se establece la longitud de las hebras.

```
1. longitudHebra (decimal dir[3], pos[3], longitud){
2.      decimal dist [3]
3.      dist = distanciaXYZ (pos, dir)
4.      dist = dist * longitud
5.      Para x=0 hasta 2 {
6.              Si (pos[x]<0)
7.                      dir[x] = pos[x] - dist [x]
8.              de lo contrario
9.                      dir[x] = pos[x] + dist [x]
10.             x++}
```

En el algoritmo anterior se tiene como entrada dos vectores la dirección de la hebra, que es la misma que la dirección de la normal, la posición de su raíz y la longitud, que es una escalar. En general *pos* es un punto que indica donde empieza la hebra y *dir* es otro punto que indica donde termina, por tanto para diferentes longitudes se necesita que la distancia entre estos puntos sea aumentada o reducida. Para esto primero se calcula la distancia en *x*, en *y* y en *z* entre estos dos puntos (línea 3), cada una de estas distancias se almacena en el vector *dist* cuyas componentes son multiplicadas o escaladas en la línea 4. En las líneas 5 a la 9 se modifica la posición del punto *dir*, es decir, se acerca o se aleja del punto *pos*.

Una vez que se ha modificado la distancia entre los extremos de la hebra es necesario colocar los nodos de la polilínea esqueleto que serán utilizados en la animación como partículas. Para tener una distribución adecuada de los nodos a lo largo de la polilínea, es decir que exista la misma distancia de separación entre nodos se hizo uso de la interpolación lineal en el siguiente algoritmo.

```
1. ColocaNodos {
2.      decimal delta
3.      Para i=0 hasta i< NUMERO_HEBRAS{
4.              Para j=0 hasta j< NUMERO_NODOS {
5.              delta =j/(NUMERO_NODOS-1.0)
6.              Hebra[i][j].pos[0]=(1-delta)*Hebra[i][0].pos[0]+delta*Hebra[i][0].dir[0]
7.              Hebra[i][j].pos[1]=(1-delta)*Hebra[i][0].pos[1]+delta*Hebra[i][0].dir[1]
8.              Hebra[i][j].pos[2]=(1-delta)*Hebra[i][0].pos[2]+delta*Hebra[i][0].dir[2]
9.              j++ }
10.     i++}
```

Finalmente para la longitud se usan imágenes con componente azul donde negro representa la longitud mas corta y el azul representa la longitud mas larga (Figura 5.3).

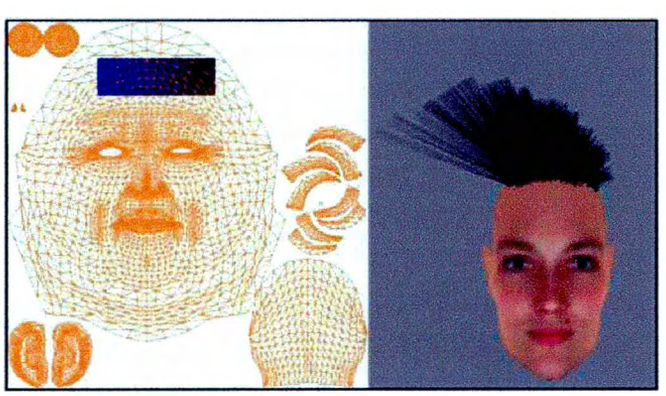

Fig. 5.3 Mapa de Longitudes

5.3 CARACTERÍSTICAS DE ESTILO

Las características de estilo permiten al usuario modelar distintas formas de hebras de cabello y por consiguiente modelar diferentes peinados. Así como con las características geométricas el usuario puede usar imágenes para especificarlas, el usuario podrá definir, en un mismo peinado, áreas lacias, flecos rizados, chinos, trenzas, mechones, entre otras con imágenes a colores.

5.3.1 RIZOS

Si se observa un rizo individualmente (Figura 5.4a), matemáticamente puede ser representado por una señal senoidal, pero si se desea una representación más precisa se tendría que añadir una señal de ruido a la señal senoidal para eliminar su simetría característica (Figura 5.4b), para evitar añadir dicha señal y aprovechar el hecho de que en el cabello rizado se presenta en general un comportamiento senoidal, para modelar los rizos, se usará una señal senoidal y cosenoidal, ambas con distintos periodos.

a) b)

Fig. 5.4 a) Rizo individual enmarcado en el círculo
b) Cabello rizado

Los rizos son especificados por imágenes con componente roja. La intensidad mínima produce cabello lacio, una intensidad moderada produce rizos y la intensidad máxima produce chinos.

Como se había mencionado anteriormente, cada hebra de cabello tiene una polilínea esqueleto y una polilínea gráfica, además la polilínea gráfica sigue la trayectoria de la polilínea esqueleto. Para representar diferentes niveles de rizado con la polilínea gráfica y que ésta siga la trayectoria de la polilínea esqueleto nos hemos basado en el resultado que aloja la suma de señales, es decir, una señal resultante con sus componentes reducidas o aumentadas según sea el caso (Figura 5.5).

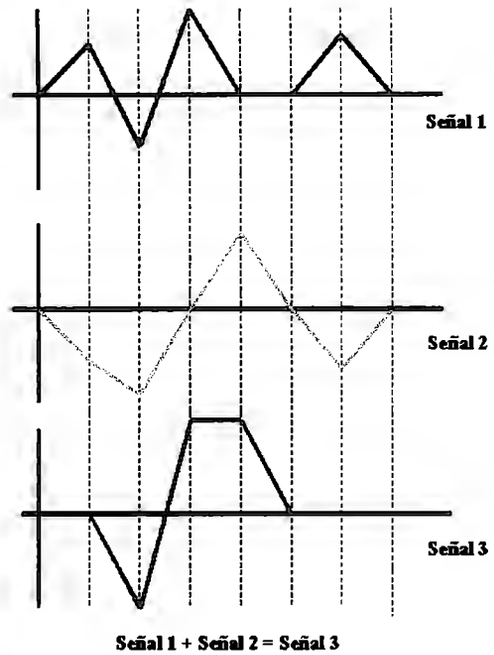

Fig. 5.5 Suma de señales

Para que la polilínea gráfica siga la trayectoria de la polilínea esqueleto estará conformada por un término que representa la interpolación lineal entre los nodos de la polilínea esqueleto y un segundo término que representa el rizado formado por señales senoidales y cosenoidales. En la ecuación 5.1 se muestra la expresión matemática de la polilínea gráfica rizada.

$$
\begin{aligned}
&x = [x_0(1-\Delta) + x_1\Delta] + [\cos\theta] \\
&y = [y_0(1-\Delta) + y_1\Delta] + [sen\theta] \\
&z = [z_0(1-\Delta) + z_1\Delta] + [sen\theta + \cos\theta] \qquad (5.1) \\
&con \qquad \Delta \in [0,1]
\end{aligned}
$$

Finalmente la imagen con componente roja permite ir variando el periodo de las señales (el término θ) para obtener diferentes niveles de rizado (Figura 5.6). Cabe mencionar que cuando se trata de cabello lacio (es decir la componente roja esta al mínimo) se usa el dibujado de líneas que proporciona OpenGL y en este caso la polilínea esqueleto sí es dibujada.

Fig. 5.6 Distintos niveles de rizados.
De izquierda a derecha el nivel de rizado aumenta

5.3.2 TRENZAS

En [31] se pueden encontrar instrucciones para peinar una amplia variedad de estilos de trenzas que van desde las trenzas clásicas de tres mechones (Figura 5.7a), hasta trenzas de corona (Figura 5.7b) pasando por las trenzas del personaje Arwen de los libros El Señor de los Anillos (Figura 5.7c).

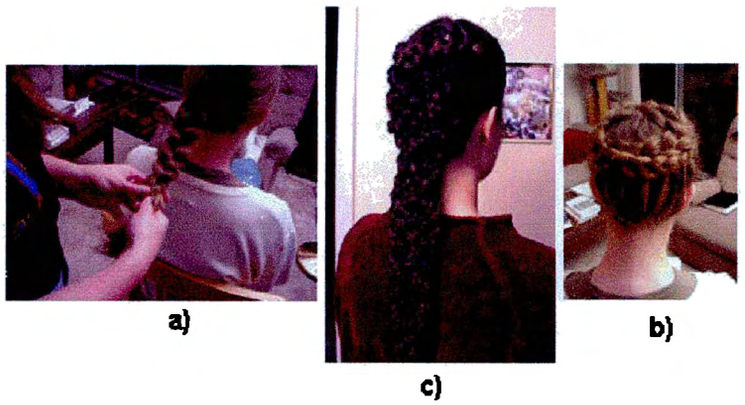

Fig. 5.7 Estilos de trenzas. a) Trenza Triple, b) Trenza de Corona, c) Trenza Arwen

El estilo de trenza que se decidió modelar es la trenza triple ya que es popularmente conocida. Si se examina detenidamente este tipo de trenza se podrá observar que típicamente está conformada por tres mechones que se cruzan entre sí. Este cruce origina que los mechones adquieran una forma espiral con periodo distinto por cada mechón. Por tal motivo para representar dicho comportamiento se recurrirá a la ecuación paramétrica de una espiral.

La ecuación paramétrica de una espiral está dada por la ecuación 5.2.

$$x = r\cos t$$
$$y = rsent$$
$$z = t \qquad (5.2)$$

donde:

r	radio de la espiral
t	es el parámetro

los términos x y y representan gráficamente un círculo, sin embargo, el término z provoca que los valores que se van obteniendo de x y y vayan siguiendo una ruta sobre el eje z. Si se cambia, por ejemplo, $y = rsen\ t$ por $y = t$ y $z=t$ por $z = rsen\ t$ ahora la ruta que se seguirá será sobre el eje y.

Como se acaba de mencionar, en la ecuación 5.2 la asignación $z=t$ provoca que la espiral siga una ruta fija sobre el eje z. Este comportamiento no es muy útil para representar la trenza debido a que es necesario que la espiral siga la ruta determinada por la polilínea esqueleto, para lograr este comportamiento se elimina la igualdad $z=t$, y se procede de una manera similar a como se hizo para los rizos, es decir, agregar el término de interpolación

lineal entre los nodos de la polilínea esqueleto y sumarlos con los términos de la ecuación 5.2 con lo cual se obtiene la ecuación 5.3.

$$\begin{aligned} x &= x_0(1-\Delta) + x_1\Delta + r\cos\theta \\ y &= y_0(1-\Delta) + y_1\Delta \\ z &= z_0(1-\Delta) + z_1\Delta + rsen\theta \qquad (5.3) \\ con &\qquad \Delta \in [0,1] \end{aligned}$$

La trenza triple requiere tres ecuaciones similares a la ecuación 5.3, una para cada mechón, con la diferencia de que a cada ecuación se le agrega una fase distinta w (ecuación 5.4) para obtener el aspecto característico de este tipo de trenza.

$$\begin{aligned} x &= x_0(1-\Delta) + x_1\Delta + r\cos(\theta + w) \\ y &= y_0(1-\Delta) + y_1\Delta \\ z &= z_0(1-\Delta) + z_1\Delta + rsen[2(\theta + w)] \qquad (5.4) \\ con &\qquad \Delta \in [0,1] \end{aligned}$$

En la figura 5.8 se muestra un ejemplo de una trenza triple simplificada obtenida con la ecuación 5.4.

Fig. 5.8 Trenza simplificada

Finalmente las trenzas son especificadas mediante una imagen color magenta que determina en donde poner o en donde no poner trenzas.

5.3.3 OPTIMIZACIÓN DE SENOS Y COSENOS

En las ecuaciones 5.1 y 5.4 para modelar rizos y trenzas respectivamente se hace uso intensivo de operaciones seno y coseno. Estas operaciones requieren de un tiempo considerable para ser resueltas por la unidad central de proceso, por tal motivo se precalculó una tabla de senos y otra tabla de cosenos. Sin embargo, como se requiere representar distintos niveles de rizado y las trenzas en sí, estas dos tablas no son suficientes y se necesita precalcular varias tablas con periodos distintos.

El cálculo y almacenamiento de estas tablas con señales senos y cosenos con diferentes periodos es inadecuado ya que se generan muestras repetidas y se ocupa espacio en memoria que se podría ahorrar. Por tal motivo, sólo se precalcula una tabla de senos y otra de cosenos con un periodo de 32π (que representa el nivel de rizado máximo) de 1024 muestras cada una.

Posteriormente para obtener diferentes periodos, que son utilizados para los rizos y para las trenzas, se hace uso de la decimación. El proceso de decimación (explicado a detalle en [32]) consiste en obtener una señal con frecuencia y en consecuencia periodo submúltiplo de la señal original, por ejemplo, sea $x_b[n]$ una secuencia

ordenada de muestras que es resultado de tomar los valores de otra x[n], (a cada N-ésima muestra), entonces $x_b[n]$ es una secuencia decimada de la segunda y se representa como:

$$x_b[n] = x[nN]$$

El proceso de decimación se ilustra en la Figura 5.9

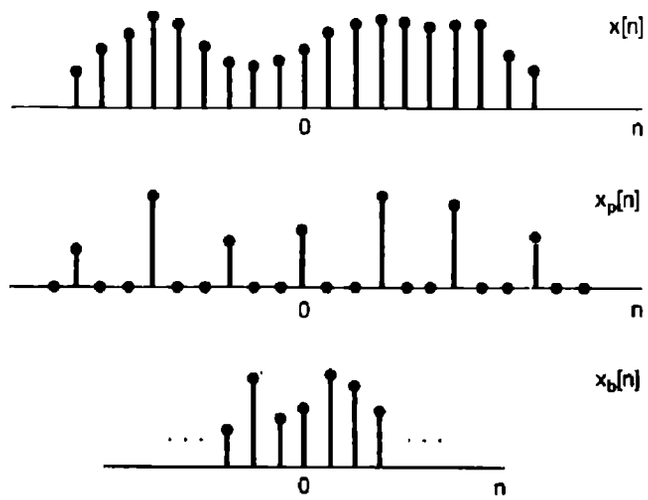

Fig. 5.9 Decimación. x[n] es la secuencia original, $x_p[n]$ ilustra las muestras de x[n] que integran a $x_b[n]$. Note que $x_b[n]=x_p[nN]$

Por lo tanto la decimación de las tablas de seno y coseno (como sus muestras están ordenadas) consiste solamente en tomar cierto número de muestras (Tabla 5.2) para obtener niveles de rizado menores.

Tabla 5.2 Decimación

Muestras	Periodo	Nivel de Decimación
1024	32π	1:1
512	16π	2:1
256	8π	4:1
64	2π	16:1
32	π	32:1

Finalmente el nivel de decimación (para los rizos) es obtenido con ayuda de la intensidad del color rojo de la imagen y es utilizada para saber cuantas muestras van a ser tomadas de las tablas de seno y coseno, es decir, para un nivel de decimación 1:1 se tomarán todas las muestras, para un nivel de decimación 2:1 se tomarán cada dos muestras y así sucesivamente.

5.3.4 COLOR

El cabello humano no es de un solo color y puede adquirir natural y artificialmente una gran variedad de tonalidades. El color del cabello es un aspecto importante en el peinado ya que cambia radicalmente el estilo y la forma en que es percibido. Por ejemplo, un hombre con cabello largo hasta los hombros de color oscuro actualmente pasa desapercibido pero si se cambia ese color por un color rojizo o simplemente se agregan rayitos amarillos prácticamente está garantizando que llame más la atención.

Por tal motivo y para que el usuario pueda modificar el color de su peinado como desee ya sea por zonas (Figura 5.10) o en forma gradual se ha agregado un mapa a colores el cual determina el color con el que van a ser dibujadas las polilíneas gráficas.

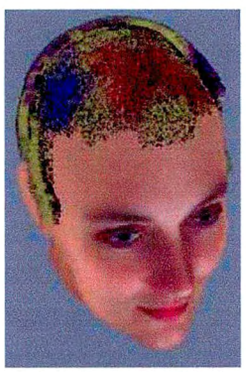

Fig. 5.10 Cabello Colorido

6. ANIMACIÓN DE CABELLO

La animación del cabello es un elemento que hay que tomar en cuenta si se desea que la apariencia final del peinado sea realista. El movimiento del cabello debe de responder a fuerzas como viento y gravedad además deseamos que esta animación sea en tiempo real. Como la animación de cabello que se propone está basada en física, primeramente se dará una descripción de lo que es un sistema de partículas así como las partes que lo integran, su movimiento y las fuerzas que provocan este movimiento. Posteriormente se mostrará una descripción de los métodos de integración que son utilizados para resolver las ecuaciones diferenciales de movimiento de las partículas y finalmente se abarcará lo concerniente al desarrollo e implementación de la animación en el cabello.

