Image-Based Hair Capture by Inverse Lighting

Stéphane Grabli, François X. Sillion, Stephen R. Marschner, Jerome E. Lengyel

Image-Based Hair Capture by Inverse Lighting

Stéphane Grabli, *i*MAGIS -GRAVIR
François X. Sillion, *i*MAGIS -GRAVIR
Stephen R. Marschner, Stanford University
Jerome E. Lengyel, Microsoft Research

Abstract

We introduce an image-based method for modeling a specific subject's hair. The principle of the approach is to study the variations of hair illumination under controlled illumination. The use of a stationary viewpoint and the assumption that the subject is still allows us to work with perfectly registered images: all pixels in an image sequence represent the same portion of the hair, and the particular illumination profile observed at each pixel can be used to infer the missing degree of directional information. This is accomplished by synthesizing reflection profiles using a hair reflectance model, for a number of candidate directions at each pixel, and choosing the orientation that provides the best profile match. Our results demonstrate the potential of this approach, by effectively reconstructing accurate hair strands that are well highlighted by a particular light source movement.

Key words: Hair Reflectance, Hair Modeling, Reflectance Analysis, Shape from Shading.

1 Introduction

More and more computer graphics applications, for instance video games or teleconferencing, require virtual models of people. For this reason, recently, a great amount of effort has gone toward digitizing people. However although hair plays a significant role in a person's appearance, the efficient acquisition of hair geometry remains an important unsolved problem.

Indeed, the usual digitization techniques fail in the face of the complex geometry of human hair, which is an intricate gathering of tens of thousands of thin elements, that are nearly invisible at human scale.

This article presents a method for retrieving the geometry of hair strands by analyzing images. We investigate how to extract as much information as possible from a series of images of a subject's hair, taken under a single known viewpoint and a moving light source. Each lit hair strand reflects light according to its orientation; the idea is then to infer this orientation using a hair reflectance model from the observed images. Although the discussed method is complete as itself, it does not pretend to solve the problem of hair acquisition. It must rather be viewed as an innovative approach to this arduous task, that still needs to be exploited. Furthermore, as we will see in section 6, it must be combined with other more classical approaches to become practical.

After a brief overview of previous work, we outline the approach in section 3. Then, in section 4, we detail a possible implementation. Lastly, in sections 5 and 6, we present our results and conclude with directions for future work.

2 Related work

Although a large body of work deals with modeling [9, 5, 20, 18, 21], animating [9, 5, 18] and rendering [15, 14, 19, 13] human hair, few articles treat the question of its acquisition. Likewise, the extensive research on "Shape from Shading" [3, 16] only addresses the case of relatively continuous surfaces, and doesn't offer techniques suited to hair.

With [17], Nakajima is the only one, to our knowledge, having considered hair modeling from pictures. His approach is purely geometric and consists of building a 3D hair volume from pictures showing the subject's hair from various viewpoints. Hair strands are then generated inside this volume, without any mechanism to ensure faithfulness in their directionality. This simple method presents several limitations: in particular, it seems unlikely to work well on complex hairstyles.

Our method requires a reliable model of reflectance for hair. Kajiya and Kay [11] first introduced a lighting model for hair, to render their "Teddy Bear". It includes two components: diffuse and specular. The diffuse component is derived from the Lambertian model applied to a very small cylinder, considering a hair strand to be lit on the whole half-cylinder facing the light source. Goldman, in [8], improved it by solving its lack of directionality. This improvement is particularly interesting in the case of backlighting simulation. Furthermore, in [1], Banks, also aiming at rendering fur, adds a self-shadowing term to the Kajiya-Kay model. The integration of a shadowing treatment in the model is interesting, but, as we will see in section 4, hardly applicable in our case.

3 Approach

Our approach consists of capturing the geometry of the hair strands geometry by studying images taken under controlled lighting conditions. We chose to observe the

hair from a few viewpoints, and, for each viewpoint, to move the light source along a specific path, taking pictures for many light source positions along this path. The light source locations as well as the intrinsic camera parameters are known and controlled. The images produced are organized in sequences, each corresponding to a single camera position and a specific light source path. Figure 1a shows a sample of pictures from one sequence.

The main hypothesis concerns the chosen reflectance model and the hair material. Our system is a pipeline, taking sequences of hair pictures as input, and producing the geometry of the hair strands as output. We describe each step of this pipeline here in general terms, and provide implementation details in section 4.

3.1 Construction of a Sequence Mask

One of the main goals of this image analysis stage is to detect the hair strands that are best highlighted in the sequence, and to characterize their direction in image space. This step relies on the following assumption: for a given pixel position, all pictures in the sequence show projections of the same hair strand.

