Lightweight face relighting

Sylvain Paris François X. Sillion Long Quan
ARTIS† / GRAVIR - IMAG HKUST‡

Abstract

In this paper we present a method to relight human faces in real time, using consumer-grade graphics cards even with limited 3D capabilities. We show how to render faces using a combination of a simple, hardware-accelerated parametric model simulating skin shading and a detail texture map, and provide robust procedures to estimate all the necessary parameters for a given face. Our model strikes a balance between the difficulty of realistic face rendering (given the very specific reflectance properties of skin) and the goal of real-time rendering with limited hardware capabilities. This is accomplished by automatically generating an optimal set of parameters for a simple rendering model. We offer a discussion of the issues in face rendering to discern the pros and cons of various rendering models and to generalize our approach to most of the current hardware constraints. We provide results demonstrating the usability of our approach and the improvements we introduce both in the performance and in the visual quality of the resulting faces.

1. Introduction

Relighting objects taken from images and especially human faces is a very common topic. As soon as one aims at mixing images coming from different sources, the lighting environment has to be taken into account to render a satisfying picture. If this aspect is neglected, it strongly impedes the consistency of the composited image: the different parts of the image look like they have been made separately and just pasted side by side, they never merge into a single picture.

We focus on faces because they are obviously a key feature and their diversity is challenging our approach. One of the main issues is the rendering of skin because of its complex interaction with light through many layers (oil, epidermis, blood,...). Many proposed techniques have targeted rendering precision and ignored efficiency.

Our approach is different: our rendering technique is fast (real time) even on widely available graphics cards and is light enough to allow the card to render other objects. To achieve this, we restrict ourselves to a basic rendering engine. Of course, we do not claim to reach the same realism than a complex model that does not run in real time. Nevertheless we produce images with a high visual quality, superior to typical results in games and similar real time applications. Moreover, portable devices such as cellular phones or PDAs have increasing graphical capabilities and will probably be equipped with 3D chips in the near future. Therefore even when all desktop machines have high performance graphics cards, there will still be a need for techniques running with limited hardware requirements.

Our algorithm starts with the following input data:

- A 3D model of the face to relight.
- A calibrated photograph (*i.e.* with the corresponding projection matrix) of the face taken in a dark room with a flash.
- A light probe image taken in the same conditions.

To acquire the 3D geometric model, we have explored various acquisition methods. We have mainly used a 3D scanner which directly provides a precise mesh. We have also worked with models obtained from Computer Vision techniques, which use only a camera to acquire the data.

Considering all these issues, we propose in this paper a face relighting engine which runs with very low hardware requirements and a robust method which is easily applicable to acquire the data needed for this engine.

Method overview For our rendering engine, we use basic functions which are now supported by all 3D graphics hardware. The global skin appearance is obtained through a simple parametric model. This model renders all the lighting dependent effects such as shading, highlights and can be extended to shadows if needed. Then the details like the skin roughness, imperfections, the eyes and mouth, and facial hair are added by a texture. This pipeline is very simple and requires a small computation time. Nevertheless it is expressive enough to capture most of the important features needed to render faces, as discussed later in the paper. From this rendering engine stem two main issues: computing the

† ARTIS is a research project in the GRAVIR/IMAG laboratory, a joint unit of CNRS, INPG, INRIA and UJF.
{sylvain.paris|francois.sillion}@imag.fr
‡ quan@cs.ust.fr

detail texture and optimizing the parameters for the underlying skin model.

We first review in Section 2 the techniques that are most related to ours. In Section 3, we show how to build the detail texture. We then explain in Section 4 the process to determine the parameters of the underlying model. The rendering engine is described in Section 5. We illustrate our results in Section 6 and conclude in Section 7.

2. Previous work

Many approaches to face rendering have been developed focusing on various aspects: modeling the geometry, acquiring the skin reflectance, building a generic model of such a reflectance, and synthesizing facial expressions. Most of the existing work covers several subjects simultaneously. Since we are concentrating on relighting, we are especially concerned with reflectance acquisition and rendering issues.

Reflectance acquisition This problem is related to the concept of *bidirectional reflectance distribution function* (BRDF) or *bidirectional surface scattering reflectance distribution function* (if we consider that light penetrates inside the skin) which embeds the light interaction with surface materials. Different techniques have been developed either to sample the BRDF [26,27] or the reflectance [5,32]. An interesting result [26] from this research is that skin BRDF cannot be perfectly matched by the classical parametric models. This shows that our method is not physically perfect but since we target visual quality, this is not a major drawback. All these acquisition methods are very accurate but they all require some specific equipment to acquire the data and they handle so much information (2000 photographs in [5]) that they cannot achieve fast rendering. Georghiades et al. [9] have a similar approach to [5]: they also build a pixel-by-pixel model but reduce the amount of information using a linear parameterization of the face picture. However their method is limited by the assumption that a face has a purely diffuse reflectance, which prevents them from rendering highlights.

Jensen et al. [15] have shown that skin should be modeled considering *subsurface scattering* but today it is still unclear how accurate data can be obtained on a given real face (for instance [30] proposes a layered model without giving a means to measure the parameters). And even the fastest rendering [14] is still far from real time.