6.1 SISTEMA DE PARTÍCULAS

Las partículas son objetos con masa, posición y velocidad, además responden a fuerzas. Por su sencillez, las partículas son los objetos más fáciles de simular además permiten representar una amplia variedad de comportamientos. Por ejemplo, representar estructuras no rígidas con resortes – amortiguadores o en nuestro caso representar hebras de cabello.

Para representar una hebra de cabello hemos optado por usar una serie de nodos conectados entre sí. Para animar estos nodos, primeramente se representarán como un sistema de partículas y posteriormente se aplicarán las ecuaciones de movimiento sobre este sistema.

6.1.1 MOVIMIENTO DE LAS PARTÍCULAS

El movimiento Newtoniano de una partícula esta dado por la siguiente ecuación diferencial de segundo orden.

$$f = ma \quad ó \quad \ddot{x} = \frac{f}{m} \qquad (6.1)$$

Para resolver más fácilmente esta ecuación, es necesario convertirla en una ecuación de primer orden poniéndola en términos de la velocidad, o sea:

$$a = \dot{v} \Rightarrow \dot{v} = \frac{f}{m} \quad y \quad v = \dot{x} \qquad (6.2)$$

de esta forma se tienen dos ecuaciones diferenciales de primer orden cuya solución numérica es más fácil de encontrar.

Una vez que tenemos la versión simplificada de la ecuación 6.1, lo que interesa es conocer la posición de la partícula transcurrido un tiempo t bajo la influencia de fuerzas f. Es claro que con el valor conocido de las fuerzas y la masa, se puede obtener la aceleración de la partícula. Si se integra la aceleración con respecto al tiempo t, se obtendrá una nueva velocidad de la partícula, si la velocidad se integra de nuevo se obtendrá una nueva posición (ecuación 6.2). Para realizar estas integraciones se usan métodos numéricos, algunos de los cuales se describirán en secciones posteriores.

6.1.2 SISTEMA DE PARTÍCULAS BÁSICO

La simulación de partículas involucra dos etapas, las partículas en sí mismas y las entidades que aplican fuerzas sobre las partículas. Por el momento se dará una descripción de un sistema de partículas y mas adelante se mostrarán los tipos de fuerzas que actúan sobre este sistema.

Un sistema de partículas está conformado por variables y procedimientos, las variables permiten almacenar el estado actual de cada partícula y los procedimientos permiten modificar el estado de dichas partículas.

Como se mencionó anteriormente, una partícula tiene masa, posición y velocidad (Figura 6.1a), además están bajo la influencia de fuerzas, por tal motivo el sistema de partículas debe de estar conformado de estas cuatro variables. Además es deseable tener variables extras para almacenar el número total de partículas y el tiempo t para el cálculo de nuevas posiciones (Figura 6.1b).

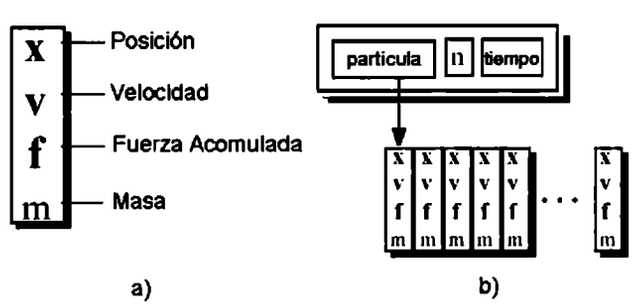

Fig. 6.1 a) Estructura de una partícula
b) Sistema de Partículas

Con respecto a los procedimientos, es necesario tener uno encargado de calcular las fuerzas que actúan sobre las partículas y otro procedimiento que calcula numéricamente los pasos de integración para conocer la nueva posición de las partículas.

6.1.3 FUERZAS

Las fuerzas que actúan sobre el sistema de partículas se dividen en tres grandes categorías:

- Fuerzas unarias, como la gravedad y fuerzas debido a la fricción. Estas actúan en forma independiente y ejercen fuerza constante sobre cada partícula.

- Fuerzas n-arias, como resortes y amortiguadores. Estas actúan sobre un conjunto fijo de partículas.

- Fuerzas de interacción espacial, como las fuerzas de repulsión y atracción. Estas actúan dependiendo de la posición de las partículas.

6.2 MÉTODOS NUMÉRICOS

Como se mencionó en la sección 6.1.1, para resolver el tipo de ecuaciones diferenciales descritas por las ecuaciones 6.1 y 6.2 es necesario el uso de métodos numéricos. En general, las ecuaciones diferenciales describen la relación entre una función desconocida y sus derivadas.

6.2.1 PROBLEMA DE VALOR INICIAL

Para resolver una ecuación diferencial es necesario hallar una función que satisfaga dicha relación y si es posible otras relaciones complementarias.

En un problema de valor inicial, el comportamiento de un sistema es descrito por una ecuación diferencial de la forma:

$$\dot{x} = f(x,t) \qquad (6.3)$$

donde:

f	es una función conocida		
x	es el estado del sistema	$\dot{x}$	es la derivada de x

En simulaciones gráficas basadas en física x y $\dot{x}$ se consideran vectores. En un problema de valor inicial se conoce la relación $x(t_0) = x_0$ en un tiempo t_0 dado y se desea conocer x para todo tiempo t.

De acuerdo a lo anterior, en un plano 2D, $x(t)$ define una trayectoria de una partícula en movimiento **p** en el plano. En cualquier punto x la función diferencial f puede ser calculada dando como resultado un vector diferencial, por lo tanto para todo el conjunto de puntos x, la función f define un campo de vectores sobre el plano. El vector obtenido en x representa la velocidad de la partícula en movimiento **p** que debería tener si se mueve a través de x, por tal razón, se puede decir que f "conduce" a **p** de un punto a otro. Ahora bien, la dirección hacia donde sea llevada la partícula **p** depende de la condición inicial que se haya establecido y finalmente todo movimiento futuro de **p** será determinado por la función f. La trayectoria que vaya formando **p** forma una curva denominada curva integral.

6.2.2 SOLUCIONES NUMÉRICAS

La solución numérica para encontrar una función cuyo comportamiento se conoce, consiste en tomar pasos discretos de tiempo iniciando en el valor inicial $x(t_0)$. La selección de dicho paso se realiza usando la función f para calcular un cambio aproximado en x, es decir Δx, sobre el intervalo Δt, para incrementar x en Δx unidades resultando el nuevo valor de x. De esta forma, para hallar una solución numérica de una función derivativa f, es necesario proporcionar valores numéricos a x y t, dando como resultado un valor numérico para $\dot{x}$. Los métodos numéricos existentes realizan una o más de estas evaluaciones por cada paso de tiempo.

6.2.2.1 Método de Euler

Uno de los métodos de integración numérica más sencillo es el método de Euler. Sea $x_0 = x(t_0)$ el valor inicial de x y $x(t_0 + h)$ el valor estimado de x en un tiempo $t_0 + h$, donde h es el paso estimado. El método de Euler calcula $x(t_0 + h)$ de la siguiente forma:

$$x(t_0 + h) = x_0 + h\dot{x}(t_0) \qquad (6.4)$$

Esta evaluación da como resultado lo siguiente, en vez de que la partícula **p** siga la curva integral, sigue un camino poligonal, que aproxima a dicha curva, cada intervalo es determinado evaluando $\dot{x}$ al principio y posteriormente es escalada por h. Cabe mencionar que entre más pequeño sea el paso h, más exacto será el camino poligonal y se obtendrá una mejor aproximación de la curva integral.

Sin embargo, aunque el método de Euler es sencillo, no es exacto. Por ejemplo supongamos que se tiene una función f cuya curva integral es una serie de círculos

concéntricos y una partícula **p** bajo las restricciones de *f*, dicha partícula debe orbitar por siempre en el círculo sobre el cual comenzó su movimiento. Sin embargo, con el método de Euler, en cada paso, **p** se moverá sobre una línea recta de un círculo a otro de mayor radio, dando como resultado una espiral (ver Figura 6.2). Reduciendo el tamaño del paso solo reduce los efectos de este fenómeno pero no los elimina.

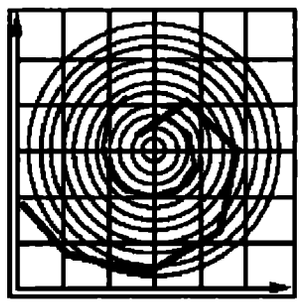

Fig. 6.2 Inexactitud del Método de Euler.
Distintos pasos de tiempo son representados

Otro de los problemas que tiene el método de Euler es su inestabilidad. Veamos un ejemplo, sea $f = -kx$, cuyo efecto sobre la partícula **p** es reducirla exponencialmente a cero. Para pasos lo suficientemente pequeños, se obtiene un comportamiento razonable, pero para pasos más grandes, como por ejemplo, cuando $h > 1/k$, se tiene que $|\Delta x| > |x|$, es decir el cambio asociado a x es mayor en cada h por lo tanto la solución oscila alrededor de cero. Más aún, si $h = 2/k$ los cálculos divergen y el sistema se desestabiliza por completo.

Para entender cómo se podría mejorar el método de Euler, es necesario observar el error que produce. La clave para entender este aspecto es el uso de series de Taylor; asumiendo que $x(t)$ es constante, se puede expresar su valor al final del paso, es decir, $x(t_0 + h)$ como una sumatoria de derivadas, es decir:

$$x(t_0+h) = x(t_0) + h\dot{x}(t_0) + \frac{h^2}{2!}\ddot{x}(t_0) + \frac{h^3}{3!}\dddot{x}(t_0) + \ldots + \frac{h^n}{n!}\frac{d^n x}{dt^n} + \ldots \qquad (6.5)$$

Como se puede observar, la formula usada en el método de Euler se obtiene truncando la serie, descartando todo aquel término que no sea el primero ni el segundo. Este truncamiento es correcto y por tanto el método de Euler también lo es solo cuando todas las derivadas de mayor orden a uno fueran cero, lo cual ocurre cuando $x(t)$ es lineal. El término que define el error (la diferencia entre la fórmula de Euler y la serie de Taylor completa) está dominado por el término $\frac{h^2}{2!}\ddot{x}(t_0)$, por lo cual se establece que dicho error es del orden $O(h^2)$. Ahora bien, si el tamaño del paso tomado se reduce a la mitad, es decir pasos de tamaño $h/2$, obtendríamos solo una cuarta parte del error, sin embargo la cantidad de pasos tomados sería el doble comparado con la cantidad de pasos cuando el tamaño del paso es h. Esto significa que el error que se acumula sobre un intervalo t_0 y t_1 depende

linealmente de h. Teóricamente, el uso del método de Euler permite calcular x sobre un intervalo de t_0 a t_1 con cierto error modificable mediante la elección de un paso de tamaño h apropiado. En la práctica, una gran cantidad de pasos pequeños se podrían necesitar, dependiendo del error obtenido y de la función f a procesar.

6.2.2.2 El método del punto medio y Runge-Kutta

Si se pudiera calcular $\ddot{x}$ como se calcula $\dot{x}$, se podría lograr una exactitud de $O(h^3)$ en vez de $O(h^2)$ agregando un término adicional a la serie de Taylor truncada:

$$x(t_0+h) = x(t_0) + h\dot{x}(t_0) + \frac{h^2}{2!}\ddot{x}(t_0) + O(h^3) \qquad (6.6)$$

Recordemos que la primera derivada $\dot{x}$ está dada por una función $f(x(t),t)$ y por simplicidad, se asumirá que la función diferencial f depende del tiempo solo directamente a través de x, por lo tanto $\dot{x} = f(x(t))$ y aplicando la regla de la cadena obtenemos:

$$\ddot{x} = \frac{df}{dx}\dot{x} = f'f \qquad (6.7)$$

Para no evaluar f', que numéricamente suele ser complicado y costoso, se puede aproximar el término de segundo orden solo en términos de f, y sustituir dicha aproximación en la ecuación 6.6, lo cual resulta en un error de orden $O(h^3)$. De esta forma, la función f se expande en series de Taylor de la siguiente forma:

$$f(x_0+\Delta x) = f(x_0) + \Delta x f'(x_0) + O(h^2) \qquad (6.8)$$

Introduciendo $\ddot{x}$ en esta expresión con:

$$\Delta x = \frac{h}{2} f(x_0) \qquad (6.9)$$

se tiene:

$$f(x_0 + \frac{h}{2} f(x_0)) = f(x_0) + \frac{h}{2} f(x_0) f'(x_0) + O(h^2)$$
$$= f(x_0) + \frac{h}{2}\ddot{x}(t_0) + O(h^2) \qquad (6.10)$$

donde $x_0 = x(t_0)$. Multiplicando ambos lados de la ecuación por h y reordenando, se obtiene:

$$\frac{h^2}{2}\ddot{x}+O(h^3)=h(f(x_0+\frac{h}{2}f(x_0)))-f(x_0) \qquad (6.11)$$

Sustituyendo el lado derecho de la ecuación 6.11 en la ecuación 6.6 y recordando que $\dot{x}=f(x(t))$, se obtiene la siguiente fórmula:

$$x(t_0+h)=x(t_0)+h(f(x_0+\frac{h}{2}f(x_0))) \qquad (6.12)$$

La ecuación 6.12 primero evalúa un paso Euler, después evalúa la segunda derivada en el punto medio de dicho paso para actualizar x. Aunque el método del punto medio es más preciso, requiere dos evaluaciones de f.

Si f es evaluada más veces, el error tiende a disminuir. Un método que realiza esto es denominado método Runge-Kutta de orden 4 y tiene un error por paso de orden $O(h^5)$ (el método del punto medio también se denomina método Runge-Kutta de orden 2) cuyas ecuaciones se muestran a continuación:

$$k_1=hf(x_0,t_0)$$
$$k_2=hf(x_0+\frac{k_1}{2},t_0+\frac{h}{2})$$
$$k_3=hf(x_0+\frac{k_2}{2},t_0+\frac{h}{2})$$
$$k_4=hf(x_0+k_3,t_0+\frac{h}{2})$$
$$x(t_0+h)=x_0+\frac{1}{6}k_1+\frac{1}{3}k_2+\frac{1}{3}k_3+\frac{1}{6}k_4$$

6.2.2.3 Método de Verlet

El método de Verlet es otro método de integración numérica sencillo. Típicamente en los sistemas de partículas, cada partícula tiene dos variables principales: posición x y velocidad v. Después de un paso de integración, una nueva posición x' y una nueva velocidad v' son calculadas. Usando el método de Euler se tiene lo siguiente:

$$x'=x-v\Delta t$$
$$v'=v+a\Delta t \qquad (6.13)$$

donde:

Δt	cambio del tiempo		
a	aceleración		
x	posición de la partícula	x'	nueva posición
v	velocidad	v'	nueva velocidad

En el método de Verlet, a diferencia del método de Euler, en lugar de tener la posición y velocidad de cada partícula, se tiene su posición actual x y su posición anterior x^*. Matemáticamente el método de Verlet está basado en dos expansiones de Taylor, una hacia adelante y otra hacia atrás, es decir:

$$x(t_0 + h) = x(t_0) + hx'(t_0) + \frac{h^2}{2!} x''(t_0) + \ldots$$
$$x(t_0 - h) = x(t_0) - hx'(t_0) + \frac{h^2}{2!} x''(t_0) - \ldots \qquad (6.14)$$

En términos de la ecuación de movimiento 6.1, estas ecuaciones se pueden rescribir como:

$$x^{t+h} = x^t + hv^{t+h} + \frac{h^2}{2}\frac{F^t}{m}$$
$$x^{t-h} = x^t - hv^{t+h} + \frac{h^2}{2}\frac{F^t}{m} \qquad (6.15)$$

Si se suman ambas ecuaciones, obtendremos la fórmula de integración de Verlet:

$$x^{t+h} = 2x^t - x^{t-h} + h^2\frac{F^t}{m} \qquad (6.16)$$

De esta forma el paso de integración es el siguiente:

$$x' = 2x - x^* + a\Delta t^2$$
$$x^* = x \qquad (6.17)$$

donde:

Δt	cambio del tiempo
a	aceleración
x	posición actual
x*	posición anterior

6.3 DESARROLLO

Como se mencionó en el capítulo de Modelado, cada hebra de cabello se representa mediante una polilínea esqueleto y una polilínea gráfica. La polilínea esqueleto está compuesta por un conjunto de nodos conectados. Estos nodos son utilizados en nuestro sistema de partículas sobre los cuales se calcula la animación basada en física.