First, on each image of the sequence we create a mask indicating the pixels for which strands outlines are the most visible. Each pixel of each mask has an associate vector defined in the image plane. The vector's direction gives the direction of the hair strand projecting onto that pixel, and its magnitude is proportional to the contrast intensity of this hair strand in the picture. We then have, for a given pixel position, a collection of vectors, the size of which is at least equal to zero and at most equal to the number of images in the sequence. All these vectors are assumed to represent the projection of the same hair strand. Figure 1a shows a sample of these masks.

In order to determine the orientation of the hair strand, all the vectors of a single collection must agree. Therefore, in a second step, we identify the relevant vectors in each collection by an election mechanism. Then we extract from each chosen collection a single representative vector and store it into a new mask, which we call the sequence mask. This mask contains the pixels corresponding to the most visible hair strands in the whole sequence, as well as the associated 2D vectors. Figure 1b shows such a sequence mask. A possible implementation for this construction is given in section 4.2

3.2 Construction of a Pixel Profile

The resolution of the input images must be good enough to consider that we have a single hair strand projecting onto one pixel (in order to have only one direction related to a given pixel position). The system's basic idea is that, for a given pixel position, the color sequence observed across the picture sequence can be related to the

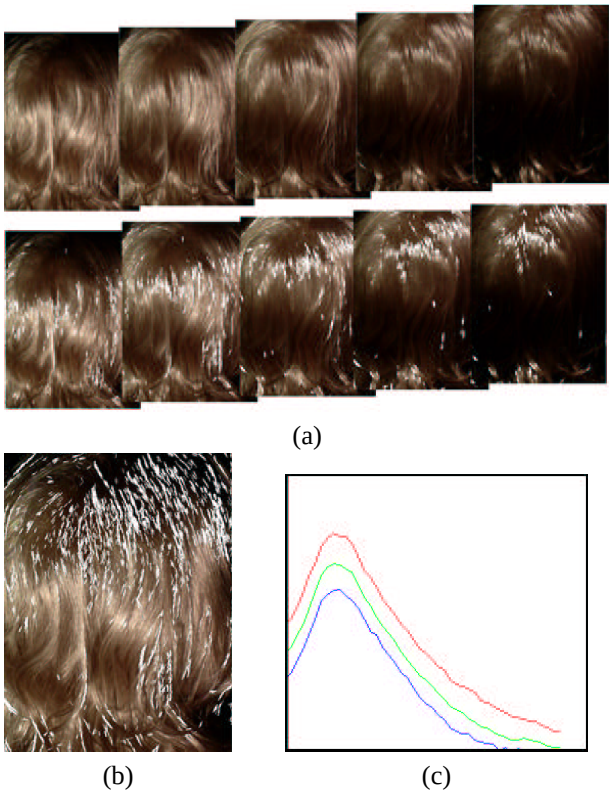


Figure 1: a) For each picture in the sequence, a mask is computed. b) The data set mask. It is displayed by drawing the vector associated to each pixel marked in the mask. c) A measured pixel profile. It shows the RGB components (in picture, R:highest, G:middle, B:lowest) of the colors taken on by on pixel through the sequence.

reflectance map of the hair strand projecting onto that pixel position. The set of these three curves (for red, green and blue) forms what we will call the "measured pixel profile", illustrated in Figure 1c.

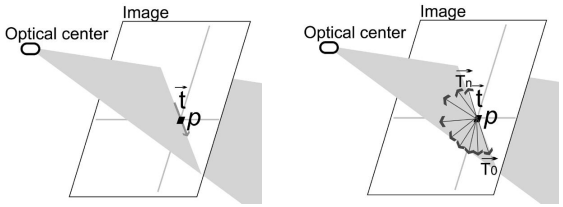
3.3 Computing 3D vectors in the sequence Mask

We now have a set of 2D vectors defined in image space. Each of these 2D vectors is the projection of a 3D vector indicating the orientation of the corresponding hair strand in space. From each 2D vector $\vec{t}$ we want to infer the corresponding 3D vector $\vec{T}$. We use, for that, the geometric information given by the 2D vector and the camera parameters as well as the observed reflectance. This step can be divided into three parts:

Generation of 3D candidate vectors

Let us consider a 2D vector $\vec{t}$. Basic geometric considerations show that the corresponding 3D vector $\vec{T}$ lies in the plane containing $\vec{t}$ and the camera's optical center. This plane's equation can thus be derived from the camera parameters. The idea is then to generate a dense set of 3D

candidate vectors lying in that plane, and finding the one closest to the real hair strand. Figure 2 illustrates this process.