Therefore, like many others [2, 3, 10, 13, 19, 25, 29, 35] we have chosen a parametric model and we seek a suitable set of parameters. These methods achieve good approximations for common materials. Moreover, the parameters can be robustly computed even with sparse data thanks to the limited number of unknowns to determine. This allows simpler acquisition conditions than in previous sampling methods.

Texture enhancement Because our 3D mesh does not contain all the geometric details to render the skin roughness, the parametric reflectance alone gives unrealistically smooth results. Face features such as the eyes and mouth are also missing. We add all these details using a texture map. Rushmeier et al. [34] have studied the perceptual effect of various situations and ours (*i.e.* precisely texturing a smooth surface) is one of the cases that give the best improvement. The idea of improving a general model with a texture map has been widely used [22, 23, 33, 39] and all these authors indicate that the combination must be multiplicative to preserve the underlying shading.

When face animation is required [25, 40], generic local 3D models are adapted instead of applying a correction to match the input photographs. Even if this technique is useful for animation, its results illustrate the loss of fidelity due to a generic 3D model.

An approach very similar to our method is the method of Wen et al. [42] which use a reference sphere to model the skin reflectance and to remove shading from the input images. But like some others [9, 22, 24], this method assumes that skin is a purely diffuse material. As can be expected, the highlights never move thus impairing the consistency of the relighting.

Complex material rendering Many authors [4,16,18,27, 28, 31, 38] propose to render in real time very complex materials under various lighting environment using hardware acceleration. None of these methods have rendered skin yet but the results are very promising and skin can surely be accurately approximated. The main drawback of these methods considering our goal is that they only achieve real time rendering for a single object even with the latest available hardware.

Debevec et al. [7] propose an original solution to the face relighting problem. They only measure the target lighting and then directly shoot a video of the subject in this environment with an immersive light stage. This hybrid approach is by definition photo-realistic but requires a specific light stage and introduces all the limitations inherent in live shooting.

Contributions

Most of the previous approaches first target precision of the results. Undoubtedly, they achieve this goal, but these methods often rely on such a complex rendering process that they cannot be used in real time or are limited to a single object.

Focusing on rendering performance, we show that a face can be rendered with only a simple reflectance model (Phong) and a detail texture. We propose an original approach to compute the parameters to reach an approximation that conveys high visual quality. Compared to existing methods, our technique introduces a new tradeoff between quality and speed: while being real time, it produces images with convincing lighting effects. The technique is demonstrated to be light enough to allow the graphics card to render the rest of the scene or additional effects. We illustrate this by proposing some improvements which can greatly enhance the results in some cases: over-exposure, eye highlights and cast shadows.

3. Detail texture

As briefly explained in Section 1, a texture map is used to add all the details that are independent of the underlying skin model: the eyes, the mouth, etc. We name it the *detail texture*. Since the model renders all the lighting-dependent effects, the texture should be lighting independent (*i.e.* without shadow, highlight, etc.). Conceptually, to create the texture, we "subtract the shading from the input photograph".

We first need a *reflectance map* [12] of the skin. The entire 3D model is then rendered with this reflectance map. This gives us the information for the "subtraction".

Skin reflectance map We build a specific model of the skin from the input photograph. Compared to the parametric model used by the rendering engine, this model is limited to the input lighting environment and viewpoint but it is more accurate.

A face presents almost all the 3D orientations facing toward the camera. On the sphere of the directions (*aka* the Gaussian sphere), the normals cover the whole front facing hemisphere. Thus, under the approximation that skin follows the same reflectance map on the whole face, the skin reflectance is sampled on the hemisphere with only one image. To know where skin lies in the input photograph, we can either use a segmentation algorithm or ask the user to paint the skin region. Figure 4 shows a sample segmentation made by the user, which only requires a few minutes.

Each skin pixel is associated to its normal on the 3D model. This results in color samples on the Gaussian sphere. Because of the hair, there are fewer samples for the upward directions (and for the downward directions for bearded people). The samples are then grouped into clusters and the color of each cluster is determined through a robust mean that discards outliers: we only consider the samples whose distance to the classical mean is less than the standard deviation. The outliers may result from errors in the 3D model, inaccuracies in the skin segmentation, etc. With the robust mean, all these artifacts are handled without any user intervention.

Using a front facing flash eliminates self-shadowing. This is important because with other lighting condition, coping with shadows would have been tedious (such pre-processing can take hours [38]) and would most probably have introduced more approximations and outliers.

We then extend the cluster values to a dense and continuous reflectance function by interpolation and extrapolation. For a given point P on the sphere, we use a scheme that:

1. For each distance d, averages all the clusters at the same distance d of P to get one value $v(d)$ per distance.
2. Gives a weight $w(d) \approx e^{-d}$ to $v(d)$ to guarantee that only the closest values to P have a significant weight.
3. Compute the mean of the values to get the value of P.

This process slightly smoothes the resulting function and removes spurious high frequencies. This fits the demonstration of Ramamoorthi and Hanrahan [31] who have shown that, since skin has a low frequency BRDF, it has a low frequency reflected radiance distribution even under high frequency lighting like a point light source.