Por otra parte, para que el sistema de partículas se pudiera utilizar para representar cada hebra de nuestro cabello fue necesario adoptar un mecanismo que permitiera mantener a las partículas del sistema unidas de tal manera que estuvieran conectadas una tras otra y separadas por una distancia dada. Dentro de las alternativas que se adoptaron para lograr lo anterior fue el uso de resortes y amortiguadores colocados entre las partículas. La otra alternativa consiste en aplicar restricciones que permitan mantener a las partículas a una distancia dada.

6.3.1 RESORTES Y AMORTIGUADORES

En una simulación masa-resorte, los resortes son elementos estructurales que mantienen todo el sistema unido además son utilizados para aplicar fuerzas opuestas e iguales entre las partículas. Sin embargo usar solo resortes en la animación de cabello hace que el sistema oscile debido a la falta de fricción, por otro lado es necesario que exista un elemento que permita reducir la velocidad de las partículas. Esta reducción de velocidad se logra con amortiguadores conectados con los resortes entre las partículas.

Para poder usar los resortes en el sistema de partículas es necesario recurrir a la Ley de Hook. La ley de Hook es una función de la longitud del resorte cuando es estirado o comprimido en relación a la longitud del resorte en reposo y la constante del resorte. La constante del resorte es una cantidad que relaciona la fuerza que es producida por el resorte desde que es estirado hasta que llega a estado de reposo. Matemáticamente esto es:

$$F_s = k_s(L - r) \qquad (6.18)$$

donde:

F_s fuerza que ejerce el resorte
k_s constante del resorte
L longitud del resorte comprimido o estirado
r longitud del resorte en reposo.

En relación a los amortiguadores, la fuerza ejercida por estos es proporcional a la velocidad relativa de las partículas conectadas y la constante del resorte, k_d, es decir:

$$F_d = k_d(v_1 - v_2) \qquad (6.19)$$

donde:

F_d fuerza que ejerce el amortiguador
k_d constante del amortiguador
v_1 velocidad de la partícula 1
v_2 velocidad de la partícula 2

De esta forma los resortes y amortiguadores quedan en cada hebra de cabello distribuidos como se ilustra en la Figura 6.3.

Fig. 6.3 Representación de una hebra de cabello con resortes y amortiguadores

Matemáticamente se tiene que la fuerza *F'* ejercida por los resortes y amortiguadores es:

$$F' = F_s + F_d$$
$$F' = k_s(L - r) + k_d(v_1 - v_2) \qquad (6.20)$$

Los problemas que surgieron con esta alternativa fueron los siguientes. Primero aunque la animación no oscilaba mucho debido a una buena selección de constantes tanto para el cambio de tiempo como para las constantes de los resortes y amortiguadores, existían ocasiones en que las hebras de cabello se estiraban más de lo normal, sobre todo cuando la fuerza del viento tenía la misma dirección que la gravedad (Figura 6.4).

a) b)

Fig. 6.4 a) Fuerza de Gravedad sin fuerza del viento b) Fuerza del Viento con la misma dirección que la Fuerza de Gravedad. Se observa que en b) el cabello se estiró haciéndolo parecer más largo que en a) aunque es la misma longitud.

Por otro lado cuando se agregaban más nodos a la polilínea esqueleto, como el peso total de la hebra aumentaba, los resortes se estiraban más ocasionando que la longitud inicial de las hebras cambiara. Esto último se podía corregir reduciendo las masa de cada nodo o aumentando las constantes de los resortes, sin embargo, en ocasiones el sistema de partículas simplemente se desestabilizaba por completo. Para evitar esta desestabilización había que volver a seleccionar otras constantes tanto para el cambio de tiempo como para los amortiguadores y las masas de cada partícula. Al final, se tenía que configurar de nuevo las constantes cada vez que se aumentaban o reducían los nodos de cada polilínea esqueleto.

6.3.2 RESTRICCIONES DE DISTANCIA

En general, el mecanismo de integrar resortes y amortiguadores entre los nodos de las hebras del cabello, proporciona implícitamente una restricción de distancia entre estos nodos debido a la constante de reposo del resorte r (ecuación 6.18), sin embargo existe otra alternativa que se adaptó para tener la posibilidad de eliminar los efectos negativos de los resortes y de los amortiguadores.

En general la restricción de distancia (ecuación 6.21) consiste en tres fases. Primero se debe establecer una distancia inicial entre las partículas de cada hebra. Después, en la animación, se debe detectar si esta distancia o esta restricción ha sido violada, esto es que la distancia entre los nodos pudiera ser mayor o menor que la inicial. Finalmente se deben separar o juntar las partículas según sea el caso (Figura 6.5).

$$|P_1 - P_2| = \lambda \qquad (6.21)$$

donde:

λ separación entre la partícula P_1 y la partícula P_2

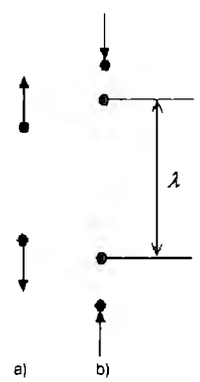

Fig. 6.5 Restricción de distancia. a) Separando las partículas
b)Juntando las partículas

Para satisfacer la restricción de distancia mostrada en la ecuación 6.21 se implementó el siguiente algoritmo:

```
1. Satisfacer Restricción () {
2.      Para cada partícula  de una hebra de cabello {
3.      Calcular la distancia relativa D entre partículas Pi y Pi+1
4.      Calcular la distancia absoluta |D|
5.      dif = (λ -|D|) / |D|
6.      Si |D| es mayor a λ
7.              Pi+1 += D*dif
8.      Si |D| es menor a λ
9.              Pi+1 -= D*dif    }
10. }
```

Con el uso de la restricción de distancia ya no es necesario calcular las fuerzas debido a los resortes y amortiguadores esto trae como consecuencia que se tengan que resolver menos ecuaciones. Además la animación de cabello mejora debido a que los efectos de estiramiento de los resortes es eliminado, lo cual resulta en un comportamiento más convincente.

6.3.3 MÉTODOS DE INTEGRACIÓN SELECCIONADOS

Hasta el momento ya se han explicado las dos alternativas que se utilizaron para mantener el sistema de partículas unido, sin embargo para animar el sistema es necesario resolver las ecuaciones diferenciales de movimiento Newtoniano. Como nuestro sistema debe poder ejecutarse en tiempo real, es necesario seleccionar un método de integración sencillo que proporcione resultados adecuados.

Dentro de los métodos que se seleccionaron fueron el método de Euler y el método de Verlet. El método de Euler se utilizó en conjunto con el sistema de partículas conectado con los resortes y amortiguadores. En general con la selección del método de Euler se logró una animación estable siempre y cuando se seleccionara un cambio de tiempo pequeño sin embargo, a consecuencia de los resortes, a veces parecía que el cabello estuviera compuesto de ligas.

Por otro lado, el método de Verlet se utilizó en conjunto con la restricción de distancia. Los resultados fueron mejores en comparación a la combinación Euler – resortes/amortiguadores ya que el comportamiento de los resortes resultante fue eliminado totalmente además el método de Verlet es estable [33]. Debido a que la velocidad está dada implícitamente, es más difícil que la posición y la velocidad de la partícula lleguen a estar desincronizadas. Esto se debe al hecho de que $2x - x^* = x + (x-x^*)$ y $x - x^*$ es una aproximación de la velocidad actual, de hecho es la distancia recorrida por la partícula en el último cambio de tiempo. Sin embargo el método de Verlet no es siempre exacto pero para la animación en tiempo real que queríamos lograr esa exactitud no fue un factor decisivo en el resultado final.

6.3.4 DETECCIÓN Y RESPUESTA A LAS COLISIONES

En general las colisiones permiten a las partículas no traspasarse entre sí o no traspasar cuerpos externos, de esta forma se obtiene más realismo en la animación. Hay dos partes interesantes con respecto a las colisiones: detectar las colisiones y responder a dichas colisiones. La detección de colisiones consiste en conocer si una partícula ha colisionado con otra o con un objeto en la escena. En general la detección de colisiones es difícil, pero hay casos en los que la detección de colisiones suele ser sencilla como por ejemplo la colisión partícula – plano, partícula – elipsoide y partícula – partícula.

En nuestro caso, para agregar colisiones a la animación, la cabeza se consideró como un elipsoide y cada partícula se consideró como una esfera con radio igual a cero, es decir, un punto. La diferencia que se tiene en tomar estas figuras geométricas para la cabeza y para las partículas es que el radio de un elipsoide no puede ser definido por un solo valor

como el radio de la esfera, si no está definido por un radio vector. Este radio vector es conformado por dos o tres componentes que indican dos o tres valores de escala según sea un elipsoide en 2D o 3D, por ejemplo una esfera con radio unitario al escalarla usando el radio vector (3, 4) se transformará en un elipsoide con radio en x igual a 3 y con radio en y igual a 4 (Figura 6.6).

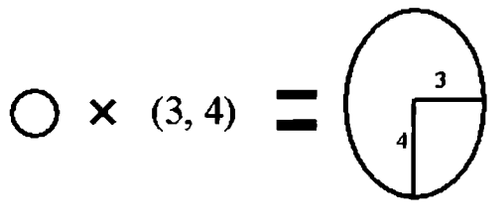

Fig. 6.6 Uso del radio vector para transformar una esfera en elipsoide

Para detectar las colisiones entre cabello – cabeza se utilizó la técnica para detectar colisiones entre esfera – esfera, donde una de ellas está en movimiento (la partícula) y la otra esta fija (la cabeza) con la diferencia de que se utilizó el radio vector para realizar las transformaciones necesarias. El algoritmo para detectar estas colisiones es el siguiente:

```
1. Detección de colisiones () {
2. Para cada partícula del cabello {
3.     posición = posParticula / rVector
4.     cambioEnDistancia = posición-centroEsfera
5.     Calcular la distancia  |D| = √(posición − centroEsfera)²
6.     Si |D| < radioUnitario
7.             Llamar a Responder a la colisión   }
8. }
```

Una vez que se sabe que una colisión ha ocurrido, es necesario responder a ella. Para describir la respuesta a las colisiones, es necesario obtener dos componentes ortogonales para los vectores de velocidad y fuerza, la primera componente ortogonal será normal a la superficie en colisión y la segunda componente será tangencial a esta superficie [34]. Por ejemplo, si **N** es la normal a un plano donde hace colisión la partícula, entonces la componente normal del vector x será $x_n = (\mathbf{N}\cdot x)x$ y la componente tangencial será $x_t = x - x_n$.

La colisión mas sencilla que existe es la colisión elástica sin fricción [34]. En este tipo de colisión la componente normal de la velocidad se multiplica por -1, de esta forma la partícula sigue su movimiento sobre la superficie en colisión. En una colisión inelástica, la componente normal de la velocidad es multiplicada por $-r$ donde r es la constante de restitución y toma valores entre cero y uno. En $r = 0$, la partícula no rebota, en $r = 0.9$ la partícula rebota al máximo.

Cuando la partícula está sobre la superficie en colisión con la componente normal de la velocidad en cero, se dice que la partícula está en contacto. Por ejemplo si la partícula

Tecnológico de Monterrey, Campus Ciudad de México

es empujada hacia el plano en contacto ($\mathbf{N}\cdot f < 0$) una fuerza de contacto $f_c = -(\mathbf{N}\cdot f)f$ es aplicada y cancela a la componente normal de f.

Finalmente en nuestro caso se desea que cada partícula, cuando se encuentre en contacto o dentro del volumen del elipsoide, se posicione fuera y se deslice sobre éste hasta caer como ocurría con el cabello, de esta forma el algoritmo para responder a la colisión es el siguiente:

```
1. Responder a la Colisión () {
2.      PosParticula = centroEsfera + cambioEnDistancia*(rVector/|D|)
3. }
```

Debido a que el método de Verlet no requiere conocer la velocidad de la partícula no se tienen que realizar operaciones extras para modificarla como ocurriría con el método de Euler.

Por el momento no se ha considerado el cálculo de colisiones entre cabello. La mayor razón se debe a que cuando el cabello esta bajo la influencia de fuerzas como el viento, el ser humano no puede inferir tan fácilmente que hebra de cabello está colisionando con otra o que mechón está en contacto con otro. Esto se debe a que el cabello tiene una apariencia difusa, es decir, no podemos distinguir fácilmente donde empieza una hebra de cabello y donde empieza otra. Por otro lado no estamos lidiando con casos mencionados por Chang *et al* [35] como cuando una trenza se deshace o cuando las cerdas de un cepillo tienen contacto con una esfera en movimiento.

6.3.5 SIMPLIFICACIÓN DE CÁLCULOS

La incorporación de la restricción de distancia y el método de Verlet mejoró la animación debido a que las ecuaciones diferenciales resultado de los resortes – amortiguadores ya no se deben resolver, sin embargo para obtener una animación en tiempo real o al menos interactiva, es necesario reducir las ecuaciones de movimiento y de detección y respuesta a las colisiones.

A continuación se explican las alternativas que se idearon. Primeramente se recurrió a la interpolación de movimiento. La interpolación de movimiento permite obtener nuevos cuadros de animación intermedios entre un cuadro de animación inicial y un cuadro de animación final.

Esta interpolación fue adaptada de la siguiente forma. Primero se definieron dos tipos de hebras de cabello. Aquellas a las cuales se les iban a aplicar las ecuaciones de movimiento denominadas hebras fundamentales y el segundo tipo las cuales iban a ser obtenidas interpolando linealmente los resultados de las hebras fundamentales denominadas hebras interpoladas. Una vez definido estos dos tipos, se calculan las nuevas posiciones de las hebras fundamentales, posteriormente se toman dos hebras fundamentales (de la misma

cara que conforma el modelo 3D), una sirve como posición inicial y la segunda como posición final, con base a estas posiciones se obtienen las nuevas posiciones de las hebras interpoladas (Figura 6.7).

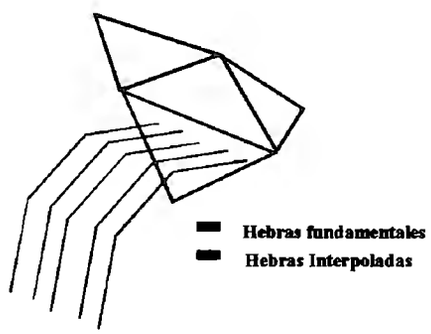

Fig. 6.7 Hebras Fundamentales y Hebras Interpoladas

Con este método se aumentaron los cuadros por segundo en la animación, sin embargo cuando se aumentaba el número de hebras interpoladas (Figura 6.8), se notaba que una gran cantidad de hebras interpoladas se agrupaban alrededor de las hebras fundamentales lo cual perjudicaba la apariencia final del cabello.