(a) (b)

Figure 2: a) $\vec{T}(p)$ necessarily lies in the plane containing the optical center of the camera and $\vec{t}(p)$ b) A collection of 3D vectors $(\vec{T}_k(p))_k$ is generated in that plane.

Synthesis of Pixel Profiles

Using a hair reflectance model, we compute, for each 3D candidate vector, a reflectance map according to the lighting and observing conditions of the sequence. These maps are stored as pixel profiles, called "synthesized pixel profiles". For each pixel of the mask, we then have a set of 3D candidate vectors, and for each of these 3D vectors, a synthesized pixel profile. Section 4.3 provides more details about the chosen illumination model, and the synthesis of pixel profiles.

Election of a 3D candidate vector

Let us again consider a pixel of the mask. The real reflectance of the hair projecting onto that pixel is represented in the corresponding measured pixel profile. We now search among the synthesized pixel profiles, for the one that looks most like the measured one; the associated 3D vector will provide the hair strand orientation. The analysis of the correlation between pixel profiles curves is detailed in section 4.3 This stage finally extracts a mask of 3D vectors from the sequence mask of 2D vectors.

3.4 Hair Strands

By chaining the resulting 3D vectors, we finally build hair strands.

4 Implementation

4.1 Data Sequence Acquisition

Our approach requires viewpoints and light positions under which the subject is observed and lit, to be known and controlled. Furthermore, we need to keep the subject still for the duration of each sequence acquisition. Concerning the subject's immobility, we decided to work with a synthetic wig[1], so that we could focus on the image processing itself. To control the light source and camera positions, we used the Stanford Spherical Gantry [4] as our acquisition apparatus. It consists of two arms which carry a light source and a camera, and a turntable on which the subject rests. Both arms and the turntable move under computer control, allowing the subject to be viewed and illuminated under practically any configuration.

The camera we used was a 3CCD video camera, and to eliminate noise in the images, we combined images taken using several different exposure times at each position [6].

[1]This allows a proof of concept, but naturally in the long term we will have to work with human subjects.

4.2 Sequence Mask building

Image analysis

The main visible phenomena concerning hair are due to lighting, such as specular reflections or shadowing. We chose to use the Sobel contrast detector [10] to characterize the most visible hair strands in each picture. This gradient-type filter detects contrasts in a particular direction. The result's magnitude is proportional to the contrast. In our case, we apply the filter horizontally and vertically in order to obtain a 2D vector in the image plane. The resulting vector has the direction of maximum contrast, and is therefore perpendicular to the curve delimiting the hair strand projecting there.

We finally get a vector per pixel, with direction perpendicular to the outline and magnitude proportional to the contrast. Pixels showing an insufficient contrast are of low magnitude, and are associated with uncertain directions. Therefore, we set a threshold in order to keep only vectors with high enough magnitude. By rotating all the remaining vectors by 90 degrees, we produce, for each image, a vector field giving the hair strand's directions for high contrast pixels. This step results in a sequence of masks of pixel positions, with a 2D vector associated with each position (see figure 1b). Afterwards, let N refer to the number of images contained in a sequence and p to the (x, y) coordinates of a point in image space. p will be called "pixel" or "position". The mask sequence produced previously is denoted $(L_i(p))_{i\in\{0,\dots,N\}}$. $L_i(p) = 1$ means that the pixel p of image i succeeded the magnitude test, we then say that p is "marked" in L_i. Lastly, the collection of 2D vectors associated with a position p is referred to as $(\vec{t}_i(p))_{i\in\{0,\dots,N\}}$. We consider that $L_i(p) = 0 \Rightarrow \vec{t}_i(p) = \vec{0}$.

Election of relevant vectors

As mentioned before, each collection $(\vec{t}_i(p))_i$ represents the projection of a single hair vector. Although in theory all the vectors of a same collection should be similar, in reality some noise might appear, making difficult the task of representing a collection by a single vector. Our

implementation uses two tests to check each collection's validity. Let us consider the collection $(\vec{t_i}(p))_i$ associated with the pixel position p.

- Collection size We consider that a collection must contain a sufficient number of samples in order to avoid noise and to be relevant. Therefore, we count the number $\mathcal{N}_p$ of masks L_i for which $L_i(p) = 1$ and set up a threshold indicating the minimum number of pictures, $\mathcal{N}_{min}$, in which a pixel p must be marked in order to be selected. Thus, p passes the test if and only if $\mathcal{N}_p = card\{L_i(p), \forall i \in \{0, .., N\}/L_i(p) = 1\} \geq \mathcal{N}_{min}$ This test helps eliminate noise as well as pixels that are shadowed for too many pictures of the sequence, making their pixel profile unusable.