Note that the map (Fig. 5-c) contains both diffuse and specular components because no separation is done after sampling the photograph. Highlights do not suffer from the outlier removal because the viewpoint does not change and they are therefore fixed on the Gaussian sphere.

Ratio image The whole 3D model is then rendered with the same pose as the photograph using the acquired skin reflectance map (Fig. 5-c): it is the *shaded face*. As shown in [22], the combination of a detail texture with a base image should be multiplicative. Therefore, the detail texture (Fig 6-c) is the *ratio image* between the input photograph (Fig 6-a) and the shaded face (Fig 6-b). It can be formally defined by:

$$ratio\ \ image = \frac{input\ \ photograph}{shaded\ \ model}$$

where the operation is done pixel by pixel, RGB-component by RGB-component[1].

Limitations and discussion

The detail texture results from a process that makes several approximations. We discuss here their limitations and validity.

First of all, we assume the skin to be an uniform material over the whole face. Clearly, the skin is not exactly the same on the nose and on the chin. Nevertheless, as it is always composed in the same way there is no huge variation across the face. The small variations are partly corrected in the final rendering by the detail texture. If one looks carefully, the detail texture (Fig 6-c) still exhibits some highlights on the most shiny regions (*e.g.* on the nose). This comes from this approximation because these regions are shinier than the "average" reflectance. However, these remaining highlights have a limited intensity corresponding to the difference with the average.

We also consider that everything is a detail based on the skin reflectance. This is obviously wrong for the eyes, the mouth, etc. The color of these details is so different from the skin color that it induces extreme values in the texture (*i.e.* RGB ratios are far from 1) and both colors are almost decorrelated. These details contain many high frequency features

[1] In the *shaded model*, 0 is replaced by a small value to avoid problems

(iris in the eyes, small wrinkles on the lips, etc.), which hide most of the shading effects. Therefore, although they are not skin details, this approximation does not lead to visible artifacts. The only caveat may be the eye and lip highlights: they are not removed by the ratio image because they are not rendered in the shaded face. They are then rendered by the detail texture independently of the light and the viewpoint. This can impede the overall consistency especially in close-up. This is overcome by removing them from the input photograph by inpainting either automatically [1, 40] or manually.

Since we cannot afford to handle a 3D model precisely up to the skin roughness, the skin textured aspect is rendered by the detail texture. This implies that it is lighting independent whereas it results from the interaction of the light with the micro-relief of the skin surface. This interaction is well described by subsurface scattering [15] but cannot be rendered in real time [14]. Fortunately, contrary to the eye highlights that give strong cues about the lighting environment, these low amplitude variations do not carry much information. Nevertheless, they are a key feature for realism which has to be captured. The main point is to avoid saturation in the input photograph which makes the skin texture disappear in saturated areas. We therefore use a flash of limited power.

4. Parameters of the underlying model

In the previous section, we have built a texture which embeds all the lighting-independent parts of the face. We now explain how to manage the lighting-dependent part.

This step must be efficient and lightweight while being expressive enough to match the main characteristics of the skin as seen in the input photograph. As explained in Section 2, these constraints lead to a parametric model like those proposed by Phong, Torrance-Sparrow [41] or Lafortune [17].

In this paper we have chosen the classical Phong model in the RGB space because it is implemented in all 3D cards and therefore achieves the best performance. In the following paragraphs we focus on this model but a similar study can be done on any other parametric model.

This model is controlled by 10 parameters for the material: the RGB values of ambient, diffuse and specular colors and the shininess exponent ^{s} and 9 parameters for the light: the RGB values of ambient, diffuse and specular colors. If $\mathbf{v}$, $\mathbf{l}$, $\mathbf{r}$ are the view, light and reflection directions and $\mathbf{n}$ the surface normal, the intensity of component X (R, G or B) is:

$$I_X(\mathbf{v}) = X_\mathrm{l}^\mathrm{A} X_\mathrm{m}^\mathrm{A} + X_\mathrm{l}^\mathrm{D} X_\mathrm{m}^\mathrm{D}\, \mathbf{l}\cdot\mathbf{n} + X_\mathrm{l}^\mathrm{S} X_\mathrm{m}^\mathrm{S}\, (\mathbf{r}\cdot\mathbf{v})^s$$

where superscript indicates "ambient", "diffuse" or "specular" and subscript "light" or "material".

Our strategy is to match as closely as possible the input photograph with the model. We first explain how to recover the lighting parameters and then the skin parameters.

4.1. Lighting parameters

The input photograph is lit by a flash that is approximated by a point light source. Its characteristics are determined with the probe picture. The light position is computed thanks to the highlight position on the probe. In the (θ, ϕ) spherical coordinate system, we have $(\theta_l, \phi_l) = (2\theta_h, 2\phi_h)$ where l stands for the light and h for the highlight. In practice, the angular deviation is about 2 degrees.