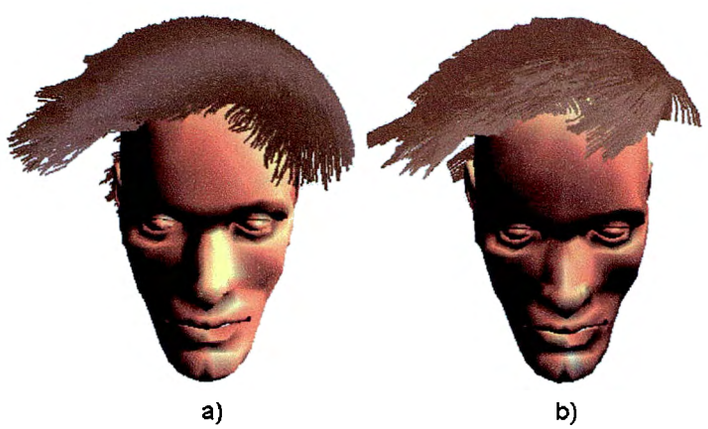

Fig. 6.8 Comparación entre a) Cabello sin interpolar (5000 hebras fundamentales) y b) Cabello interpolado (5000 hebras en total, 500 hebras fundamentales y 4500 hebras interpoladas)

Por los motivos anteriores fue necesario idear otro método que permitiera la reducción de ecuaciones a calcular sin alterar la apariencia final. Para lograrlo se han tomado algunas ideas relacionadas con modelar el cabello usando mechones. En el modelado de mechones las ecuaciones de movimiento son calculadas solo en algunas hebras de cabello denominadas hebras base. A continuación se muestra el algoritmo que permite reducir el número de ecuaciones de movimiento y de detección y respuesta a las colisiones, además permite modificar que tanto se deben agrupar las hebras asociadas con la hebra base. Dicha funcionalidad no está presente en el método de los mechones.

```
1. Genera Hebras Base
2. Genera Hebras Asociadas
3. Algoritmo Hebras Base - Asociadas {
4.      Por cada hebra base {
5.      Calcula las ecuaciones de movimiento y almacena los resultados (pos_actual, pos_nueva)
6.      Calcula la cantidad de movimiento ΔM =| pos_nueva - pos_actual|
7.      Por cada hebra asociada cambiar su posición P(x, y, z) por P(x, y, z)=P(x, y, z)+ΔM
8.      }
9.}
```

En el algoritmo las hebras base son aquellas sobre las que se calculan las ecuaciones de movimiento y las hebras asociadas son las hebras que dependen de los valores que se van calculando para las hebras base. Cada hebra base y sus hebras asociadas están posicionadas en la misma cara del modelo 3D.

Con la aplicación de este método se obtuvieron mejores resultados en términos de la apariencia y de la animación (Figura 6.9). Además, como se había mencionado anteriormente, se puede controlar la distancia entre las hebras base y sus asociadas para que se obtenga una apariencia mojada del cabello simplemente modificando la expresión $P(x, y, z)=P(x, y, z) + \Delta M$ por $P(x, y, z)=P(x, y, z) + \Delta M - c$ donde c es el nivel de agrupamiento de las hebras asociadas.

Fig. 6.9 Comparación entre a) Cabello con 5000 hebras base y b) Cabello con 5000 hebras en total, 500 hebras base y 4500 hebras asociadas

6.3.6 NIVEL DE DETALLE FÍSICO IMPLÍCITO

El algoritmo de las hebras base – asociadas permitió la reducción de cálculos, sin embargo cuando se tienen hebras de cabello rizadas o chinas y trenzas se deben de usar más nodos para su representación. En nuestro modelo de cabello rizado se usan entre 20 a 30 nodos y una trenza esta formada por 40 a 60 nodos dependiendo de su longitud. Por otra parte un peinado tiene alrededor de 10 000 hebras de cabello, entonces para un peinado con

rizos se tendrían como máximo 300 000 nodos a los cuales se les tendría que aplicar física; lo cual trae como consecuencia que el uso del algoritmo de las hebras base-asociadas no sea suficiente para obtener una animación interactiva.

Las polilíneas esqueleto y las polilíneas gráficas nos ayudan a resolver este problema. Como se había mencionado anteriormente, la polilínea esqueleto representa el camino que es seguido por la polilínea gráfica y está conformada por 4 o 5 nodos a los cuales se les aplica el algoritmo hebras base – asociadas. La polilínea gráfica representa las hebras de cabello rizadas, chinas y trenzas que se dibujan en el peinado final, sobre los nodos de esta polilínea no se aplica ninguna ecuación de movimiento y su posición está regida por la polilínea física. De esta forma aunque el peinado con hebras rizadas esté conformada por 300 000 nodos, gracias a la polilínea esqueleto solo entrarían al cálculo como máximo 50 000 nodos para un peinado con 10 000 hebras de cabello y 5 nodos por cada polilínea esqueleto.

7. VISUALIZACIÓN Y APARIENCIA

El cabello es una característica intrínseca en los seres humanos. Estamos tan familiarizados con su apariencia que sabemos exactamente como debe lucir, por tal motivo dependiendo del nivel de realismo obtenido en la simulación, un usuario común puede establecer que tan bueno o que tal malo es el peinado.

En términos computacionales, simular la apariencia natural del cabello es una tarea compleja debido a su estructura y a su comportamiento. Se debe determinar la geometría de cientos de hebras; se debe simular la dispersión de la luz a través de esa geometría para plasmar la naturaleza anisotrópica del cabello y calcular sombras entre las hebras del cabello y finalmente en el dibujado de las hebras se debe reducir el aliasing. Aunque en el sistema el modelado, la animación y la capacidad de realizar peinados es lo más importante, no se deja atrás la apariencia del cabello. Ya que la animación del cabello debe ejecutarse en tiempo real, para plasmar una apariencia del cabello adecuada se adaptó la técnica que propone Stalling *et al* [21][22] que nos permite plasmar la naturaleza anisotrópica del cabello y para el antialiasing se usará el canal alfa para dibujar las líneas con transparencia.

7.1 LA NATURALEZA ANISOTRÓPICA DE LAS SUPERFICIES

Una gran variedad de objetos como metales pulidos, discos compactos, telas como el satín y en nuestro caso el cabello están compuestos por superficies que reflejan anisotrópicamente los rayos de luz que inciden en ellos. Esta anisotropía es causada por micro estructuras que están alineadas predominantemente en una sola dirección, por ejemplo, en el metal cuando se pule se forman paralelamente una gran cantidad de ralladuras finas o en el cabello sus hebras están orientadas en una misma dirección.

En otras palabras, la anisotropía se origina debido a que la distribución de las normales de la superficie a lo largo de las ralladuras o de las hebras es diferente, esto origina que en ciertas áreas de la superficie se refleje más luz que en otras. Si el espectador está lo suficientemente lejos de la superficie la característica anisotrópica no podrá distinguirse (Figura 7.1).

Fig. 7.1 Ejemplo de un disco matizado con reflexión isotrópica (izquierda), y reflexión anisotrópica (derecha)

La naturaleza anisotrópica del cabello puede ser capturada en las texturas tridimensionales [9]. Las texturas tridimensionales están soportadas por hardware pero no en la mayoría de las tarjetas gráficas económicas, por lo tanto, este método no es adecuado para aplicaciones en tiempo real. Además estas texturas no son adecuadas para representar cabello largo. Por otro lado las texturas bidimensionales están soportadas por cualquier hardware gráfico además se puede explotar las capacidades de mapeo de textura de dicho hardware. Por tal motivo se optó por implementar la reflexión anisotrópica mediante una textura bidimensional.

7.2 MODELO DE PHONG

Para poder describir la interacción de la luz en una superficie es necesario conocer su vector normal ***N***, además dependiendo de la orientación, una superficie puede tener diferentes vectores normales. Sea ***L*** la dirección de la fuente de luz, ***V*** la dirección del espectador y ***R*** el vector de reflexión (Figura 7.2),

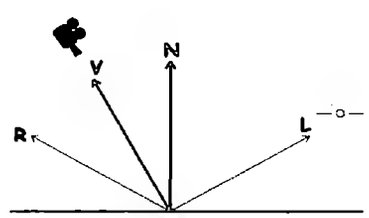

Fig. 7.2 Vectores para el Modelo de Phong

entonces la intensidad de la luz en un punto en particular de la superficie esta dada por la ecuación 7.1.

$$I = I_{ambiente} + I_{difusa} + I_{especular}$$
$$= k_a + k_d L \bullet N + k_s (V \bullet R)^n \qquad (7.1)$$

donde k_a, k_d, k_s son los coeficientes de las componentes ambiental, difusa y especular respectivamente.

En la ecuación 7.1, el primer término representa la intensidad de la luz ambiental. La luz ambiental es luz que ha sido dispersada una gran cantidad de veces en el ambiente que su dirección es difícil de determinar. Cuando la luz ambiental choca en una superficie se dispersa igualmente en todas direcciones. En general la luz ambiental se presenta debido a las múltiples reflexiones que ocurren en el ambiente.

El segundo término de la ecuación 7.1 describe la reflexión difusa, esta componente es la luz que viene de una fuente de luz. Una vez que choca en una superficie, se dispersa igualmente en todas direcciones, por tal motivo parece que tiene la misma brillantez sin importar el lugar donde el espectador este localizado.

El último término representa la reflexión especular, esta componente es originada en una fuente de luz y tiende a reflejarse sobre la superficie en una dirección específica. Los reflejos especulares están centrados alrededor del vector $\boldsymbol{R}$, su grosor es controlado por el exponente $\boldsymbol{n}$ llamado brillantez. A diferencia de la reflexión difusa, la reflexión especular tiene diferente brillantez dependiendo de la posición del espectador.

7.3 ILUMINACIÓN DE LÍNEAS EN R^3

Como se mencionó anteriormente el vector normal se utiliza para calcular la interacción de la luz con la superficie. En el caso de las líneas no se puede definir una normal y un solo vector de reflexión únicos para un punto en esa línea ya que hay una variedad de posibles vectores normales y de reflexión. Sin embargo para todos los posibles vectores normales, solo se tiene que seleccionar aquel vector normal que es coplanar al vector de la dirección de la fuente de luz $\boldsymbol{L}$ y al vector tangente $\boldsymbol{T}$ (Figura 7.3), de este modo ya se puede calcular la componente difusa usando el segundo término de la ecuación 7.1. De manera similar, para todos los posibles vectores de reflexión, se elige aquel que es coplanar a los vectores $\boldsymbol{L}$ y $\boldsymbol{T}$ (Figura 7.3) para poder calcular la componente especular usando el tercer término de la ecuación 7.1.

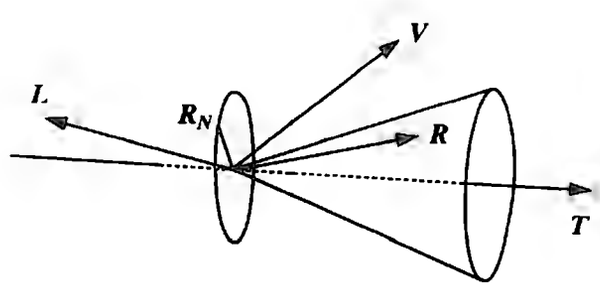

Fig. 7.3 Para las líneas, hay una infinidad de posibles vectores $\boldsymbol{R}$ que yacen en el cono alrededor de $\boldsymbol{T}$. Para calcular la iluminación se elige aquel que está en el plano $\boldsymbol{LT}$

El proceso de seleccionar la normal apropiada puede ser difícil. Para evitarlo, la componente difusa se puede expresar en términos de $\boldsymbol{L}$ y $\boldsymbol{T}$ siguiendo los siguientes pasos:

1. Proyectar el vector $\boldsymbol{L}$ en el plano $\boldsymbol{N} - \boldsymbol{T}$ entonces $\boldsymbol{L} = \boldsymbol{L_N} + \boldsymbol{L_T}$ (Figura 7.4).

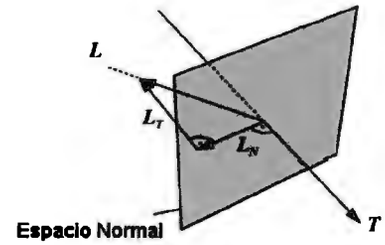

Fig. 7.4 Proyección del vector ***L***

2. Aplicando el teorema de Pitágoras y basándose en la proyección se obtiene la ecuación 7.2.

$$L \bullet N = |L_N| = \sqrt{1-|L_N|^2} = \sqrt{1-(L \bullet T)^2} \qquad (7.2)$$

Así la componente difusa se calcula usando la ecuación 7.3.

$$I_{difusa} = k_d \sqrt{1-(L \bullet T)^2} \qquad (7.3)$$

De manera similar para evitar el proceso de selección del vector de reflexión para calcular la componente especular se realizan las siguientes operaciones:

$$\begin{aligned} V \bullet R &= V \bullet (L_N - L_T) \\ &= V \bullet ((L \bullet N)N - (L \bullet T)T) \\ &= (L \bullet N)(V \bullet N) - (L \bullet T)(V \bullet T) \\ &= \sqrt{1-(L \bullet T)^2}\sqrt{1-(V \bullet T)^2} - (L \bullet T)(V \bullet T) \end{aligned}$$

Note que $\boldsymbol{R_N} = \boldsymbol{L_N}$ *y* $\boldsymbol{R_T} = -\boldsymbol{L_T}$. Por lo tanto para calcular la componente especular se usa la ecuación 7.4.

$$I_{especular} = k_s(\sqrt{1-(L \bullet T)^2}\sqrt{1-(V \bullet T)^2} - (L \bullet T)(V \bullet T))^n \qquad (7.4)$$

7.4 GENERACIÓN VISUAL DE LAS LÍNEAS ILUMINADAS

A pesar de que las ecuaciones de iluminación de líneas y superficie son similares, la forma en que se hace el cálculo para las líneas es distinta al de las superficies ya que para el caso de las líneas por cada nueva dirección de la luz se debe calcular un nuevo vector normal, sin embargo las ecuaciones 7.3 y 7.4 pueden ser evaluadas explotando las capacidades que proporcionan las tarjetas gráficas para el mapeo de textura.

7.4.1 MAPEO Y CREACIÓN DE LA TEXTURA ANISOTRÓPICA

El mapeo de textura permite "pegar" imágenes a polígonos o líneas. Estás imágenes generalmente se conocen como mapas de textura. Los mapas de textura son arreglos rectangulares de datos. Cada pieza del mapa de textura se denomina elemento de textura o texel. Aunque el mapa de textura es rectangular puede asignarse a objetos no rectangulares como esferas, cilindros y otros objetos tridimensionales.

Para el mapeo de textura se usa OpenGL. Para cada vértice de la hebra de cabello se especifica un vector homogéneo de coordenadas de textura. Generalmente las primeras componentes de este vector homogéneo son tomadas como índices de un mapa de textura en una, dos o tres dimensiones. Un mapa de textura puede contener colores con o sin componentes alfa, que se utiliza para modificar el color original del objeto al cual se le está asignando dicho mapa. Además es posible modificar las coordenadas de textura mediante la matriz de transformación de textura. Esta transformación es crucial para evitar recalcular las ecuaciones 7.3 y 7.4.

7.4.2 COMPONENTE DIFUSA

De la ecuación 7.3 se observa que la componente difusa está en términos de los vectores $\boldsymbol{L}$ y $\boldsymbol{T}$. Se sabe que el producto punto está dado por:

$$A \bullet B = A_1 B_1 + A_2 B_2 + \ldots + A_n B_n \qquad (7.5)$$

donde A, B son vectores con n componentes. Entonces para resolver el producto punto en la ecuación 7.3 definamos t_0 un vector de textura con 4 componentes, tal que $t_0 = \boldsymbol{T}$ para cada vértice de la línea y M una matriz 4x4 de transformación de textura tal que

$$M = \frac{1}{2}\begin{bmatrix} L_1 & 0 & 0 & 0 \\ L_2 & 0 & 0 & 0 \\ L_3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

multiplicando t_0 por M:

$$t_1 = t_0 M = \begin{bmatrix} t_{01} & t_{02} & t_{03} & 1 \end{bmatrix} \frac{1}{2}\begin{bmatrix} L_1 & 0 & 0 & 0 \\ L_2 & 0 & 0 & 0 \\ L_3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} t_{01} L_1 + t_{02} L_2 + t_{03} L_3 + 1 \end{bmatrix}$$

entonces por la ecuación 7.5 y recordando que $t_0 = \boldsymbol{T}$ tenemos que:

$$t_1 = \frac{1}{2}(L \bullet T + 1) \qquad (7.6)$$

Sea P un mapa de textura unidimensional que representará la componente difusa y t_1 el índice de P, además $t_1 \in [0, 1]$ entonces para obtener $\boldsymbol{L} \cdot \boldsymbol{T}$ se despeja de la ecuación 7.6 es decir:

$$L \bullet T = 2t_1 - 1 \qquad (7.7)$$

Sustituyendo 7.7 en la ecuación 7.3 tenemos que:

$$I_{difusa} = P(t_1) = k_d \sqrt{1-(2t_1-1)^2} \qquad (7.8)$$

la ecuación 7.8 se utiliza para calcular la componente difusa cuyos valores se van almacenando en el mapa de textura P. Note que la matriz de transformación de textura M sirve para modificar el mapa de textura P cada que la dirección de la luz cambie sin necesidad de recalcular la componente difusa.

