

# Single-View Hair Modeling Using A Hairstyle Database

Jose Jaita<sup>1</sup>

**Abstract**—En este trabajo se replica el método propuesto por el paper escogido. En el paper se presenta un nuevo metodo para digitalizar peinados 3D completos y muy complejos a partir de una unica foto de referencia. Primero se contruyo de forma manual una gran base de datos modelos de peinados de varios repositorios en línea. Luego a partir de una simple imagen se modela el peinado, aqui se introdujo varios metodos de modo que los hilos de cabello parezcan lo mas real posible.

## I. INTRODUCCIÓN

Con el espectacular aumento en el rendimiento de las tecnologías gráficas actuales, los detalles visuales de humanos digitales en juegos, mundos virtuales en línea y aplicaciones de realidad virtual se vuelven significativamente más exigentes, alcanzando casi la calidad y el realismo que se observa en la producción cinematográfica. Si bien las animaciones y las representaciones convincentes ahora son posibles en tiempo real, la digitalización de avatares virtuales reales sigue siendo aún un reto, que incluye equipos de hardware complejos y artistas digitales con talento. Con el objetivo de escalar la producción y llevar la creación de personajes 3D a los consumidores, se han dedicado importantes investigaciones a la digitalización automática de rostros humanos [1][Blanz y Vetter 1999], cuerpos [2][Zhou et al. 2010], y manos [3][Wang y Popović 2009] de una sola imagen de entrada o video. Estos métodos de reconstrucción tridimensional generalmente se basan en modelos paramétricos de baja dimensión que codifican variaciones de forma estadísticas de una población.

Un reto que se presenta es la digitalización del cabello, el cual viene hacer un reto bien grade debido a la complejidad de su naturaleza, un sin fin de formas de peinados y factores externos que afectan como la gravedad, viento, torsion, etc.

El método se basa principalmente en una base de datos, la cual contiene peinados canónicos, aproximadamente 300 peinados, los cuales fueron creados manualmente y publicados disponible en algunos sitios web [4][Electronic Arts 2014; Newsea 2014], a partir de trazos generados por el usuario se busca los peinados de la base de datos que tengan mayor coincidencia, luego de mejora el renderizado de los hilos de cabello para que se parezca mas a la entrada y que cumpla cierta logica con respecto a variables externas como la gravedad, fricción, etc.

De todo el pipeline propuesto por el paper, en este trabajo se implementa a partir de tener ya un modelo de

peinado, si bien aún contamos con una imagen de entrada como referencia, pero esta se usa para aplicarle propiedades al peinado que se renderizará, tales como efectos de torsion, gravedad, etc.

## II. EXTRACCIÓN DE CARACTERÍSTICAS

Nuestra entrada es una imagen con un peinado, entonces lo siguiente que haremos será extraer características para aplicarle a un peinado de la base de datos.

### II-A. Segmentación

La primera parte de esta etapa es la segmentación, para esto nos basamos en el metodo propuesto en [5][Ceron-Guzman,2016]. Convertimos a grises la imagen de entrada, luego le aplicamos una binarizacion, pero para esto es necesario un buen umbral para decidir que pixeles seran blancos o negros segun su intensidad, el metodo de otsu calcula automáticamente el umbral de imagen basado en clústeres. El algoritmo supone que la imagen contiene dos clases de píxeles después del histograma bi-modal (píxeles en primer plano y píxeles de fondo), luego calcula el umbral óptimo que separa las dos clases para que su dispersión combinada sea mínima. En la Fig 1 se muestra el resultado de aplicar el metodo de [6]Otsu, lo que sigue es extraer la parte esencial de la imagen de entrada, para esto aplicamos una esqueletización la cual se basa en una serie de dilataciones y erosiones de modo que obtengamos el esqueleto de la imagen de entrada, la Fig 1 muestra el resultado de la esqueletización.



Fig. 1. Izquierda: entrada, Derecha: esqueletización

### II-B. Modelamiento mediante helicoides

Una heicoide es una curva en 3D la cual tiene curvatura y torsion constante. La ecuación de la helicoide que pasa a travez del origen de coordenadas y es paralela al eje-y es la que se muestra en la Fig 2.

\*Este trabajo es patrocinado por CONCYTEC

<sup>1</sup>J. Jaita, estudiante de la maestría en ciencias de la computación, Universidad Católica San Pablo, jose.jaita@ucsp.edu.pe

$$H_{\text{Prim}}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} a(\cos(t) - 1) \\ bt \\ a \sin(t) \end{pmatrix} \quad \text{with } a \geq 0 \text{ and } b \geq 0.$$

Fig. 2. Ecuacion parametrica del helicoides

donde  $a$  y  $b$  son respectivamente el radio y la pendiente del helicoides, para modelar la señal usamos ecuaciones parametricas, ya que con estas no tenemos la restricción de las funciones normales. Para esta parte nos basamos en el paper [7]Cherin et al. [2014], el cual consta de hallar los puntos de mayor curvatura y a partir de ahí es que a través de múltiples helicoides se puede modelar el esqueleto de la imagen de entrada. Los puntos de mayor curvatura vs vienen hacer aquellos puntos en donde la pendiente es formada por el angulo de 90 grados, es decir paralelo al eje-y.