We then look at the light color. It cannot be directly reached in the highlight: it is always saturated because of the direct reflection in the mirror. But as the mirror BRDF is never a ideal peak, there is always a halo around the highlight. This halo does not saturate the camera sensors and can be used to measure the light *hue* (*i.e.* the proportion between R, G and B). This method gives no information about the color intensity. Even *high dynamic range* imaging [6] could hardly capture this information because it need specialized instruments (professional flash, filters, etc) which do not fit within our goals. The intensity is arbitrarily set to an approximate value, which means that the light intensity is now expressed in relation to the input photograph. In practice, with a good setup which makes the photographs bright but not saturated, one is a satisfying value. However, to work with different flash lights, one must take photographs of a white probe (*i.e.* a white mate ball) to calibrate their relative intensity.

The ambient lighting is measured by integrating the light samples in a ring surrounding the highlight halo. The whole spherical probe is not used to guarantee that we have no outliers (mainly from the support of the probe). Since there is no saturation, both hue and intensity are recovered. However to be fully consistent with the flash light, their relative intensities have to be adjusted. Unfortunately, the ambient contribution is so low that it cannot be used as a precise reference to set the flash intensity. The ambient intensity is therefore optimized relatively to the flash with a gradient descent performed once the skin parameters are known. The resulting variation is almost unnoticeable but it makes the set of parameters consistent.

4.2. Skin parameters

As suggested in [36] and also [5], we consider that the specular reflection introduces no spectral distortion in the light; it is directly reflected by the upper oily layer of the skin. This means that the specular color is (i_s, i_s, i_s). Only its intensity i_s remains unknown. Diffuse and ambient colors are due to a deeper interaction between the light and the skin. Because these colors result from the same cause, they have the same value $(r_{skin}, g_{skin}, b_{skin})$. With the shininess exponent, s, there are 5 unknowns. The strategy to match the input photograph as well as possible is to make these values evolve with an optimization process.

Unfortunately, there is a strong ambiguity due to the specular part of the model. If the shininess exponent is low, the specular component is flat and thus equivalent to the ambient component. High values of the exponent make the specular component concentrated only on a tiny area and therefore equivalent to no specular reflection. Optimizing the exponent at the same time as the other parameters makes the whole process unstable due to numerous local minima. s is therefore determined separately before the other parameters.

Specular exponent The idea is to formalize the intuitive remark that a high exponent corresponds to a small highlight and a low exponent to a wide one. We propose a method that measures the "size" of the highlight and relates it to the Phong exponent s. This computation cannot rely on a full diffuse-specular separation because it occurs before the determination of the corresponding parameters. Therefore it implies an indirect estimator.

As explained in Section 4.2, each color in the reflectance map is a mix between the light and skin colors. Since the flash is bright and almost white and the skin is darker and colored, color saturation is a good indicator of the mix proportions. To measure the highlight "size", the regions with the fastest saturation changes. This is a robust criterion as it does not depend on the absolute values of the flash and skin saturations and does not require a full separation between the diffuse and specular parts.

The highlight position on the reflectance map is known thanks to light position previously computed. Since the sampling resolution is finer near the equator region (Sec. 3), the angular deviation of the fastest variations are measured toward the left and right directions. These two values are averaged to obtain a robust estimation of the angle α_{sat}^{max}.

We then study the Phong model. We compute the specular intensity $I_{flash}(\alpha)$ and diffuse intensity $I_{skin}(\alpha)$ relative to the angle α, as well as their ratio $\rho(\alpha)$. α_I^{max} is the angle where the ratio has its most important change. We conclude with the argument that the largest saturation change is linked to the largest change in the ratio between the specular and diffuse components. Hence we have:

$$\alpha_{sat}^{max} = \alpha_I^{max} \tag{1}$$

To derive the expressions, we make some approximations: the view and light directions are constant on the face

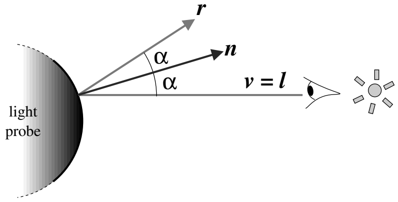


Figure 1. Lighting configuration for highlight measurement. View direction v and light direction l are considered equal and constant on the face.

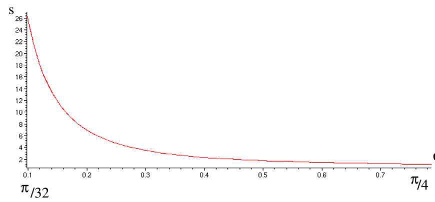


Figure 2. Plot of relation (3).

(the variations are at most 2.5 degrees in our experiments) and equal (the difference is at most 2 degrees in our experiments), and the ambient intensity is neglected compared to the diffuse one in the reflectance model (this is reasonable because the input photograph is made in a dark room). Using the notation of Figure 1 and $\sim$ standing for "proportional to", $I_{skin}(\alpha) \sim \cos(\alpha)$ and $I_{flash}(\alpha) \sim \cos^s(2\alpha)$ which gives the ratio:

$$\rho(\alpha) \sim \frac{\cos^s(2\alpha)}{\cos(\alpha)} \tag{2}$$

And since α_I^{max} corresponds to the largest variation of $\rho(\alpha)$, it implies:

$$\frac{d^2\rho}{d\alpha^2}(\alpha_I^{max}) = 0$$

which leads with equations (1) and (2) to the following relation where $\zeta = \sin(\alpha_{sat}^{max})$:

$$s = \frac{4\zeta^4 - 2\zeta^2 - 1 - \sqrt{-16\zeta^6 + 8\zeta^4 + 1}}{8\zeta^2(\zeta - 1)(\zeta + 1)} \tag{3}$$

Relation (3) fully determines the exponent s with α_{sat}^{max} measured on the reflectance map. It is plotted in Figure 2. As the slope is very steep, precision is crucial. We therefore use a high angular resolution of about 1 degree.