7.4.3 COMPONENTE ESPECULAR

De la ecuación 7.4 se observa que la componente especular depende de dos productos puntos $\boldsymbol{L} \cdot \boldsymbol{T}$ y $\boldsymbol{V} \cdot \boldsymbol{T}$. Para calcular $\boldsymbol{V} \cdot \boldsymbol{T}$ se redefine la matriz de transformación M tal que

$$M = \frac{1}{2}\begin{bmatrix} L_1 & V_1 & 0 & 0 \\ L_2 & V_2 & 0 & 0 \\ L_3 & V_3 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

de esta forma la transformación de textura será dependiente de la dirección de la fuente de luz y de la dirección del espectador. De forma similar a la sección anterior sea t_2 un índice de textura y procediendo de la misma forma se tiene que

$$t_2 = \frac{1}{2}(V \bullet T + 1) \qquad (7.9)$$

despejando $\boldsymbol{V} \cdot \boldsymbol{T}$ y sustituyendo 7.7 y 7.9 en 7.4 tenemos que la componente especular esta dada por:

$$I_{especular} = k_s(\sqrt{1-(2t_1-1)^2}\sqrt{1-(2t_2-1)^2} - (2t_1-1)(2t_2-1))^n \qquad (7.10)$$

Para expresar correctamente la ecuación 7.10 es necesario el uso de un mapa de textura bidimensional tal que tenga índices t_1 y t_2, de esta forma el mapa de textura bidimensional $P(t_1, t_2)$ almacenará los valores obtenidos en la ecuación 7.10. Finalmente

para calcular el modelo de Phong se agrega a $P(t_1, t_2)$ la contribución de la componente ambiental (la constante k_a) y la contribución de la componente difusa (ecuación 7.8).

Una vez que se ha calculado el mapa de textura, este se asigna a los vértices de las hebras del cabello con las funciones que OpenGL proporciona para la generación automática de coordenadas de textura y la asignación del mapa de textura a dichos vértices. En la Figura 7.5 se muestra un ejemplo del mapa de textura que es aplicado a las hebras del cabello.

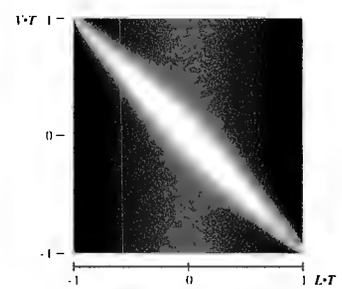

Fig. 7.5 Mapa de Textura Bidimensional para simular la reflexión anisotrópica del cabello con k_a=0.1, k_d=0.3, k_s=0.6 y n=40

7.5 ALIASING

Todas las primitivas que son dibujadas en el monitor de la computadora, como líneas, círculos y triángulos, tienen un problema común: el Aliasing. El aliasing produce un efecto dentado en estas primitivas y ocurre, por ejemplo, cuando se dibujan líneas que no sean horizontales o verticales. El aliasing es producido por la discretización cuyo enfoque es "todo o nada" [36], por lo cual cada píxel se reemplaza con el color de la primitiva o permanece sin cambio (Figura 7.6).

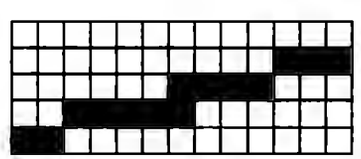

Fig. 7.6 Línea con aliasing

Para evitar este efecto dentado en las polilíneas gráficas es necesario reducir estos efectos para producir un modelo de cabello con apariencia más realista.

7.5.1 MEJORAS EN LA APARIENCIA DEL CABELLO

La aplicación del mapa de textura de la Figura 7.5 a las polilíneas gráficas permite simular la naturaleza anisotrópica del cabello, sin embargo existen otros aspectos relacionados con su apariencia.

El cabello tiene básicamente dos apariencias. Una apariencia cuando se encuentra en forma individual y la segunda cuando se encuentra agrupado o en mechones. Si se observa una hebra de cabello parece transparente, además si se examina detenidamente, en la raíz es

mas gruesa que en la punta. En cambio los mechones son opacos en el área central pero conforme se examinan sus alrededores estos parecen cada vez más transparentes.

Cuando se dibujan las polilíneas gráficas, por omisión, tienen un píxel de ancho; dibujar así estas polilíneas resulta en un cabello con aspecto más o menos natural (Figura 7.7a) sin embargo se puede mejorar esta apariencia añadiendo el canal alfa al color del cabello (Figura 7.7b).

Fig. 7.7 a) Peinado dibujado con polilíneas de un píxel de ancho
b) Peinado con componente alfa

Para simular el grosor característico de las hebras del cabello se procede de la siguiente forma. Se asigna el máximo valor alfa para el nodo raíz y un valor mínimo para el nodo de la punta, finalmente se interpola linealmente para obtener los valores correspondientes a los nodos que conforman el cuerpo de la polilínea (Figura 7.8).

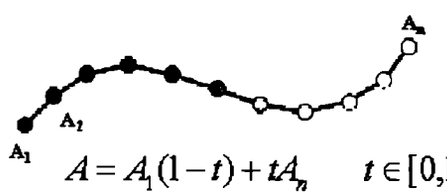

Fig. 7.8 Método para plasmar el grosor característico de las hebras de cabello

El beneficio que se obtuvo al incorporar el canal alfa al color del cabello fue una mejora sustancial en la apariencia del cabello además se redujeron significativamente los efectos de aliasing que se presentan en la Figura 7.7a porque la transparencia que tienen las polilíneas causan que el color final del píxel sea una mezcla del color de la polilínea y el color del objeto u objetos que están detrás de ella. Este método para reducir el aliasing está soportado por hardware gráfico económico sin embargo no hay que dejar de mencionar que aunque el antialiasing para líneas de OpenGL también es soportado por hardware gráfico económico al momento de ser activado, el rendimiento de la aplicación se reduce.

La descripción de otros métodos para reducir los efectos del aliasing puede encontrarse en el Apéndice C.

7.5.2 Quincunx Antialiasing

Quincunx Antialiasing es un método para reducir el aliasing en pantalla completa, en el Apéndice C se presenta una descripción de este método. Quincunx fue desarrollado por Nvidia y es soportado por hardware. Cuando se activa, los resultados obtenidos son ligeramente mejores.

8. INTERFAZ

Uno de los mayores problemas que el usuario afronta cuando va a diseñar un peinado por computadora es el manejo de la interfaz. Las alternativas comerciales presentadas en el capítulo Estado del Arte requieren de un conocimiento previo para empezar a colocar cabellos y realizar peinados básicos, además el proceso que se debe seguir es un tanto tedioso (Apéndice B) sobre todo si el usuario no conoce los comandos básicos de la interfaz. Por otra parte un usuario que no ha tenido contacto con un modelador comercial no estará acostumbrado al tipo de interfaz que este le proporciona debido a la infinidad de comandos que están presentes en estos programas.

A consecuencia de esto, se han diseñado otras interfaces para que el usuario pueda modelar peinados, de esta forma para definir las propiedades geométricas y de estilo del cabello se tienen varias alternativas. Una de las interfaces más socorridas es el uso de una interfaz interactiva tridimensional [1][7][37], en estas interfaces el usuario puede posicionar el cabello en cualquier sección de un modelo tridimensional, sin embargo este mecanismo es tedioso por la gran cantidad de cabellos o mechones que se tienen que posicionar aunque el usuario puede ver los resultados en tiempo real y modificarlo si algo ha salido mal. Por otra parte el uso de una interfaz tridimensional que es mostrada en un medio bidimensional (monitor de la computadora) suele traer dificultades a los usuarios inexpertos ya que el uso de un teclado y un ratón no es suficiente para utilizar intuitivamente la interfaz.

Otra alternativa es diseñar una interfaz automática capaz de posicionar las raíces del cabello por medio de coordenadas baricéntricas basándose en las caras triangulares del modelo tridimensional [30]. Esto resulta en una distribución aleatoria de los folículos sin embargo debido a que ciertas tareas están automatizadas el usuario no puede controlar fácilmente la distribución del cabello.

Existe otra interfaz que representa fielmente la forma natural en que un estilista peina, esta interfaz permite al usuario manipular una especie de peine electrónico sobre un modelo de cabeza de poliuretano con censores. Los resultados del peinado son mostrados en el monitor [38], sin embargo requiere de hardware especial que no está disponible comercialmente.

Para el diseño de la interfaz del cabello se tomaron las siguientes consideraciones para desarrollar una interfaz fácil de usar en la cual se puedan generar rápidamente peinados.

1. Cualquier usuario que esté familiarizado con el uso del ratón puede manejar la interfaz.

2. La interfaz debe ser interactiva para que el usuario pueda observar en tiempo real los resultados.
3. La interfaz se diseñó para que el usuario no tenga que aprender comandos complicados.
4. La animación y generación del peinado se automatiza.
5. La interfaz no utiliza hardware especial.
6. Debido a que el medio en el que se va a visualizar es bidimensional, la interfaz en la que el usuario va a trabajar es bidimensional con una ventana extra para que pueda ver los resultados en 3D.

De esta forma la interfaz que se propone es una interfaz basada en un programa de pintado ya que la mayoría de los usuarios comunes están familiarizados con este tipo de programas, además de proporcionar las herramientas necesarias para modelar peinados mediante imágenes. El uso de imágenes permite eliminar la dificultad y lo tedioso de poner mechones en una interfaz tridimensional y de definir características para las hebras de cabello.

8.1 DISEÑO DE LA INTERFAZ

Las interfaces de pintado no son adecuadas para modelar cabello ya que fueron diseñadas para hacer dibujos por tal motivo fue necesario adaptar ciertas características y agregar nuevos elementos para crear una interfaz nueva adecuada para modelar, animar y generar peinados.

La interfaz propuesta está dividida en tres áreas. El área de herramientas, las cuales son las herramientas típicas que proporciona un programa para pintado; el área de trabajo en la cual el usuario genera sus imágenes y finalmente un área con vista tridimensional la cual muestra el modelo 3D del peinado así como también se observa su animación.

8.1.1 ÁREA DE HERRAMIENTAS

El área de herramientas proporciona opciones al usuario para que pueda pintar figuras, figuras rellenas, figuras rellenas con degradados, diferentes tamaños de aerógrafos y pinceles. También se cuenta con una barra básica y una paleta de colores para que el usuario pueda variar colores a su conveniencia (Figura 8.1). También existe una barra de iconos en la que el usuario puede guardar y cargar las imágenes que va creando y también para que pueda cargar una variedad de modelos tridimensionales, finalmente se tiene un botón cuya función es mostrar el peinado en el área tridimensional cuando el usuario hace clic sobre él.

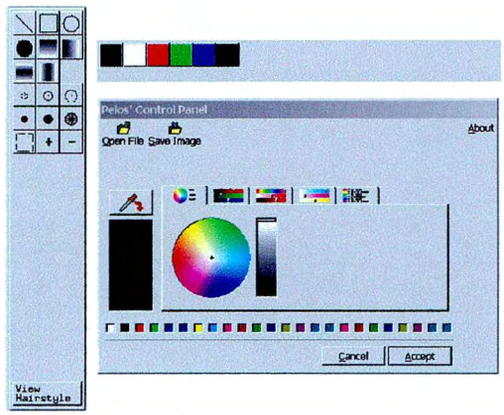

Fig. 8.1 Área de Herramientas

Las figuras disponibles en las herramientas son líneas, rectángulos y elipses. En el capítulo 3 de [36] se hace un estudio de los algoritmos para generar estas primitivas, sin embargo para aprovechar el hecho de que la interfaz está diseñada en OpenGL y C/C++, se utiliza el trazado de líneas y de polígonos de OpenGL (en [39] se encuentra una descripción detallada sobre estas características) para diseñar funciones encargadas del dibujado de las líneas, cuadros y elipses.

Para el dibujado de estas primitivas es necesario conocer y establecer las coordenadas de inicio y de fin, así como también el color que seleccionó el usuario. Para esto se diseñó una clase genérica que almacena estos datos `cShape`, posteriormente se diseñó una clase para cada primitiva (`cLine`, `cRectangle`, `cCircle`) que hereda las variables y funciones de la clase `cShape` con su respectiva función de dibujado.

Para poder responder a los eventos del ratón cuando el usuario inicia y finaliza una de estas primitivas se hace uso de la librería GLUT y los siguientes procedimientos

```
void glutMouseFunc(void (*func)(int button, int state, int x, int y))
void glutMotionFunc(void (*func)(int x, int y))
```

`glutMouseFunc` establece una llamada al procedimiento `*func` en la ventana actual, es decir, cuando el usuario presiona o libera los botones del ratón se produce esta llamada. `glutMotionFunc` establece una llamada al procedimiento `*func` cuando el ratón está en movimiento con cualquiera de sus botones presionados.

Por otra parte para obtener una representación uno a uno con las coordenadas del área de pintado y las coordenadas obtenidas del ratón se utilizó una proyección ortogonal (consultar [39] para una descripción más detallada) con la cual no se alteran las dimensiones de los objetos ya que son proyectados directamente en la pantalla.

Dibujado de Líneas

Una vez que se han establecido las coordenadas de inicio (x_o, y_0) y las coordenadas de fin (x_1, y_1) con ayuda de las funciones `glutMouseFunc` y `glutMotionFunc` se procede a dibujar la línea en OpenGL.

Dibujado de Rectángulos

Los rectángulos se dibujan de diferente forma de acuerdo a las necesidades del usuario. De esta forma se puede dibujar solo el contorno o rectángulos con relleno sólido y con relleno degradado horizontal, vertical, reflejado horizontal y reflejado vertical. Para poder hacer estos degradados se recurrió al uso del matizado suave o matizado de Gouraud que proporciona OpenGL [39]. Estos degradados están compuestos por dos colores, el color que seleccione el usuario y el color negro, de esta forma, se pueden especificar valores máximos y valores mínimos solo con una componente de color.

Dibujado de Elipses

Las elipses también pueden ser dibujadas sin relleno, con relleno sólido y con degradado radial. Para dibujarlas se usó la ecuación paramétrica de la elipse (ecuación 8.1) y el trazado de polígonos de OpenGL.

$$x = r_x \cos\theta$$
$$y = r_y sen\theta$$

donde:

r_x	radio en x
r_y	radio en y
θ	es el parámetro

Para determinar el centro de la elipse se calcula el punto medio entre las coordenadas de inicio (x_o, y_0) y de fin (x_1, y_1) introducidas por el usuario.

Aerógrafos y Pinceles

Los aerógrafos y pinceles tienen tamaños de 10, 15 y 20 píxeles y son un patrón de círculo con relleno degradado radial para los aerógrafos y un círculo con relleno sólido para los pinceles. Estos patrones se almacenan en arreglos para ser utilizados como imágenes RGB con ayuda de las funciones `glDrawPixels` y `glReadPixels` de OpenGL (en [39] se encuentra una descripción más detallada de estas funciones). El color de estas herramientas es especificado por el usuario, de esta forma también puede usarlas como borradores.

Paleta de Colores y barra de Iconos

La barra de colores solo proporciona los colores básicos rojo, verde, azul, negro y blanco, la necesidad de adicionar más colores a esta barra es eliminada usando una paleta de colores. Para agregar esta paleta de colores así como la barra de iconos para abrir imágenes y modelos y salvar imágenes se usó una librería para el diseño de interfaces gráficas en C++ llamada Fox [40].

8.1.2 ÁREA DE TRABAJO

En los programas de pintado el área de trabajo está conformada por un lienzo blanco en el cual el usuario traza su dibujo. Para que esta área sea funcional para el modelado de cabello se agregó una capa [41] a este lienzo en donde se pinta una representación bidimensional del modelo 3D con ayuda de sus coordenadas de textura. Esta representación bidimensional auxilia al usuario al momento de pintar las imágenes para que conozca que zonas del modelo tridimensional está especificando (Figura 8.2).

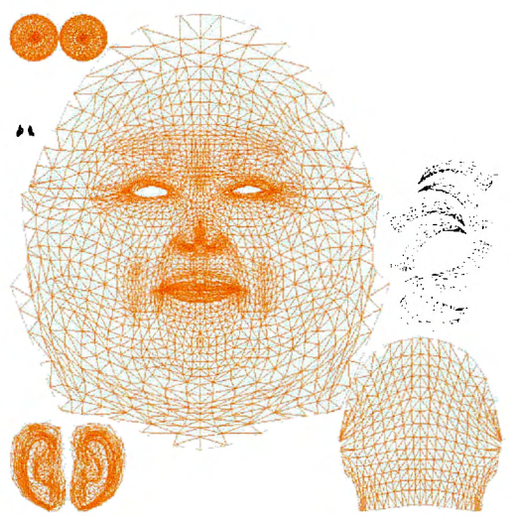

Fig. 8.2 Área de Trabajo

Por otro lado la adición de la capa permite tener en diferentes contextos de OpenGL tanto la representación bidimensional como las imágenes que genera el usuario lo cual evita que se sobrescriban píxeles.