- Homogeneity in the vector directions Each collection is supposed to clearly indicate a single vector. For that reason, this test concerns the direction of the vectors $(\vec{t_i}(p))_i$ associated with a pixel position p. We calculate the angle variance $v_\theta(p)$ of the $\vec{t_i}(p)$ vectors through the whole sequence, and we set a threshold $v_{\theta_{max}}$, to give the maximum variance. So, p passes the test if and only if $v_\theta(p) \leq v_{\theta_{max}}$. For each selected pixel p a single vector $\vec{t}(p)$ is obtained for the entire sequence by summing and then normalizing the vectors of the corresponding collection.

In this way we obtain the sequence mask as well as its associated vectors field $(\vec{t}(p))_p$(see figure 1b).

4.3 Retrieving 3D vectors

Generation of 3D candidate vectors

We showed in section 3.3 how a set of 3D candidate vectors was generated for each pixel marked in the sequence mask. Let $(\vec{T_k}(p))_k$ be the 3D candidate vectors associated to each pixel p of the mask.

Synthesis of Pixel Profiles

The synthesis of reflectance maps using the candidate vectors $\vec{T_k}(p)$, for a pixel p, requires a reliable hair reflectance model. We chose the one introduced by Kajiya and Kay in [11], improved by including the backlighting treatment presented by Goldman in [8]. Letting $\vec{T}$ be the hair tangent unit vector, $\vec{L}$ the unit vector pointing from the hair position P to the light position and $\vec{E}$ the unit vector pointing to the eye from this same position, this hair reflectance model can be written as:

$$\Psi_{hair} = f_{dir} \times (\textit{diffuse} + \textit{specular})$$

where f_{dir} is the directionality term introduced by [8], characterizing the reflection and the transmission properties, where *diffuse* is the model diffuse component,

$$\textit{diffuse}(P) = K_d \times sin(\vec{T}, \vec{L})$$

K_d being the diffuse reflection coefficient and where *specular* is the model specular component,

$$\textit{specular}(P) = K_s \times cos^\alpha(\vec{E}, \vec{E'})$$

K_s being the specular reflection coefficient, E' the vector of the reflection cone the nearest to E, and α the Phong coefficient. Consider again a pixel p, supposing that we wish to compute the reflectance map of the candidate vector $\vec{T_k}(P)$. $\vec{E}$ is the unit vector lying on the ray joining p to the optical center. In order to compute the vector $\vec{L}$, we need to know the hair position P in space. This position necessarily lies on the line joining the optical center to the pixel. If the hair volume was precisely known, we could determine the exact hair position by computing the intersection of this voulme with the line previously defined. For the moment, as we do not have a model of the hair volume, we use as first approximation a semi-ellipsoid, to simulate long hair. This choice does not penalize our result insofar as the distance between P and the light source is much greater than the distance between the real P's position and the approximated one and as thus, the computed $\vec{L}$ is close to the real one. The vector $\vec{L}$ is then set, using the known and light position.

K_d and K_s remain to be determined. They depend on the hair material, whose definition constitutes our second hypothesis. The model used to define a material distinguishes its diffuse, specular and ambient properties. Each of them corresponds to a color, which can be extracted from pictures using image segmentation[2][7]. K_d and K_s are set using the diffuse and specular color thus obtained.

We can then calculate the reflectance profile for each 3D vector, under the experiment's lighting and viewing conditions. We finally have, for each pixel p of the mask, a synthesized pixel profile $f_{t_p}^{(k)}(x)$ for each candidate vector $\vec{T_k}(p)$, $k \in \{0, ..., m\}$. Figure 3 shows a sample of synthesized pixel profiles. It is important to notice that these profiles only consider the interactions between a hair strand and the light source. The interactions with neighboring hairs such as indirect reflections or shadowing, are not taken into account.

Electing a 3D vector by studying correlation

The election of a vector among the candidate vectors is done by studying the correlation between the measured pixel profile and each of the synthesized pixel profiles.

[2]More precisely, using the k-mean algorithm with $k = 3$, we extract three colors from the images. One color stands for a shadowed hair, the second one represents the hair diffuse color and the last one corresponds to the hair specular color.

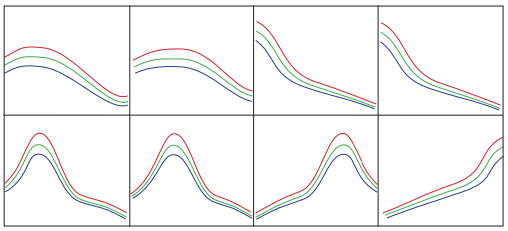


Figure 3: A sample of synthesized pixel profiles (Intensity versus light source position) for different candidate vectors.