To validate the technique, a photo of the same person has been scanned with different skin conditions. As illustrated in Figure 8, numerical results agree with the images. The results are best seen on video.

Other parameters For the four remaining parameters, we run an optimization process which minimizes the difference between the input photograph and an image of the 3D model rendered with these parameters. Since we target the best possible visual match, the perceptual *Luv* space is used to quantify the difference between both images. The comparison is only made on the skin region using the classical L_2 norm:

$$\sqrt{\sum_{skin} ((L_i - L_r)^2 + (u_i - u_r)^2 + (v_i - v_r)^2)}$$

where i is for the input photograph and r is the rendered 3D model.

The minimization is done through a varying step gradient descent which performs a coarse-to-fine refinement of the parameters. Since gradient descent techniques often suffer from local minima, we have validated its robustness by running the process 200 times on the same input data with random starting points. The standard deviation on the parameters is less than 2.5%. This shows that the process is almost insensitive to the starting point, a "good" initial guess only speeds up convergence.

Figure 3 gives an overview of the entire process leading to each parameter needed by the rendering engine.

5. Implementation of the rendering engine

Our rendering engine is based on OpenGL but it can be easily implemented on any platform that provides basic 3D rendering functions. We have restricted the set of functions to the most basic ones to ensure a very low hardware dependency. In particular, we have not used the recent programmable pipelines based on *vertex programs* and *pixel shaders* because the available functions still vary a lot according to the graphics card. Nevertheless, since these features are extensions of the basic ones, one can implement our engine using the latest improvements to get better performance but the engine will be then reserved to the latest cards.

Basically, our engine performs first a Phong rendering (`GL_LIGHTING`) and then a multiplicative texture mapping (`GL_MODULATE`). Unfortunately, a straightforward implementation is not enough for our purposes. Details brighter than the underlying skin color (the eyes for instance) require a multiplicator higher than 1 whereas a basic texture can only store values in $[0,1]$. To work around this constraint, we use two passes: the first one uses a texture with halved values and the second one is drawn without texture nor lighting but with a $(1,1,1)$ color and the blending equation `glBlendFunc(GL_DST_COLOR,GL_ONE)` which results in doubling the pixel values. The available range is then $[0,2]$. The method can be extended to $[0,2^n]$ with a texture multiplied by 2^{-n} and n doubling passes. Note that each of these doubling passes divides the color precision by two: the multiplication by 2^{-n} rounds down the values losing n bits of precision. In practice, up to $n=2$ results are visually similar and $n=3$ produces an almost unnoticeable alteration, which makes multiplicator values up to 8 available without visible loss of quality.

Since our rendering engine is very lightweight, it can support various improvements to enhance the final visual quality. We present some of them in following sections.

Camera sensitivity When someone is photographed under the sun or with a strong flash, some regions in the picture are poorly exposed. Parts of the face that are in the shadow are often under exposed and those directly in the light are over exposed. Technically, this comes from the limited range of sensitivity of the sensors compared to the intensity range in the observed scene (see [6] for details). The effects are characterized by the disappearance of the details either in the darkness (underexposure) or in the highlights (overexposure).

Underexposure is straightforwardly obtained by lowering the light intensity. Overexposure can be simulated by clamping to 1; this corresponds to a simple sensor model which is linear in [0,1] and saturates all the values higher than 1. However, although light intensity can be set higher than 1, direct rendering does not perform the correct computation because clamping occurs before the texture mapping whereas we need it afterwards. We use the same process as previously: we divide the light intensity by 2^n so it is lower than 1 and makes n doubling passes. This assures that no value is clamped before the texture mapping and thus achieves the correct result since the clamping only occurs during the pass combination. This effects is shown in Figure 10-a.

Eye highlights Eye highlights are a very strong cue for the lighting environment: due to their very high specularity, eyes behave almost like mirrors that reflect the surrounding objects. However, because of the iris, the eyes contains a lot of details and, thus, only high intensity features of the environment are actually distinguishable. Therefore we can limit the surrounding environment to only the light sources and still having a convincing effect.

Practically, we place two hemispheres corresponding to both eyes. These spheres are purely specular with a very high shininess exponent. Because of this high exponent, the hemispheres have to be finely subdivided (in practice, we use two hemispheres with 256 vertices each). We perform an additive composition: the light reflected by the cornea is superimposed on the light coming from the deeper part of the eye (iris, vitreous, etc.). This gives a hint of the radius of the spheres, it should be approximately the cornea radius *e.g.* 7.8 millimeters [44].

Figure 12 illustrates the impact of the highlights in the eyes: without it, eyes look dry and frozen whereas with it, they have the expected shiny aspect. The improvement can be better seen in the video.