Cuando alguna de las figuras es pintada, para auxiliar al usuario, se debe ir dibujando desde que hizo clic para definir las coordenadas de inicio hasta que libere el botón del ratón, además las figuras que ya pintó no deben ser borradas. Esto se logra con la técnica denominada "rubber banding" y consiste en estirar un objeto desde una zona inicial hasta donde el cursor se encuentre presente. El objeto es continuamente borrado y redibujado conforme el ratón se vaya moviendo. El fondo y las figuras debajo del objeto permanecen intactos.

La técnica de "rubber banding" se puede implementar en OpenGL fácilmente usando operaciones lógicas. Las operaciones lógicas permiten modificar los valores de los fragmentos entrantes con los fragmentos que se encuentran actualmente en el buffer de color [39]. Hay 16 operaciones lógicas que actúan sobre píxeles fuente (fragmento entrante) y píxeles destino (fragmento que esta en el buffer de color) para que puedan ser copiados o reemplazados. Estas 16 operaciones son seleccionadas con la función `glLogicOp` y se activan y desactivan con `glEnable` y `glDisable` respectivamente.

Para que la técnica de "rubber banding" funcione en OpenGL es necesario activar las operaciones lógicas con `glEnable(GL_COLOR_LOGIC_OP)`, usar la operación lógica X_OR y el modo de desplegado debe utilizar un solo buffer con componentes RGB. A continuación se muestra el algoritmo con palabras reservadas de OpenGL y GLUT que se utilizó para esta implementar esta técnica.

```
1. clicConElRaton (enteros: btn, state, x, y) {
2.      si btn==GLUT_LEFT_BUTTON & state == GLUT_DOWN {
3.              estableceCoordenadasIniciales (Figura, x, y)
4.              glLogicOp(GL_XOR)
5.              bandera = 1 }
6.      si btn==GLUT_LEFT_BUTTON && state == GLUT_UP {
7.              dibuja (Figura)
8.              glFlush()
9.              glLogicOp(GL_COPY)
10.             estableceCoordenadasFinales (Figura, x, y)
11.             dibuja (Figura)
12.             glFlush() }
13. }
```

```
1. ratonEnMovimiento (int x, int y) {
2.      si bandera ==0 {
3.              dibuja (Figura)
4.              glFlush() }
5.      estableceCoordenadasFinales (Figura, x, y)
6.      dibuja (Figura)
7.      glFlush()
8.      bandera =0
9. }
```

En las líneas 2-5 de `clicConElRaton` se establece el punto de inicio de la figura a dibujar y se activa la operación X_OR para que los fragmentos del buffer de color no sean eliminados por los fragmentos entrantes, la variable bandera se utiliza para especificar que se ha iniciado el pintado de la figura.

Inmediatamente después, como el ratón está en movimiento se entra a `ratonEnMovimiento` y conforme se va moviendo el ratón se establecen coordenadas finales

temporales y se dibuja la figura (líneas 5-8), recuerde que como la operación X_OR está activa, las figuras anteriormente pintadas no se borran, finalmente la variable bandera se pone a cero para indicar que el pintado está en proceso.

Por último cuando el usuario deja de presionar el botón del ratón se entra a `clicConElRaton` líneas 6-12 en donde se dibuja por última ocasión la figura en modo X_OR se establecen las coordenadas finales definitivas y se pasa a modo GL_COPY para que finalmente se dibuje la figura.

8.1.3 ÁREA 3D

En esta área se despliega en 3D el modelo con el peinado que el usuario está realizando (Figura 8.3). Cuando está activa, el usuario puede activar fuerzas debido al viento con las teclas de dirección, mover el modelo con el botón derecho del ratón y mover la fuente de luz con el botón izquierdo.

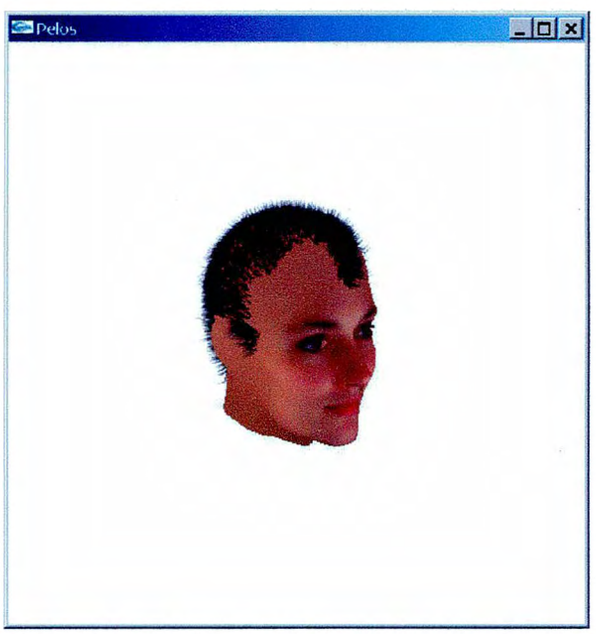

Fig. 8.3 Área de Visualización 3D

Finalmente en la figura 8.4 se muestra en conjunto la interfaz para el modelado, animación y visualización de cabello.

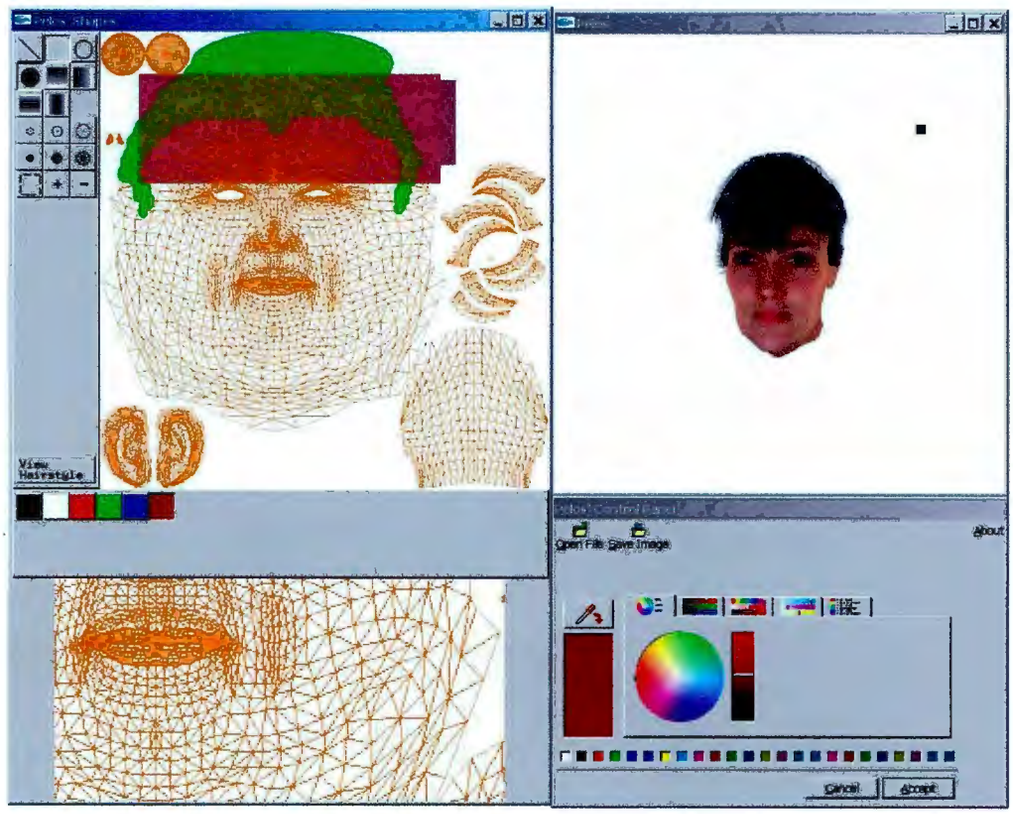

Fig. 8.4 Modelador de cabello

9. RESULTADOS

A continuación se muestran una serie de resultados obtenidos en las etapas de modelado, animación y visualización y apariencia. En la etapa de modelado se muestra una serie de peinados, en la animación se muestran diferentes tablas donde se ilustra el rendimiento de la animación así como una serie de imágenes para observar los resultados visuales de aplicar el algoritmo hebras base-asociadas y en la visualización y apariencia se muestra el efecto de agregar reflexiones anisotrópicas al modelo así como también efectos de aliasing y antialiasing en el peinado.

Para obtener los resultados de las tablas se usaron dos tipos de computadora. Un sistema portátil (sistema 1) con una configuración común y una estación de trabajo (sistema 2). La configuración de estos sistemas es la siguiente:

Sistema 1. Procesador Pentium 4 Mobile a 1.7 GHz
256 MB en memoria RAM
Tarjeta gráfica GeForce 4 440 Go con 32 MB de memoria
Windows XP

Sistema 2. Procesador Xeon 2.4 GHz
1 GB en memoria RAM
Tarjeta gráfica Quadro FX con 128 MB de memoria
Windows XP

9.1 MODELADO

Con lo que respecta a esta etapa, se muestran una serie de peinados, algunos con sus mapas que describen las características geométricas y de estilo del cabello.

En seguida se muestra un peinado típico con cabello corto (Figura 9.1), en este peinado solo fue necesario especificar el mapa de densidad. De los perfiles del modelo note como el cabello ocupa las partes especificadas por el mapa de densidades, por otra parte como no hay diferentes tonalidades de color verde, se está especificando la misma densidad para todas las caras del modelo. En la Figura 5.2 del capítulo de modelado se puede observar una distribución gradual.

Fig. 9.1 Peinado Típico con cabello corto

A continuación se muestra un peinado con cabello rizado (Figura 9.2), note como los mapas de densidad, longitud y nivel de rizado varían para modificar cada uno de estos parámetros. Por ejemplo si observamos el mapa de nivel de rizado (Figura 9.2d) se observa que en donde está especificado un rojo intenso se observan rizos (zona inferior del peinado) por otra parte en el área del fleco se observa un mechón rizado y sus mechones vecinos muestran una forma levemente quebrada. A los lados se puede observar que el cabello es lacio totalmente.

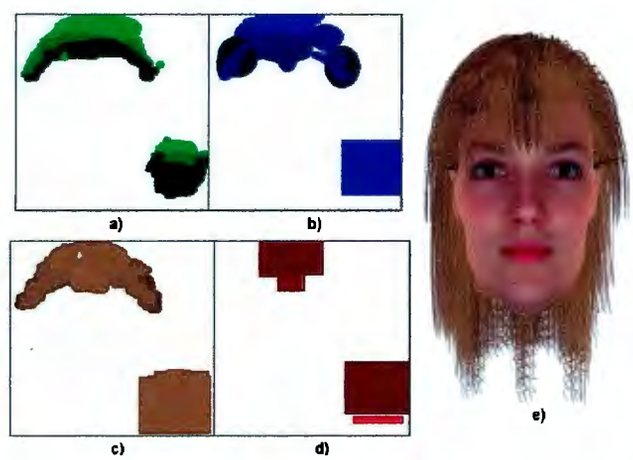

Fig. 9.2 Mapas a) densidad, b) longitud, c) color, d) nivel de rizado

Gracias a las capacidades del Modelador, se puede agregar vello facial para modelar barbas, cejas y bigotes (Figura 9.3)

Fig. 9.3 Modelo con cejas, bigote y barba

En la Figura 9.4 se muestra un conjunto de peinados realizados con la herramienta. De esta forma en 9.4a se muestra un peinado similar al de la Figura 9.1 pero pintado por zonas con diferentes colores. En 9.4b se muestra un típico peinado punk donde se puede observar diferentes longitudes, niveles de rizado y colores. La Figura 9.4c muestra un peinado rasta realizado con una serie de trencitas. En 9.4d se muestra un peinado convencional rizado con cabellos pelirrojos.

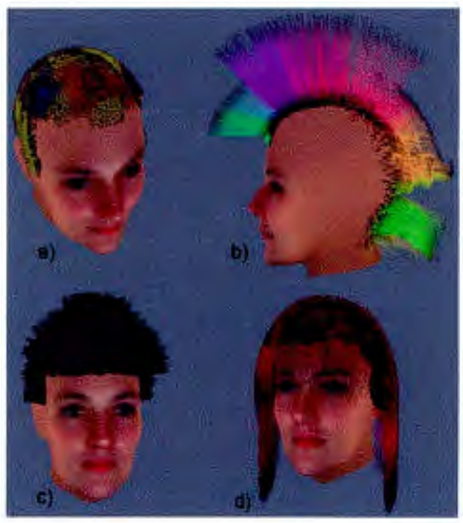

Fig. 9.4 Variedad de peinados

Finalmente en la figura 9.5 se muestra una trenza.

Fig. 9.5 Trenza

9.2 ANIMACIÓN

Como se mencionó en el capítulo de animación, el uso de la restricción de distancia para mantener unidas las partículas en vez de los resortes – amortiguadores y el método de Verlet mejoró el rendimiento de la animación ya que las ecuaciones de fuerza debido a los resortes – amortiguadores ya no tenían que ser resueltas. En la tabla 9.1 se muestra una comparación de rendimiento obtenidos en el sistema 2 entre estos dos esquemas.

Tabla 9.1 Rendimiento entre resortes - amortiguadores y restricción de distancia (Sistema 2)

Número de hebras de cabello	**Sistema de partículas con resortes - amortiguadores con método de Euler (Max. fps)**	**Sistema de partículas con restricción de distancia y método de Verlet (Max. fps)**
1000	60	63
5000	32	58
10000	15	25
20000	9	12

La incorporación del algoritmo Hebras base – asociadas permitió un aumento de rendimiento en la animación. En la tabla 9.2 y 9.3 se muestran diferentes resultados obtenidos variando el número de hebras base y el número de hebras asociadas. Además en 9.3 se puede observar el rendimiento cuando se tiene cabello lacio y cabello rizado

Tabla 9.2 Rendimiento de la Animación en Sistema 1

Número de hebras de cabello	**Hebras base**	**Hebras asociadas**	**cuadros / segundo (cabello lacio)**	**Figura**
5000	5000	0	31	9.5a
5000	2500	2500	36	
5000	500	4500	42	9.5b
14000	14000	0	10	9.5c
14000	7000	7000	14	
14000	1000	13000	19	9.5d

En la tabla 2 es interesante observar que para 5000 hebras base y ninguna hebra base se obtienen 31 cuadros por segundo, pero si sólo 500 hebras base son calculadas y 4500 son hebras asociadas se obtienen 42 cuadros por segundo. Números proporcionales son obtenidos para 14000 hebras. En las figuras 9.6a, 9.6b, 9.6c, 9.6d se puede apreciar que gráficamente no hay una diferencia notoria si la relación hebras base – asociadas es modificada.