Each pixel profile is composed of three curves (R, G and B), and we study the correlation component by component. Because the technique is the same for each of them, we will describe our method for a single component. Let $f_m(x)$ and $f_t(x)$ be the measured and synthesized pixel profiles. In the previous section, we stressed out that the synthesized curves did not include any interaction with the hair's environment. Although inter-reflection is important in the case of hair, the phenomenon which most perturbs our data is the shadowing due to the hair volume itself. Goldman in [8] and Banks in [1] include this phenomenon in their model by specifying a term $\vec{N}.\vec{L}$, $\vec{N}$ being the vector normal to the underlying surface at the hair position, and $\vec{L}$ the vector pointing to the light from this same position. This term requires a precise knowledge of the hair surface. Although this knowledge can be available in the case of hair synthesis, this can hardly be assumed in the analysis case. We first focus on unshadowed regions, leaving the issue of shadowed areas for future work. We identify in the pictures a color s_o, corresponding to hair lying in the shadow. s_o is called the "shadow threshold" and represents an intensity below which a pixel is considered to be in shadow and is not processed. This thresholding is equivalent to setting up shadow maps in the pictures. When we process a measured pixel profile, curve points lying below the threshold are ignored. Valid abscissas form valid intervals I_v:

$$I_v = \{x \in [0, N] / f_{m_p}(x) \geq s_o\}$$

The correlation computation between a measured pixel profile and a synthesized one only involves these valid intervals.

Setting up an Energy function for Correlation study

We now have to determine the most similar synthesized curve, on the valid interval. We do not expect candidate curves to match exactly the measured one, but to be similar in terms of shape. For example, a difference in amplitude is not a valid criterion to dismiss a candidate curve. Classic norms such as L_2 norm would not be adapted to this correlation study insofar as it is not invariant to an amplitude difference. Thus, we need to define a measure suited to the evaluation of curves shape likeness. We use two criteria, dealing with positions of extrema and shape of the curves (first order derivatives). We introduce a distance energy E_{dist}, made of two terms $E_{extrema}$ and E_{shape}, to quantify the similarity according to these criteria.

$E_{extrema}$**: distance energy measured at maxima.** The curves are such that the number of maxima is, except in particular cases, equal to zero or one[3]. Our energy expression applies to two curves, each of them having one maximum. Let x_{max1} and x_{max2} be these two maxima abscissa. We define:

$$E_{extrema} = K_e \times (|x_{max1} - x_{max2}|)^r$$

where K_e is a constant which includes parameters such as the interval's size and insures $E_{extrema} \in [0, 1]$, and r is the polynomial degree. $E_{extrema}$ is defined as a polynomial rather than as a linear curve so as to penalize very distant maxima more than closer ones. $r = 3$ proved to work well in our experiments.

E_{shape}**: distance energy measured on the curve's shape.** E_{shape} is supposed to measure the difference between the shapes of the two curves. Two curves having the same shape show a constant gap between them. We thus chose to first center each curve by substracting its mean value, then to calculate the mean value $\overline{d}$ of the distance between these two centered curves. Let us consider the two curves f_m and f_t defined on the valid interval $[x_{min}, x_{max}]$. $\overline{f}_m$ and $\overline{f}_t$ are the mean values calculated on this interval. Let $\tilde{f}_m(x) = f_m(x) - \overline{f}_m$ and $\tilde{f}_t(x) = f_t(x) - \overline{f}_t$ be the centered curves. We define $d(x) = |\tilde{f}_m(x) - \tilde{f}_t(x)|$, the distance between $\tilde{f}_m$ and $\tilde{f}_t$. E_{shape} is defined as the L_1 norm of $d(x)$:

$$E_{shape} = \overline{d} = \frac{1}{(x_{max} - x_{min})} \times \int_{x=x_{min}}^{x_{max}} d(x)dx$$

Electing and measuring trust

We now select, for each pixel p, the profile with the lowest energy and elect the associated candidate vector. In other words, if we let $f_{dist_p}(k)$ be the value of $(E^{(k)}_{dist}(p))_k$ as a function of k, we choose, for a pixel p, the absolute minimum of $f_{dist_p}(k)$, if it exists. Unfortunately, many cases are problematic. They can be grouped into two families:

1. The cases where determining an absolute minimum for f_{dist} is impossible or ambiguous:

[3] A real continuous function on a compact set admits at least one maximum, but we exclude the case where the maximum is realized on one of the interval's limits, making possible the case for which there is no maximum.

(a) If there is more than one local minimum for f_{dist}, and if the two lowest minima values are too close, choosing one of them is ambiguous.

(b) When f_{dist}'s variation in amplitude is too low, choosing a minimum is not necessarily significant.

2. The cases where measured distance energies are not significant, because of a bad quality of the measured pixel profile:

   (a) If the amplitude of a measured pixel profile is too low, it has a higher noise sensitivity resulting in bad quality data.