Cast shadows Our model already includes shading effects because of the underlying Phong model. However, under unidirectional lighting or with a point light source, cast shadows are also a significant lighting cue. We use the *shadow maps* [43] to add cast shadows to the Phong rendering. Compared to other classical methods such as *shadow volumes*, this method has the advantage that it can handle complex geometry like faces. But it involves three passes per light source: one rendering from the light view point, one for the lit part and one for the shadowed part. This can significantly reduce the frame rate if there are many sources. It should be used carefully (*e.g.* in a level of detail context). Figure 11 shows the improvement obtained with the shadows.

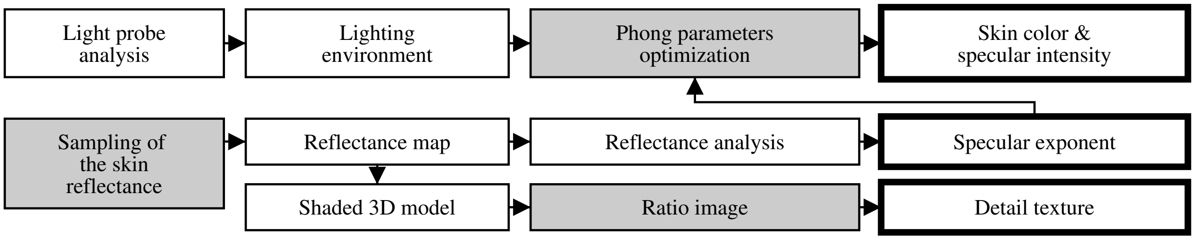


Figure 3. Overview of the analysis process. Gray boxes use the face photograph. Right boxes contain the information needed to render the new images.

6. Results

Our scanner provides a mesh at a resolution of 200×200 with a maximum error of 300 micrometers. It also shoots a 400×400 photograph of known parameters while digitizing. This photograph has a very poor quality and has no flash. We also shoot a high quality photograph with a flash. From this one we extract a 512×512 face picture and the correspondence between this new image and the 3D model is made by matching feature points between this photograph and the scanned one with known parameters. The subject is always between 2 and 3 meters from the camera. Since the flash is about 6 centimeters above the optical center, we can estimate to 2 degrees the maximum angular distance between the view direction and the light direction and to 4 degrees the maximum angular deviation of the view angle on the face. Thus, the assumption that these values are null is reasonable. Our flash is a rectangle of 1cm$\times$2.5cm, and has in our acquisition setup a maximum angular dispersion of 0.75 degrees. Thus it can be accurately approximated by a point light source.

For comparison purposes, Figure 14 shows sample images rendered with the latest available hardware. These images suffer either from their unrealistic texture (Fig. 14-a) or from a too smooth aspect (Fig. 14-a). The method presented in this paper can easily improve both techniques without losing their intrinsic qualities. Note that using these techniques instead of a Phong model limits the result to the latest graphics cards.

To validate our process, we have taken photographs with various light positions and compared them with relighting results (Fig. 16). As the reference photographs have not been taken under the same conditions as the input photograph, the pose and the facial expression are different. There are other differences to discuss. The contrast is lower in the relighted picture. This mainly comes from the 3D model, which does not capture all the geometric details (*e.g.* the lips and wrinkles are not deep enough). This makes some shadows too smooth. On the other hand, some shadow edges are too hard because subsurface scattering is not rendered. As previously discussed, the nose highlight is not strong enough and [26] mentions that parametric models do not catch the right skin properties for grazing angles, this can be seen in Figures 16-a,b on the left cheek. Some tiny details are also missing because of the texture resolution which is lower than the photograph resolution. Nevertheless, the overall appearance matches: shadows boundaries and shading are accurate. As the images are rendered in real time, lighting variations therefore make convincing shading variations including highlights and cast shadows (see the video). Moreover, without any additional feature, the method deals with a complex lighting environment (Fig 10-b) and a difficult case with a short beard through which skin appears (Fig 15-a).

We have measured the frame rate for two models: *scanner* coming directly from the scanner with 12,250 vertices and 24,060 triangles (Fig 13-a) and *simple* which has been simplified to 787 vertices and 1143 triangles (Fig 13-b). We have tested the simple environment *1L* with one light source and a complex environment *8L* with 8 sources and overexposure (two more passes). The graphics cards are from NVIDIA: a TNT 2 (low 3D capacity), a GeForce 4 MX 440 (middle 3D capacity) and a GeForce 4 Ti 4400 (high 3D capacity). The results are summarized in the following table.

	simple + 1L	simple + 8L
TNT2	95 Hz	28 Hz
MX 440	90 Hz	90 Hz
Ti 4400	110 Hz	110 Hz

	scanner + 1L	scanner + 8L
TNT2	3	1
MX 440	78 Hz	58 Hz
Ti 4400	110 Hz	60 Hz

Note that the engine is able to render the *simple* mesh on all three cards at a rate high enough to be able to render other objects and still maintain real time. This point is remarkable since the TNT2 (1999) has limited capabilities compared to the newer cards. Moreover our acquisition process produces data which resists extreme degradation of the supporting mesh. Figure 13 shows that even with 20 times fewer triangles, the face looks almost the same. There is a limited loss of contrast on the coarser mesh because creases have been smoothed but the visual quality is comparable.