Una helicoides proyecta en el plano XY viene hacer una funcion senoidal la cual posee dos caracteristicas, la amplitud y la frecuencia, entonces para una curva necesitamos estimar la amplitud y la frecuencia de una sinusoidal de modo que se ajuste mejor a dicha curva.

En el paper original toma como entrada a una serie de vertices, en cambio nosotros recibimos posiciones de pixeles del esqueleto, para una mejor notación a continuacion llamaremos a las posiciones de pixeles como vertices.

Para estimar los coeficientes usamos dos vertices, el primero es el vertice vs y el segundo es un vertice vecino (vn). Luego aproximamos la helicoides proyectada con el esqueleto, ahora notemos que  $t$ (parametro independiente de las ecuaciones parametricas) es desconocido para vs, no sabemos para que valor de  $t$  se cumple la maxima curvatura, entonces lo que haremos sera alinear el vs con la maxima amplitud de la sinusoidal y el vertice vn con el origen de coordenadas. A esto se le conoce como el alineamiento de coordenadas locales, nosotros suponemos que el helicoides proyectado pasa a través de vs y de vn, en particular, vn es localizado con  $\alpha = 0$  y vs con  $\alpha = 90$ ;

### II-C. Aproximacion de la curva con multiples helicoides

El esqueleto es dividido en segmentos según el número de puntos de máxima curvatura como se muestra en la Fig 3, luego cada segmento sera modelado mediante una helicoides. El proceso de hallar las helicoides se hace de forma iterativa, el ajuste de una helice comienza donde termina la anterior, la salida es un conjunto de helicoides  $\{H_1, H_2, \dots, H_n\}$  proyectadas en el plano XY. Esto quiere decir que el sistema de coordenada local para la helicoides  $H_i$  comienza donde termina  $H_{i-1}$ .



Fig. 3. (a) entrada, (b) helicoides

## III. RENDERIZACION EN OPENGL

### III-A. Modelo de cabeza

El paper provee una malla de triangulos que corresponden a un modelo de cabeza, este sera nuestro modelo en donde colocaremos los peinados, el archivo es un obj, en el obj nos da informacion de los vertices, su posicion y su topologia, se tuvo que hacer un ajuste a los vertices debido a que la distancia entre vertices que pertenecian a un misma malla era muy pequeña, entonces se hizo un re-escalamiento y una translacion al origen del sistema de coordenadas.

### III-B. Peinados

Los peinados se encuentran en la base de datos dada por el autor, cada peinado consiste en 10000 hilos de cabello y cada hilo consta de 300 vertices, al igual que en el modelo de cabeza se hizo un re-escalamiento y una translacion de modo que tanto el modelo de cabeza y el peinado esten perfectamente alineados.

## IV. MODELAMIENTO DEL PEINADO

Una vez obtenido el conjunto de helicoides, pasamos a modelar nuestro peinado, el paper originalmente usa una mapa de orientación, el cual se basa en extraer características locales de la imagen de entrada, pero nosotros asumimos que ya tenemos el peinado y que simplemente le aplicaremos las mismas propiedades del esqueleto a todos los hilos del peinado.

Como cada hilo se compone de 100 vertices, entonces tenemos que repartirlo equitativamente entre todos las helicoides, por ejemplo si tenemos 5 helicoides entonces tendremos 5 segmentos cada uno con 20 vertices para poder muestrear la helicoides. En esta parte se usa las ecuaciones parametricas, para modelar cada segmento usamos sus respectivos coeficientes previamente calculados.

Como la helicoides fue calculado en un sistema discreto, donde el minimo paso es el pixel y en OpenGL si tenemos dos vertices y trazamos una recta, aquí pueden entrar practicamente infinitos vertices, entonces se tuvo que hacer una adaptacion entre distancia de pixeles y vertices en OpenGL, de este modo se guarda cierta relacion entre la imagen de entrada y el peinado renderizado.

Si solo modelamos un hilo en el plano XY tendremos un resultado como se muestra en la Fig 4 si bien su proyeccion en el plano XY, es la correcta en el eje z no guarda relacion ya que se ve como si fuese un hilo recto sin profundidad y esto en la realidad no sucede.

## V. ESTIMACIÓN DE LA PROFUNDIDAD

Como la información que tenemos es en 2D, es casi imposible predecir con exactitud la profundidad de cada hilo, pero basandonos en los factores externos como la gravedad y friccion podemos dar cierta estimacion de profundidad a partir de la curva proyectada, por lo que reusamos los coeficientes para la componente z, de esta forma el renderizado final del hilo tiene mas coherencia con la realidad.



Fig. 4. Imagen sin profundidad

## VI. RESULTADOS

Se probó distintos entradas de peinados y para el renderizado se usó varios peinados de la base de datos, los resultados muestran que hay coherencia entre la entrada y el peinado renderizado.