   (b) If the valid interval's size is too small, the amount of data used to compare the two profiles is insufficient to produce reliable results.

These different cases are illustrated in figure 4.

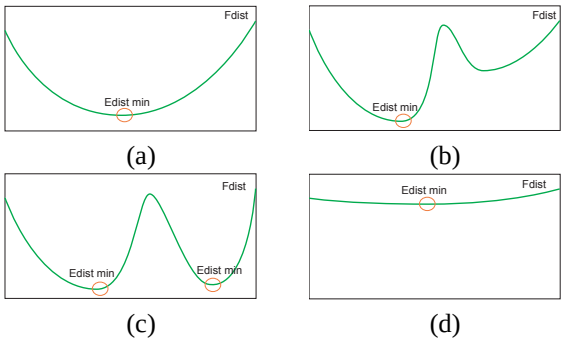


Figure 4: (a) and (b): two curves examples for f_{dist} where the determination of a minimum is not ambiguous. (b) and (c): two cases where this determination is ambiguous or not significant.

It is essential to identify these cases and to have a measure of the certainty with which a candidate vector has been elected. Therefore, we introduce a trust coefficient c_{trust} such that $c_{trust} \in [0,1]$ and $c_{trust} = 0$ in case 1a. In the other cases, its value is proportional to the area of the part of the measured curve located above the shadow threshold and to f_{dist}'s variation in amplitude. Let n_{minima} be the number of minima found for f_{dist}, $E_{dist}^{(min)}$ be f_{dist}'s minimum and $E_{dist}^{(max)}$ its maximum and x_{max} and x_{min} be the upper and lower limits of the valid interval. Then c_{trust} equals 0 if $n_{minima} \neq 1$, and

$$\frac{\lambda}{2} \times \int_{x_{min}}^{x_{max}} (f_m(x) - s)dx + (E_{dist}^{(max)} - E_{dist}^{(min)})$$

if $n_{minima} = 1$. λ is a multiplicative term such that the valid area contribution of f_m is equal to one when this one equals half of $(x_{max} - x_{min}) \times (f_m^{(max)} - f_m^{(min)})$. In practice, we ensure that this contribution never exceeds 1. Thus, for each mask pixel p, we elect a 3D candidate vector $\vec{T}(p)$, and we attribute to it a trust coefficient $c_{trust}(p)$, giving the certainty with which it has been chosen. Figure 5a shows an example of 3D vector sequence.

4.4 Thresholding the 3D vector mask

Since the trust $c_{trust}(p)$ tells how certain each 3D vector election was, it is easy to refine a mask by selecting only vectors having a high enough trust coefficient: $L_{3D} = \{p \in \{0, ..., w \times h\} / L_{2D}(p) = 1, c_{trust}(p) \geq c_{trust}^{(min)}\}$ c_{trust}^{min} being the threshold parameter, and w and h the width and height of the images. With $c_{trust}^{min} = 0.5$, we obtain masks of good quality. Figure 5 shows a 3D vector mask before and after thresholding.

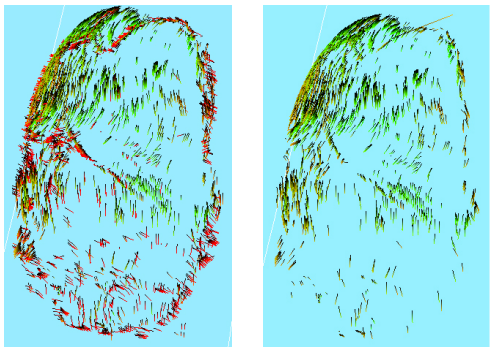


Figure 5: a) 3D vector mask before thresholding. Colors assigned to vectors indicate their trust coefficient (red standing for low and green for high). b) 3D vector mask after thresholding.

4.5 Building 3D strands

We chose to build the final 3D hair strands in two steps. First, we build chains of pixels in image space, using the 2D vector field of the sequence mask and using the "snakes" technique [12, 2]. Each node of each pixel chain is related to a 2D vector, itself associated to a 3D vector (see section 3.3). We finally get the third dimension, in the building process, by using the information given by 3D vectors of each node.

5 Results

Our system allows the extraction of hair strand geometry from picture sequences. The number of pictures in a sequence varies between 48 and 81, according to the light path. Image resolution is 486x720. The number of pixels left in the sequence mask depends both on the chosen viewpoint and light path. Indeed, the better the viewpoint and light path, the greater the number of well con-

trasted hair strands on pictures. For example, a hair strand which is parallel, in image space, to the light path produces good-quality data whereas pixel profiles obtained for hair strands perpendicular to the light path can rarely be exploited.