The method developed here can be extended to 3D models obtained from Computer Vision techniques. The main advantage of these techniques is that they use less equipment (only a camera) and a strong consistency between the geometry and the input photographs. Interested readers are referred to books like [8, 11]. The latest methods are very promising, including the work of Lhuillier and Quan [20, 21], Zhang et al. [45], and Shan et al. [37]. We tested our method with a head model produced with the method of Lhuillier and Quan [20, 21]. Even though the resulting mesh is less precise than that from a scanner, we achieve satisfying results on this model. A sample image is shown in Figure 15-b.

7. Conclusions

Our results suggest several conclusions. Nowadays hardware rendering uses programmable chips which push the limits of real-time rendering. Obviously, these cards introduce an extended expressiveness associated with an enhanced rendering power, which allows us to simulate very subtle phenomena faster and faster. For instance, subsurface scattering may soon be available in real time with dedicated hardware functions. However, this can sometimes hide the reflection that discerns which features are useful to simulate exactly and which ones can be approximated. As an example, for the non smooth aspect of the skin, what are the useful realism cues? Do we need a sub-millimeter mesh coupled to subsurface scattering rendering to achieve a low intensity, high frequency shading aspect? We have just shown that this can be roughly rendered with a modulation texture. Nevertheless, subsurface scattering is known to have a strong impact on some regions like the shadow boundaries, which are always soft on the skin because of this phenomenon. Thanks to that kind of reflection, we can plan for future work concentrating the hardware power on the spots where it is actually needed whereas the other regions are rendered with a lightweight technique like ours.

We also plan to look carefully at hair. On the one hand, hair is quite different from the skin but on the other, it also has a complex lighting behavior that naturally introduces numerous questions. We believe that there is a way to find a efficient tradeoff for hair rendering between exact rendering and simplistic approximation. Moreover, this would nicely improve our face relighting engine which lacks a specialized model for the hair and would extend our face rendering to the entire head. This is all the more interesting that some Computer Vision algorithms are now able to acquire the whole head geometry.

We always seek to keep in mind the constraints that our methods should be widely usable. In particular, we want to keep the fact that the relighted model we have built is easily reusable because we have controlled the lighting acquisition conditions and compensated for them. The fact that models acquired with different conditions can be mixed is an important asset we want to preserve in our future research.

Acknowledgments The visit of Sylvain Paris at HKUST has been supported by the Eurodoc program from Région Rhône-Alpes. We thank NVIDIA for allowing us to compare our work with their demos, Maxime Lhuillier for providing Computer Vision head models, and Raphaël Grasset and Marc Lapierre for letting us scan and use their faces.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Proc. of image inpainting. In *Proc. of ACM SIGGRAPH*, 2000.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proc. of ACM SIGGRAPH*, 1999.

[3] S. Boivin and A. Gagalowicz. Image-based rendering of diffuse, specular and glossy surfaces from a single image. In *Proc. of ACM SIGGRAPH*, 2001.

[4] K. Daubert, H. Lensch, W. Heidrich, and H.-P. Seidel. Efficient cloth modeling and rendering. In *Proc. of Eurographics Workshop on Rendering*, 2001.

[5] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proc. of ACM SIGGRAPH*, 2000.

[6] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. of ACM SIGGRAPH*, 1997.

[7] P. Debevec, A. Wenger, C. Tchou, A. Gardner, J. Waese, and T. Hawkins. A lighting reproduction approach to live-action compositing. In *Proc. of ACM SIGGRAPH*, 2002.

[8] O. Faugeras, Q. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.

[9] A. Georghiades, P. Belhumeur, and D. Kriegman. Illumination-based image synthesis: Creating novel images of human faces under differing pose and lighting. In *Proc. of IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, 1999.

[10] Z. S. Hakura, J. E. Lengyel, and J. M. Snyder. Parameterized animation compression. In *Proc. of Eurographics Workshop on Rendering*, 2000.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Pres, 2000.

[12] B. K. P. Horn. *Robot Vision*. MIT Press, 1986.

[13] K. Ikeuchi and K. Sato. Determining reflectance properties of an object using range and brightness images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(11):1139 – 1153, November 1991.

[14] H. W. Jensen and J. Buhler. A rapid hierarchical rendering technique for translucent materials. In *Proc. of ACM SIGGRAPH*, 2002.

[15] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proc. of ACM SIGGRAPH*, 2001.

[16] J. Kautz and H.-P. Seidel. Towards interactive bump mapping with anisotropic shift-variant brdfs. In *Proc. of ACM SIGGRAPH/EG Conf. on Graphics Hardware*, 2000.

[17] E. P. F. Lafortune, S. C. Foo, K. E. Torrance, and D. P. Greenberg. Non linear-linear approximation of reflectance functions. In *Proc. of ACM SIGGRAPH*, 1997.

[18] L. Latta and A. Kolb. Homomorphic factorization of BRDF-based lighting computation. In *Proc. of ACM SIGGRAPH*, 2002.

[19] H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-based reconstruction of spatially varying materials. In *Proc. of Eurographics Workshop on Rendering*, 2001.