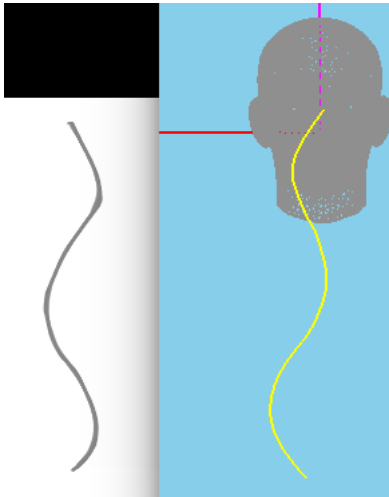


Fig. 5. Imagen de entrada y modelando 1 hilo

### VI-A. Limitaciones

Una de las limitaciones es la memoria RAM ya que la base de datos pesa aproximadamente 4Gb y hacer una búsqueda de un peinado es costoso. Otra limitación que se tiene es que la imagen de entrada tiene que ser bastante clara, ya que sino la esquelitización será errónea.

## VII. CONCLUSIONES

Se digitalizó un peinado a partir de una imagen de entrada, si bien no implementó el pipeline completo propuesto por el paper, se implementaron varias etapas, la esquelitización es fundamental y debe hacerse bien, ya que si al aplicar el procesamiento descrito anteriormente, nos da como resultado dos señales, no se podrá estimar correctamente los puntos de máxima curvatura y por ende las helicoides no modelarán correctamente la curva proyectada. El método que se usó para estimar los puntos de máxima curvatura funcionan bastante bien, siempre y cuando la parte de la esquelitización brinde como máximo una curva. Uno de los problemas que se tuvo aquí fue que si bien se tenía una sola curva algunos

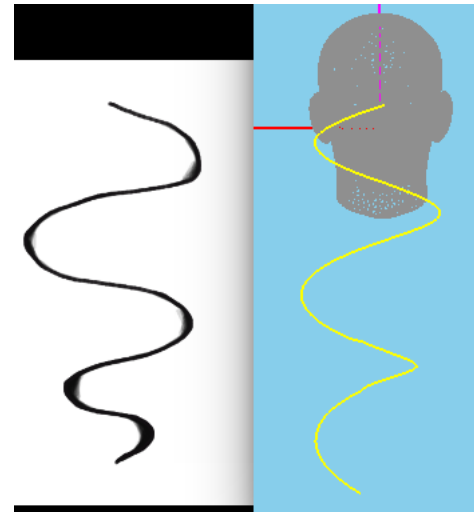


Fig. 6. Imagen de entrada y modelado 1 hilo

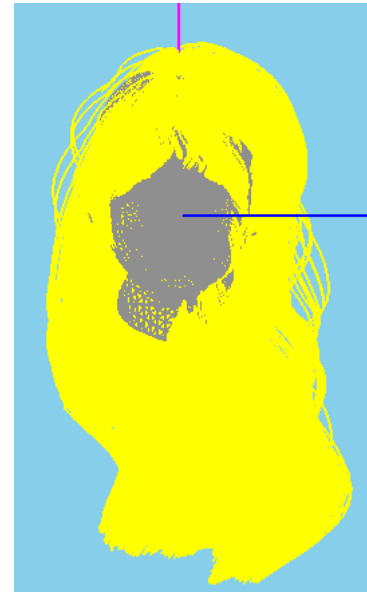


Fig. 7. Imagen de entrada y modelado completo

pixeles estaban desfasados los cuales pueden originar errores a la hora de calcular los coeficientes de la helicoides.

## VIII. TRABAJOS FUTUROS

Para trabajos futuros, para tener un mejor acabado de un peinado, se agruparía conjuntos de hilos en clusters, y los hilos que se encuentran en los límites de los clusters deberían guardar coherencia local.

## REFERENCES

- [1] BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99*, 187–194.
- [2] ZHOU, S., FU, H., LIU, L., COHEN-OR, D., AND HAN, X. 2010. Parametric reshaping of human bodies in images. *ACM Trans. Graph.* 29, 4, 126:1–126:10.

- [3] WANG, R. Y., AND POPOVIC, J. 2009. Real-time hand-tracking with a color glove. *ACM Trans. Graph.* 28, 3, 63:1–63:8.
- [4] ELECTRONIC ARTS, 2014. The Sims Resource. <http://www.thesimsresource.com/>.
- [5] Amar V. Nasrulloh, Chris G. Willcocks, Philip T. G. Jackson, Caspar Geenen, Maged S. Habib, David H. W. Steel, Boguslaw Obara, "Multi-Scale Segmentation and Surface Fitting for Measuring 3-D Macular Holes", *Medical Imaging IEEE Transactions on*, vol. 37, pp. 580-589, 2018, ISSN 0278-0062.
- [6] OenCV, Otsu: <https://docs.opencv.org/3.0-beta/doc/tutorials/imgproc/threshold/threshold.html>
- [7] CHERIN, N., CORDIER, F., AND MELKEMI, M. 2014. Modeling piecewise helix curves from 2d sketches. *Computer-Aided Design* 46, 258–262.