Tabla 9.3 Rendimiento de la Animación en Sistema 2

Número de hebras de cabello	**Hebras base**	**Hebras asociados**	**cuadros / segundo (cabello lacio)**	**cuadros / segundo (Rizado Nivel decimación 4:1)**
10000	10000	0	25	7
10000	5000	5000	31	10
10000	1000	9000	58	15
30000	30000	0	7	2
30000	15000	15000	10	3
30000	1000	29000	25	5
50000	50000	0	6	2
50000	25000	25000	9	2
50000	1000	49000	16	3

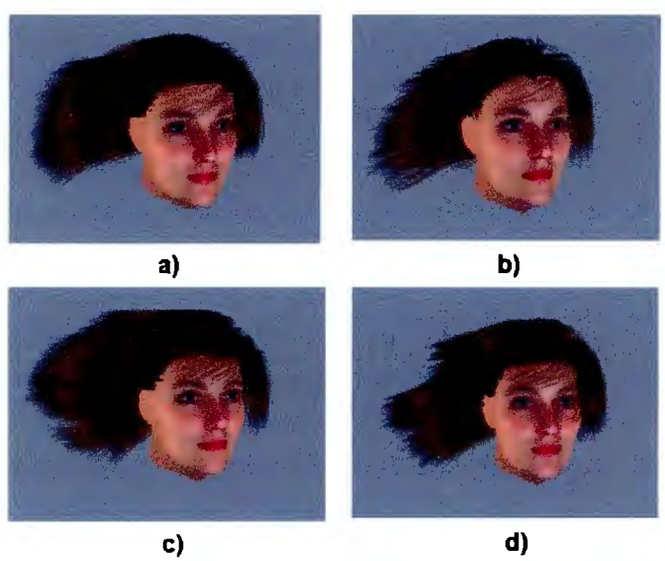

Fig. 9.6 a) 5000 hebras base, 0 hebras asociadas. b) 500 hebras base, 4500 hebras asociadas. c) 14000 hebras base, 0 hebras asociadas. d) 1000 hebras base, 13000 hebras asociadas

9.3 VISUALIZACIÓN Y APARIENCIA

El uso de una textura para representar la naturaleza anisotrópica del cabello permite mejorar drásticamente el aspecto del cabello. En las Figuras 9.7a y 9.7b se puede observar una comparación entre las imágenes donde no se aplica la textura (9.7a) y donde se aplica la textura (9.7b). Note en la Figura 9.7a el cabello luce sin su brillo característico, en la Figura 9.7b se mejora su apariencia.

Fig. 9.7 a) Sin reflexión anisotrópica. b) Con reflexión anisotrópica

Otros de los factores que ayudan a no lograr una apariencia convincente del cabello es el aliasing. Como se mencionó en el capítulo Visualización y Apariencia el aliasing se puede reducir con varios métodos, sin embargo para lograr un rendimiento interactivo o en tiempo real en la animación y para reducir el aliasing, solo se agregó el canal alfa al color del cabello y se utilizó el antialiasing por hardware Quincunx de Nvidia. A continuación se muestra un conjunto de figuras (Figura 9.8) en donde se muestran los efectos del aliasing en el cabello y la forma de cómo luce después de utilizar los dos métodos anteriormente mencionados.

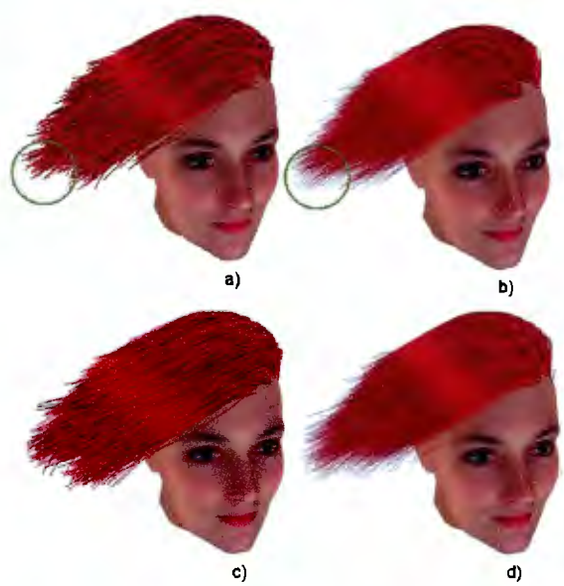

Fig. 9.8 Aliasing y antialiasing. Cabello a) sin aliasing. b) con componente alpha. c) con Quincunx Antialiasing d) con componente alpha y Quincunx antialiasing

En la Figura 9.8a y 9.8b se puede observar como la apariencia del cabello mejora con tan solo agregar la componente alfa al color, además las puntas de las hebras en 9.8b lucen más reales y sin artefactos que en 9.8a. En 9.8c el aspecto del cabello y del modelo (observe en la barbilla del modelo que ya no se notan los artefactos por el aliasing) mejora si se aplica Quincunx Antialiasing sin embargo las puntas de las hebras siguen estando muy gruesas. En la Figura 9.8d la única diferencia notoria que se tiene con 9.8b es que en las orillas del modelo de la cabeza ya no se perciben los artefactos, sin embargo el modelo del cabello luce muy similar. Por tal motivo con solo agregar el canal alfa al color permite principalmente reducir los efectos de aliasing y plasmar el grosor característico de las hebras de cabello.

10. CONCLUSIONES

La adición de cabello a personajes virtuales aumenta drásticamente su realismo debido a que es una característica intrínseca en el ser humano. A lo largo de este trabajo se ha mostrado una nueva herramienta para modelar, animar y visualizar cabello que permite al usuario diseñar rápido, fácil e interactivamente una variedad de peinados.

Esta herramienta es adecuada para simulaciones en tiempo real usando una computadora portátil común y mejores resultados se obtienen si se utiliza una estación de trabajo. El uso de una interfaz similar a la interfaz de un programa de dibujo permite al usuario familiarizarse fácilmente con la herramienta y reproducir peinados en forma menos tediosa que las alternativas presentadas ya que el uso de imágenes permite especificar características del cabello rápidamente.

La herramienta no está limitada al cabello humano ya que como es capaz de representar diferentes longitudes y formas de cabello también puede ser usada para modelar pelaje, ropa como faldas hawaianas, praderas y cualquier objeto que requiera de incorporación de pelo, solo es cuestión de cargar el modelo tridimensional adecuado.

La representación dual para modelar las hebras de cabello con el uso de las polilíneas esqueleto y las polilíneas gráficas permite generar diferentes niveles de rizado y trenzas sin tener que agregar más partículas al sistema para animarlo lo cual provoca al final un mejor rendimiento.

La inclusión de la restricción de distancia para sustituir al sistema de resortes-amortiguadores más la adición del método de Verlet para resolver las ecuaciones de movimiento permitió que la animación del cabello fuera más estable. La incorporación del algoritmo hebras base-asociadas permitió la reducción de cálculos lo cual resulta en que la animación fuera en algunos casos en tiempo real ya que llegaba a los 60 cuadros por segundo y en otros casos fuera interactiva con una taza de 20 a 30 cuadros por segundo.

El uso de hardware gráfico común permite obtener una apariencia relativamente realista del cabello sin sacrificar el rendimiento para que la herramienta sea ideal para aplicaciones en tiempo real. Por otra parte la adición de la componente alfa al color permite mejorar la apariencia del cabello ya que reduce los efectos de aliasing y permite caracterizar la variación de grosor de las hebras del cabello.

11. TRABAJO A FUTURO

Aunque los resultados obtenidos son más que adecuados para simulaciones en tiempo real es necesario mejorar algunos aspectos de la herramienta. Para complementarla es necesario agregar una serie de características del cabello preconfiguradas para permitir al usuario diseñar peinados más rápidamente. Estas preconfiguraciones serían una combinación de las características geométricas y de estilo existentes.

Para poder realizar peinados más elaborados es necesario agregar nuevas características geométricas como una herramienta de dirección que permita controlar la dirección de crecimiento del cabello y características de estilo para especificar accesorios como diademas, listones y lazos mezclados con las trenzas.

Para mejorar el aspecto visual, es necesario explotar las características del hardware gráfico disponible en el sistema 2 (capítulo 9) para agregar sombras entre cabello. Por otra parte, debido a que se está usando la componente alfa y en consecuencia se está trabajando con objetos semitransparentes el color final del píxel depende del último objeto que se dibujó, entonces para evitar que las hebras de cierto color que están detrás de otras parecieran que están delante, es necesario implementar un algoritmo que permita dibujar las hebras de cabello de atrás hacia delante dependiendo de la posición de la cámara.

12. REFERENCIAS

[1] DALDEGAN, THALMANN, KURIHARA AND THALMANN. "An integrated system for modeling, animating and rendering hair". *Computer Graphics Forum (Eurographics '93)* 12, 3 (1993), 211–221.

[2] WATANABE AND SUENAGA. "Drawing human hair using wisp model". *In Proceedings of Computer Graphics International'89* (1989), Springer Verlag, pp. 691–700.

[3] WATANABE AND SUENAGA. "A trigonal prism based method for hair image generation". *IEEE Computer Graphics & Applications 12*, 1 (January 1992), 47–53.

[4] YAN, XU, YANG AND WANG. "The cluster hair model". *Journal of Graphics Models and Image Processing* (1999). Academic Press.

[5] CSURI, HAKATHORN AND PARENT. "Towards an interactive high visual complexity animation system". *In Computer Graphics* (1979).

[6] MILLER, G. S. P. "From wire-frames to furry animals". *Graphics Interface '88* (June 1988), 138–145.

[7] KIM AND NEUMANN. "Interactive Multiresolution Hair Modeling and Editing". *ACM Transactions on Graphics 2002*, 21, 3, 620-629.

[8] PERLIN AND HOFFERT. "Hypertexture". *Computer Graphics (Proceedings of SIGGRAPH 89)* 23, 3 (July 1989), 253–262.

[9] KAJIYA AND KAY. "Rendering fur with three dimensional textures". *Computer Graphics (Proceedings of SIGGRAPH 89)* 23, 3 (July 1989), 271– 280.

[10] GUANG AND ZHIYONG. "A Method of Human Short Hair Modeling and Real Time Animation". *Proceedings of the 10 th Pacific Conference on Computer Graphics and Applications 2002.*

[11] HADAP AND THALMANN. "Interactive hair styler based on fluid flow". *In Proceedings of Eurographics Workshop on Computer Animation and Simulation '2000* (2000).

[12] ANJYO, USAMI AND KURIHARA. "A simple method for extracting the natural beauty of hair". *Computer Graphics (Proceedings of SIGGRAPH 92)* 26, 2 (July 1992), 111–120. ISBN 0-201-51585-7.

[13] ROSENBLUM, CARLSON AND TRIPP. "Simulating the structure and dynamics of human hair: Modeling, rendering and animation". *Journal of Visualzation and Computer Animation 2* (June 1991), 141–148.

[14] KURIHARA AND ANJYO. "Hair Animation with Collision Detection". *Models and Techniques in Computer Animation.* Springer Verlag, ch.

[15] BANDO, CHEN, NISHITA. "Animating Hair with Loosely Connected Particles". *Computer Graphics Forum* vol 22, nº 3 (2002)

[16] BERTAILS, F. KIM. CANI, M. NEUMANN, U. "AdaptiveWisp Tree – a multiresolution control structure for simulating dynamic clustering in hair motion". *Symposium on Computer Animation '03*, July 2003

[17] REEVES. "Particle systems: a technique for modeling a class of fuzzy objects". *ACM Transactions on Graphics 2*, 2 (April 1983), 91–108.

[18] GOLDMAN. "Fake fur rendering". *Proceedings of SIGGRAPH 97* (August 1997), 127–134. ISBN 0-89791-896-7.

[19] LEBLANC, TURNER AND THALMANN. "Rendering hair using pixel blending and shadow buffer". *Journal of Visualization and Computer Animation 2* (1991), 92–97.

[20] REEVES, SALESIN AND COOK. "Rendering antialiased shadows with depth maps". *Computer Graphics (Proceedings of SIGGRAPH 87) 21*, 4 (July 1987), 283–291.

[21] ZÖCKLER, STALLING AND HEGE. "Interactive visualization of 3d vector fields using illuminated streamlines". *IEEE Visualization '96* (October 1996), 107–114. ISBN 0-89791-864-9.

[22] STALLING, ZÖCKLER, AND HEGE. "Fast Display of Illuminated Field Lines". *IEEE Transactions On Visualization And Computer Graphics*, Vol. 3, No. 2, April-June 1997

[23] BRUDERLIN. "A method to generate wet and broken-up animal fur". *Pacific Graphics '99* (October 1999).

[24] KONG AND NAKAJIMA. "Visible volume buffer for efficient hair expression and shadow generation". *Computer Animation '99, IEEE Computer Society* (May 1999). IEEE Press.

[25] MARSCHNER, JENSEN, CAMMARANO. "Light Scattering from Human Hair Fibers". *Proceedings of ACM SIGGRAPH 2003.*

[26] JOSEPH ALTER, INC. "Shave And Haircut". http://www.joealter.com, visitada por última vez el día 30 de enero de 2004.

[27] ALIAS SYSTEMS. "Maya Fur" http://www.alias.com/eng/products-services/maya/technical_features/maya_fur/index.shtml, visitada por última vez el día 30 de enero de 2004.

[28] CURIOUS LABS INCORPORATED, AND E FRONTIER, INC. "Poser". http://www.curiouslabs.com/, visitada por última vez el día 30 de enero de 2004.

[29] WARLEY LABORATORIES. "Sasquatch", http://www.worley.com/ sasquatch/sasquatch.html, visitada por última vez el día 30 de enero de 2004.

[30] VAN GELDER AND WILHELMS. "An Interactive Fur Modeling Technique". *Proceedings of the conference on Graphics interface 1997*, p.181-188, May 1997, Kelowna, British Columbia, Canada.

[31] BERTKE. http://www.dreamweaverbraiding.com/Braiding_Tips.htm, visitada por última vez el día 30 de enero de 2004.

[32] MERTINS. "Signal analysis: wavelets, filter banks, time-frequency transforms and applications". Chichester, West Sussex, England ; New York: J. Wiley, c1999. English rev. ed. 317 p

[33] JAKOBSEN. "Advanced Character Physics", *Gamasutra January, 21, 2003* http://www.gamasutra.com/resource_guide/20030121/jacobson_01.shtml, visitada por última vez el día 30 de enero de 2004.

[34] BARAFF, WITKIN. "Physically Based Modeling: Principles and Practice", Notas de Curso en ACM SIGGRAPH 1997.

[35] CHANG, JIN, AND YU. "A Practical Model for Hair Mutual Interactions". *In Symposium on Computer Animation 2002*, ACM SIGGRAPH, p. 73-80.

[36] FOLEY, VAN DAM, FEINER, HUGHES AND PHILIPS. "Introduction to Computer Graphics". Reading, Mass. : Addison-Wesley, c1994. 559 p.

[37] CHEN, SAEYOR, DOHI, AND ISHIZUKA. "System of 3D Hairstyle Synthesis Based on the Model". *The Visual Computer*, 15(4), 159-170.

[38] LEE, CHEN, LEU AND OUHYOUNG. "A Rotor Platform Assisted System for 3D Hairstyles". *Journal of WSCG*, 10 (1-3), 271-278.

[39] NEIDER, DAVIS, WOO. "OpenGL programming guide : the official guide to learning OpenGL, version 1.2". Reading, MA : Addison-Wesley, 1999, 3rd ed, 730 p.

[40] VAN DER ZIJP, J. "Fox Toolkit", http://www.fox-toolkit.org/, visitada por última vez el día 30 de enero de 2004.

[41] WYNN. "Using P-Buffers for Off-Screen Rendering in OpenGL". Whitepaper, NVIDIA Corp., 2001.

[42] BRIGGS, CHAD. "Creating Hair in Maya". http://www.creativecow.net /articles/briggs_chad/maya_hair/, visitada por última vez el día 30 de enero de 2004.

[43] "HRAA: High-Resolution Antialiasing Through Multisampling", Technical brief, NVIDIA Corp., 2001.

[44] MÖLLER, HAINES. "Real Time Rendering". Natick, Mass. : AK Peters, 2002, 2nd ed. 835 p.

13. APÉNDICE A
LA NATURALEZA DEL CABELLO

El cabello humano geométricamente es un cilindro delgado, largo y curveado. El grosor del cabello puede variar entre 0.03mm a 0.09.

La fibra del cabello esta compuesta por tres capas que definen la forma y color de cada cabello: la cutícula, corteza y medula.

La cutícula es la capa exterior compuesta de escamas planas que se empalman entre si como tejas en un techo. Esta distribución proporciona protección a la corteza y otorga flexibilidad y fortaleza al cabello. La flexibilidad de la cutícula permite que el cabello adquiera formas onduladas y rizadas si quebrarse. Cuando el cabello esta en buena condición, las escamas de la cutícula yacen planas, creando una superficie suave y reflectiva sobre la cual los rayos de luz pueden incidir. Por otro lado, cuando el cabello esta dañado, las escamas de la cutícula pierden su alineación natural y crean bordes imperfectos. Esta superficie dispareja, no refleja los rayos de luz lo cual causa que el cabello luzca opaco y sin vida. La cutícula también es responsable de las propiedades repelentes del agua que tiene el cabello. Estas propiedades son derivadas de la presencia de lípidos en la queratina de la cutícula.