Figure 6a shows hair strands produced with one sequence, made of 81 images, with a vertical light path. This configuration produces reliable pixel profiles since a high number of hair strands are vertical, and therefore, parallel to the light path. The number of pixels left in the mask of that sequence is about 6,300 pixels before thresholding, and about 3,900 after thresholding. Computing of hair strands for a sequence takes time $O(n \times k \times N)$, where n is the number of pixels marked in the sequence mask, k is the number of 3D candidate vectors per pixel of the mask, and N is the number of pictures in a sequence. Computing of this 2D sequence mask took approximately 9 minutes on a Pentium III 800 MHz with 256 Mb of RAM. The extraction of a 3D mask from this 2D mask lasts about 1 hour.
One can notice that the rebuilt hair strands distribution is quite sparse with one sequence; indeed, one couple viewpoint/light path exploits only part of the hair. However, whatever the direction of a hair strand may be, there is a light source path exploiting it. Thus, by combining the hair strands built from complementary sequences, we expect to reconstruct most of the hair. This combination is straightforward since our method produces hair strands defined in world coordinates. The rebuilt strands that are common to two or more sequences proved to match properly in 3D.

Figures 6b and 6c show hair strands obtained by combining five sequences. We can validate our reconstruction's accuracy using a set of pictures showing the wig from various viewpoints, which ones are different from the viewpoints used for the hair strands building process. Thus, in figure 6, under each picture of rebuilt hair strands is a picture showing the wig from the same viewpoint, proving that the main hair strands directions are correctly recovered. These results are partial inasmuch as other data sequences, with different light source paths, are needed in order to build hair strands on the whole head.

Furthermore, this work serves to demonstrate the accuracy of the reflectance model used for hair. Indeed, the synthesized pixel profiles which were elected match precisely the measured ones and produced, in most of the cases, correct 3D vectors.

6 Summary and future work

We have shown that the method of analyzing reflectance to retrieve shape is a particularly interesting approach to the problem of modeling a specific subject's hair. This work suggests a number of areas for future research, such as:
Developing a complementary geometric method As said previously, our method is rather a contribution to the problem of hair acquisition than its complete solution, and therefore need to be used in addition to other approachs. In particular, our technique shows some limitation in the global positioning of hair strands in space. An ideal solution to the problem of 3D hair model acquisition could result from the joining of a purely geometric method, such as the one presented in [17], together with a reflectance analysis approach like ours. Indeed, a preliminary reconstruction of the hair volume would lead to a significant increase in the accuracy of our results.

References

[1] David C. Banks. Illumination in diverse codimensions. Computer Graphics Proceedings, Annual Conference Series, pages 327–334. ACM SIGGRAPH, ACM Press, July 1994.

[2] A. Blake and M. Isard. *Active Contours*. Springer-Verlag 1998, 1998.

[3] M. J. Brooks and B. K. P. Horn. Shape and source from shading. In B. K. P. Horn and M. J. Brooks, editors, *Shape from Shading*, pages 53–68. MIT Press, Cambridge, MA, 1989.

[4] Custom Scanner Spherical Gantry CyberWare. http://www.cyberware.com/products/sphereinfo.html.

[5] A. Daldegan, N. M. Thalmann, T. Kurihara, and D. Thalmann. An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum*, 12(3):C211–C221, 1993.