La corteza es la capa inferior de la cutícula y forma desde un 75 hasta un 90 por ciento la fibra del cabello. La corteza está formada de un gran número de células, unidas entre si mediante un material conocido como complejo de membrana celular. Dichas células tienen una forma alargada y delgada. En la corteza se produce el pigmento del cabello además, en esta capa se definen otras propiedades del cabello como la dirección en que crece el cabello, el tamaño y diámetro de cada cabello así como su textura y calidad, también es responsable de las propiedades mecánicas del cabello y dicta la deformación así como el comportamiento relacionado con la tensión del cabello cuando es afectado por fuerzas externas.

La medula compone el núcleo de la fibra del cabello cuya función se desconoce. En algunos casos, la médula puede no estar presente en la fibra sin embargo, su ausencia no parece modificar el comportamiento general de cada cabello.

Los daños principales que recibe el cabello se deben a la abrasión mecánica debida a la acción de peinar o cepillar el cabello, el uso de productos para el tratamiento del cabello como colorantes, lociones e incluso el shampoo suelen dañar el cabello. La cantidad de daño que pueda recibir el cabello debido a la abrasión mecánica depende del

coeficiente de fricción entre el peine o cepillo y la superficie del cabello. Este valor se incrementa dramáticamente si el cabello esta mojado.

Hablando en términos de iluminación, el cabello tiene una componente especular muy alta. El cabello de colores oscuros frecuentemente luce más brillante que aquellos que son claros, esto se debe a que en el cabello con tonalidades claras, parte de la luz es reflejada en la superficie del cabello mientras que la otra parte es refractada hacia el interior que posteriormente se refleja en otras superficies, cuando esto sucede, la componente difusa se incrementa lo cual trae como consecuencia que el cabello luzca más opaco. Por otra parte, en cabello oscuro, esa componente difusa es absorbida, por lo tanto el cabello oscuro luce más brillante.

Un cabello suele reflejar más fuertemente aquellos rayos de luz que son perpendicular a la dirección de este. Dicho comportamiento da como resultado la naturaleza anisotrópica del cabello.

Un humano con un cuero cabelludo y cabello en condiciones aceptables tiene alrededor de 100, 000 cabellos, cada uno de estos asociado a un folículo. El folículo es una extensión de la epidermis, la capa exterior de la piel.

14. APÉNDICE B
MODELADO DE CABELLO EN MAYA

El primer paso para crear cabello es preparar un modelo. En la Figura 14.1 se muestra una imagen del modelo que se utilizará.

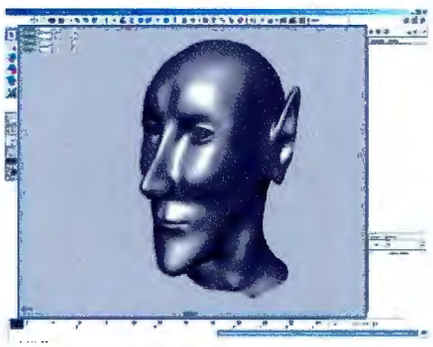

Fig. 14.1 Modelo a utilizar

En este modelo, la cabeza esta formada por superficies de subdivisión, por lo tanto está conformada de polígonos. Como no se puede utilizar PaintFX directamente sobre polígonos, se tiene que construir un cráneo con NURBS para pintar sobre ellas. Para construir dicho cráneo es necesario hacer una esfera y deformarla de tal forma que embone la parte superior y trasera del modelo donde se colocará el cabello (Figura 14.2).

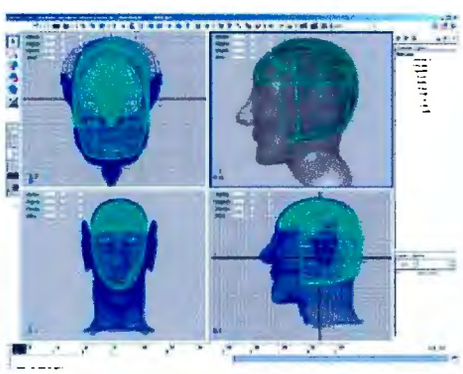

Fig. 14.2 El cráneo conformado con NURBS

El próximo paso es modelar la base para una cola de caballo. Esta base está conformada por un cilindro tal que un extremo es más ancho que el otro (Figura 14.3).

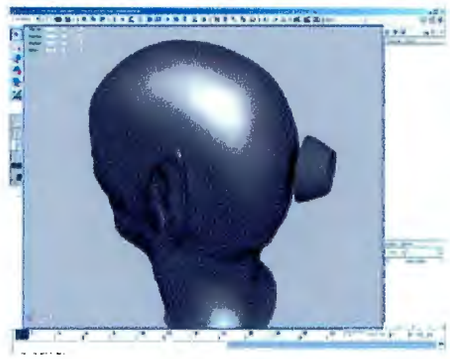

Fig. 14.3 Base de la cola de caballo

Observe el modelo y determine que dirección de crecimiento para el cabello le gustaría. Dibuje curvas NURBS para dibujar el "flujo" del cabello, estas curvas servirán como guías. Por ejemplo en la Figura 14.4 el cabello crecerá hacia atrás.

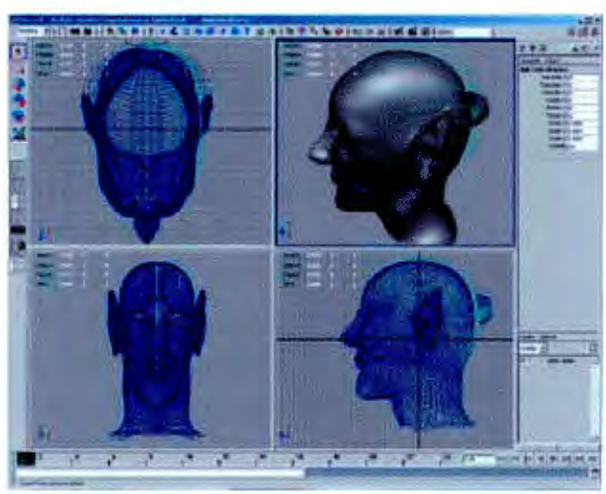

Fig. 14.4 Curvas guía (en verde) para el cabello

El próximo paso es colocar la primera zona de cabello. Como posicionar el cabello específicamente donde se desea requiere de un manejo extraordinario del cepillo de PaintFX [42] para que siga la curva, este posicionamiento se realizará a tramos.

En maya existen cepillos de PaintFX para pintar cabello. Antes de seleccionar alguno de ellos, se debe configurar el cráneo para que se pueda pintar sobre él. Seleccione el cráneo y haga clic en el menú "paint effects menu > make paintable". Seleccione la malla base y especifíquela como plantilla. Seleccione el cepillo de cola de caballo. Dibuje una curva a lo largo de la parte superior del cráneo donde desee que estén las guías (Figura 14.5).

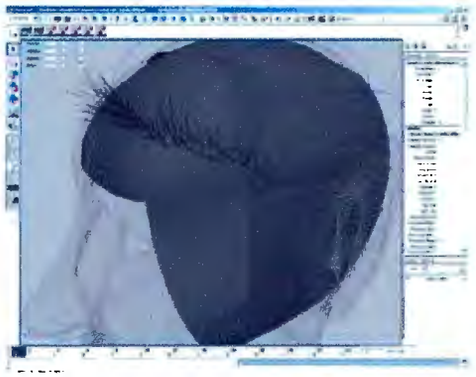

Fig. 14.5 Resultado de dibujar sobre la parte superior de la cabeza

Por el momento, el cabello está en dirección hacia arriba. Selecciónelo y agregue las curvas de control en esa selección. Con todas las curvas seleccionadas diríjase al menú "paint effects > curve utitities > set control curve. Esto provocará que el cabello se estire y fluya en la dirección de las curvas de control (Figura 14.6).

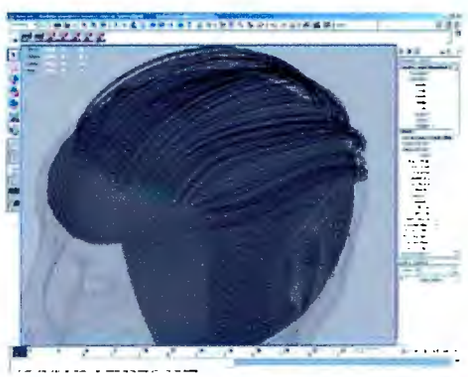

Fig. 14.6 Cabello siguiendo la dirección de las curvas de control

Note que existen huecos a través de los cuales se puede observar el cuero cabelludo. Para eliminar los efectos de esos huecos puede agregar un mapa de textura del color del cabello.

Agregue tantas curvas y dibuje tantas veces como sea posible hasta obtener una densidad de cabello adecuada.

Para crear la cola de caballo, dibuje 4 curvas de control para controlar la forma en la base, en la punta y en los lados (Figura 14.7).

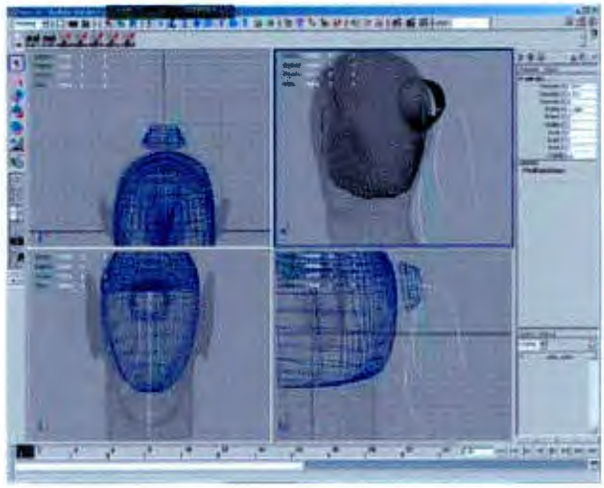

Figura 14.7 Curvas de control para la cola de caballo

Es necesario una curva para crear cabello sobre ella, para esto convierta un círculo de NURBS en una superficie plana. Coloque la superficie obtenida detrás de la base de la cola de caballo y dibuje una curva de PaintFX sobre ella. El resultado se observa en la Figura 14.8

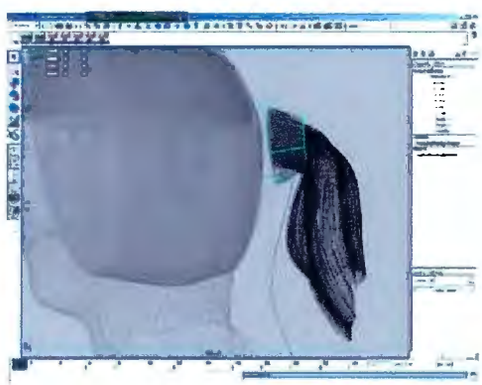

Figura 14.8 La cola de caballo creada con curvas de PaintFX

Por último genere la imagen del cabello para obtener los resultados finales (Figura 14.9).

Fig. 14.9 Peinado final con cola de caballo

15. APÉNDICE C. ANTIALIASING

Uno de los problemas en los gráficos por computadora es el aliasing o efecto de dientes de cierra. Para reducir sus efectos se han diseñado métodos de los cuales algunos se describen a continuación.

Aumento De Resolución

Para reducir el aliasing es necesario crear el efecto de tener más píxeles en la pantalla. El aumento de resolución es el método más práctico para solucionar este problema [36]. El tamaño de los artefactos nunca es más grande que el tamaño del píxel, por lo tanto si se reduce el tamaño del píxel se reduce el tamaño de estos artefactos. Este método a veces no es adecuado ya que tal vez el usuario esté usando la resolución máxima soportada por la pantalla o la aplicación limita la resolución por sí misma. Mas allá de estos limites la única solución es aumentar la resolución efectiva. La mejor forma de hacerlo es usar técnicas más sofisticadas para calcular el color de cada píxel de la pantalla en cierta forma para simular que se tienen más píxeles.

Buffer De Acumulación

En este método [39] en vez de duplicar la resolución de la imagen, se tiene un buffer con una misma resolución a la resolución original de la imagen y con más bits de color que la imagen original. Para obtener un muestreo 2x de la escena, 4 imágenes son generadas con la posición de la cámara movida medio píxel en la dirección x o y de la pantalla según sea necesario. Estas imágenes son sumadas en el buffer de acumulación. Antes de ser desplegada, la imagen obtenida es promediada (en este caso dividida entre cuatro). Los búferes de acumulación son parte de OpenGL [39] y son soportados por hardware gráfico.

Súper Muestreo

Los algoritmos de antialiasing involucran el muestreo del contenido de cada píxel en locaciones múltiples, es decir, el color es calculado en más de una locación dentro del área cubierta por el píxel. Los resultados de estas muestras son combinadas para determinar el color final del píxel. Estas muestras son esencialmente los píxeles extras usados para aumentar la resolución efectiva de la imagen. Si el contorno de un objeto está parcialmente dentro del área de un píxel, su color y el color de otro objeto que ocupa

parcialmente esa área son utilizados para calcular el color final. El resultado son transiciones suaves entre una línea de píxeles y otra línea de píxeles a lo largo de los contornos de los objetos donde el aliasing es más notorio.

El super muestreo es usado en los procesadores gráficos de Nvidia y otros fabricantes. La forma en que estos procesadores implementan esta técnica es con el enfoque de fuerza bruta, es decir, estos procesadores generan la imagen en una resolución mucho mayor que la resolución actual de la pantalla, y después escala y filtra la imagen a la resolución final antes de ser enviada a la pantalla, este escalamiento y filtrado puede reducir el rendimiento [43]. Existe una variedad de métodos para llevar a cabo esta operación, pero cada uno necesita que el procesador gráfico genere tantos píxeles adicionales como lo requiera el método de super muestreo.

El grado de escalamiento en un modo específico de super muestreo es identificado por la relación entre el número de píxeles sin escalar y el número de píxeles en la imagen final. Por ejemplo, en el súper muestreo 2x se generan dos veces más píxeles de los que son requeridos sin antialiasing. 4x genera cuatro veces más píxeles. Note que si se mide el rendimiento en cuadros por segundo, si el procesador gráfico genera cuatro veces más píxeles, entonces el rendimiento será de un cuarto. De hecho la caída de rendimiento puede ser mayor ya que es necesario contemplar la etapa de escalamiento.

Muestreo Múltiple

El muestreo múltiple soluciona los problemas de rendimiento del super muestreo. En vez de generar imágenes en resoluciones mayores, en el muestreo múltiple se calcula el porcentaje de cobertura de un polígono sobre la cuadrícula que forman los píxeles, es decir, se toman más de una muestra por píxel por pasada y se comparten los cálculos entre las muestras[44].

Para obtener estas muestras es necesario un patrón de muestreo, posteriormente las muestras son ponderadas con base a ese patrón para obtener el color final del píxel. En la Figura 15.1 se ilustran diferentes patrones de muestreo [43] donde los puntos representan las muestras, las cuales son ponderadas dependiendo del patrón de muestreo.

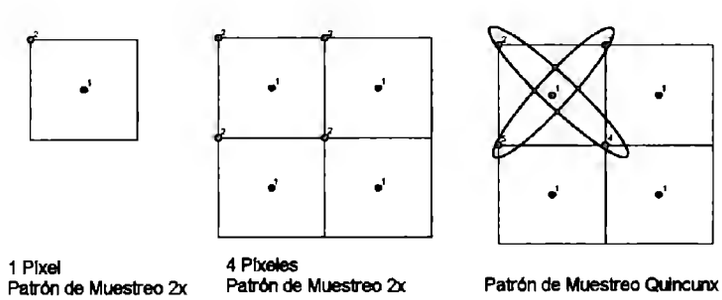

Fig. 15.1 Patrones de muestreo

El muestreo múltiple es soportado por hardware en procesadores gráficos como la familia Geforce 3 o superior de Nvidia. De esta forma, el procesador gráfico obtiene múltiples muestras que serán usadas para calcular el color final del píxel.

Quincunx

El término "Quincunx" se refiere a la distribución de cinco objetos, cuatro en cada esquina de una cuadrado y uno en el centro [44]. En este esquema, el patrón de muestreo es un quincunx, con cuatro muestras en cada esquina de la celda del píxel y una muestra al centro (Figura 15.1). Cada valor de muestra que está en las esquinas es distribuido entre sus cuatro píxeles vecinos, por lo tanto en vez de ponderar igualmente cada muestra, la muestra del centro tiene una ponderación de ½ y las otras de 1/8, por lo tanto un promedio de dos muestras son requeridas por píxel.