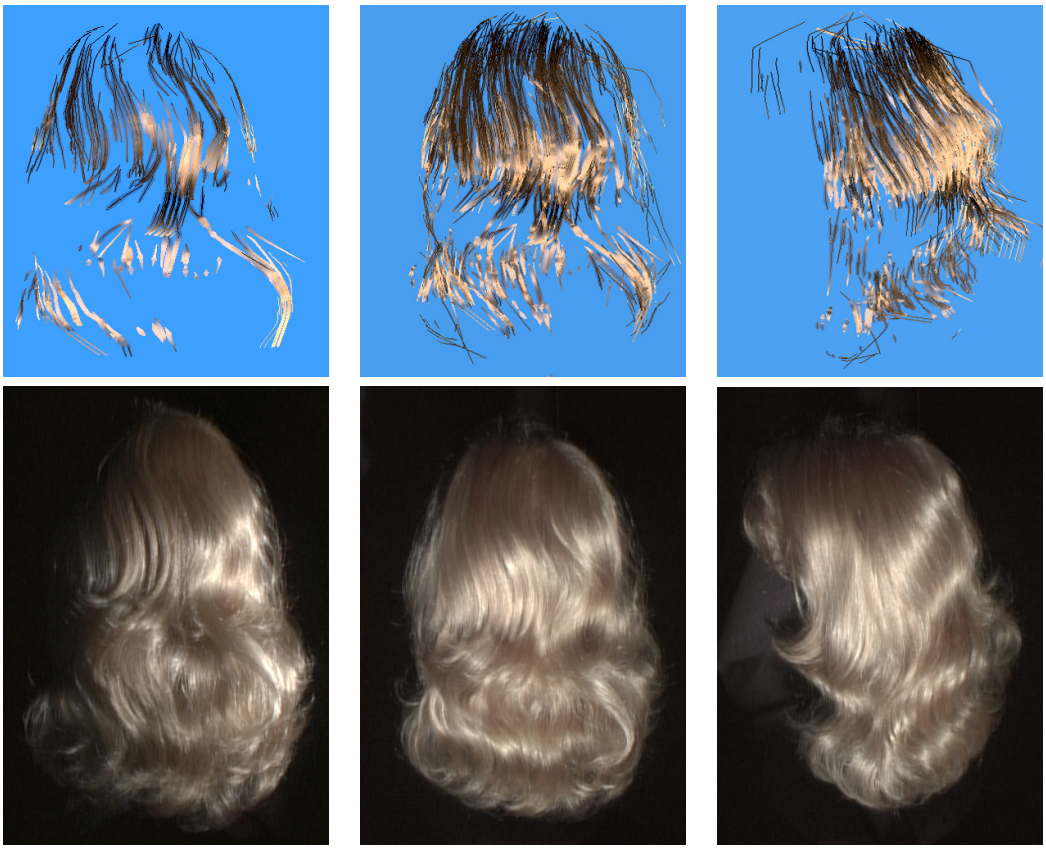
[6] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. *Computer Graphics*, 31(Annual Conference Series):369–378, 1997.

[7] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press, Boston, MA, 1990.

[8] Dan B. Goldman. Fake fur rendering. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 127–134. ACM SIGGRAPH, Addison Wesley, August 1997.

[9] Ken ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. A simple method for extracting the natural beauty of hair. *Computer Graphics*, 26(2):111–120, July 1992.

[10] B. Jahne. *Digital image processing*. Springer-Verlag New York, Incorporated, 1991.



(a) (b) (c)

Figure 6: a) Up: hair strands rebuilt with a single picture sequence and rendered using broad strands for visualization convenience. Down: picture of the sequence (same viewpoint). b) and c) Up: hair strands rebuilt with five sequences, observed from two different viewpoints and rendered using broad strands for visualization convenience. Down: pictures showing the wig from the same respective viewpoints.

[11] James T. Kajiya and Timothy L. Kay. Rendering fur with three dimensional textures. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 271–280, July 1989.

[12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. of IEEE Conference on Computer Vision*, pages 259–268, 8-11 1987.

[13] W. Kong and M. Nakajima. Hair rendering by jittering and pseudo shadow. In *Proceedings of the Conference on Computer Graphics International (CGI-00)*, pages 287–294, Los Alamitos, CA, June 19–24 2000. IEEE.

[14] A. M. LeBlanc, R. Turner, and D. Thalmann. Rendering hair using pixel blending and shadow buffers. *The Journal of Visualization and Computer Animation*, 2(3):92–97, July–September 1991.

[15] Jerome Edward Lengyel. Real-time fur over arbitrary surfaces. In *ACM 2001 Symposium on Interactive 3D Graphics*, 2000.

[16] S. Magda, D. Kriegman, T. Zickler, and P. Belhumeur. Beyond lambert: Reconstructing surfaces with arbitrary brdfs. In *ICCV01*, pages II: 391–398, 2001.

[17] Masayuki Nakajima, Kong Wai Ming, and Hiroki Takashi. Generation of 3d hair model from multiple pictures. In *IEEE Computer Graphics & Applications (12) 1999 Multimedia Modeling'97*, 1999.

[18] R. E. Rosenblum, W. E. Carlson, and E. Tripp, III. Simulating the structure and dynamics of human hair: modelling, rendering and animation. *The Journal of Visualization and Computer Animation*, 2(4):141–148, October–December 1991.

[19] Nadia M. Thalmann, Stephane Carion, Martin Courchesne, Pascal Volino, and Yin Wu. Virtual clothes, hair and skin for beautiful top models. In *Computer Graphics International 1996*, 1996.

[20] Allen Van Gelder and Jane Wilhelms. An interactive fur modeling technique. In Wayne A. Davis, Marilyn Mantei, and R. Victor Klassen, editors, *Graphics Interface '97*, pages 181–188. Canadian Human-Computer Communications Society, May 1997.

[21] Watanabe Y. and Suenaga Y. Drawing human hair using wisp model. In *Computer Graphics International, 1989*, 1989.