[20] M. Lhuillier and L. Quan. Quasi-dense reconstruction from image sequence. In *Proc. of European Conf. on Computer Vision*, 2002.
[21] M. Lhuillier and L. Quan. Surface reconstruction by integrating 3d and 2d data of multiple views. In *Proc. of IEEE Int. Conf. on Computer Vision*, 2003.
[22] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *Proc. of ACM SIGGRAPH*, 2001.
[23] C. Loscos, M.-C. Frasson, G. Drettakis, B. Walter, X. Granier, and P. Poulin. Interactive virtual relighting and remodeling of real scenes. In *Proc. of Eurographics Workshop on Rendering*, 1999.
[24] S. R. Marschner and D. P. Greenberg. Inverse lighting for photography. In *Proc. of. Color Imaging Conference*, 1997.
[25] S. R. Marschner, B. Guenter, and S. Raghupathy. Modeling and rendering for realistic facial animation. In *Proc. of Eurographics Workshop on Rendering*, 2000.
[26] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance, and D. P. Greenberg. Image-based brdf measurement including human skin. In *Proc. of Eurographics Workshop on Rendering*, 1999.
[27] D. McAllister. *A Generalized Surface Appearence Representation for Computer Graphics*. PhD thesis, University of North Carolina, 2002.
[28] D. McAllister, A. Lastra, , and W. Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *Proc. of ACM SIGGRAPH/EG Conf. on Graphics Hardware*, 2002.
[29] K. Nishino, Z. Zhang, and K. Ikeuchi. Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. In *Proc. of IEEE Int. Conf. On Computer Vision*, 2001.
[30] S. Premoze. Analytic light transport approximations for volumetric materials. In *Proc. of Pacific Graphics*, 2002.
[31] R. Ramamoorthi and P. Hanrahan. Frequency space environment map rendering. In *Proc. of ACM SIGGRAPH*, 2002.
[32] H. Rushmeier and F. Bernardini. Computing consistent normals and colors from photometric data. In *Proc. of IEEE Int. Conf. on 3-D Digital Imaging and Modeling*, 1999.
[33] H. Rushmeier, F. Bernardini, J. Mittleman, and G. Taubin. Acquiring input for rendering at appropriate levels of detail: Digitizing a Pietià. In *Proc. of Eurographics Workshop on Rendering*, 1998.
[34] H. Rushmeier, B. Rogowitz, and C. Piatko. Perceptual issues in substituing texture for geometry. In *Proc. of SPIE Conf. on Human Vision and Electronic Imaging*, 2000.
[35] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *Proc. of ACM SIGGRAPH*, 1997.
[36] S. Shafer. Using color to separate reflection components. *Color Resolution Applications*, 10(4):210–218, 1985.
[37] Y. Shan, Z. Liu, and Z. Zhang. Model-Based bundle adjustment with application to face modeling. In *Proc. of IEEE Int. Conf. On Computer Vision*, pages 644–651, 2001.
[38] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proc. of ACM SIGGRAPH*, 2002.
[39] M. Stamminger, J. Haber, H. Schirmacher, and H.-P. Seidel. Walkthroughs with corrective texturing. In *Proc. of Eurographics Workshop on Rendering*, 2000.
[40] M. Tarini, H. Yamauchi, J. Haber, , and H.-. Seidel. Texturing faces. In *Proc. of Graphics Interface*, 2002.
[41] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 1967.
[42] Z. Wen, Z. Liu, and T. Huang. Face relighting with radiance environment maps. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.
[43] L. Williams. Casting curved shadows on curved surfaces. In *Proc. of ACM SIGGRAPH*, 1978.
[44] G. Wyszecki and W. S. Stiles. *Color science: Concepts and methods, quantitative data and formulae*. John Wiley and Sons, 1982.
[45] Z. Zhang, Z. Liu, D. Adler, M. Cohen, E. Hanson, , and Y. Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. Technical Report MSR-TR-2001-101, Microsoft Research, 2001.

(a) (b) (c)

Figure 4. Skin segmentation (a) input image (b) region painted by the user (c) skin region

(a) (b) (c)

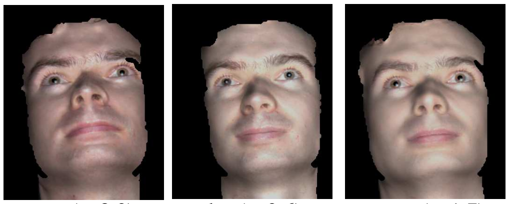
Figure 5. Reflectance map (a) color samples (b) clusters (c) final map

(a) (b) (c)

Figure 6. Detail texture creation: (a) input photograph (b) shaded face (c) detail texture

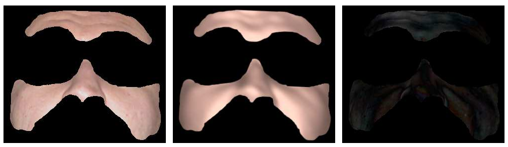
(a) (b)

Figure 7. (a) A simple light probe (b) The classification of the pixel between [red] captor saturated [blue] ambient lighting [yellow] halo



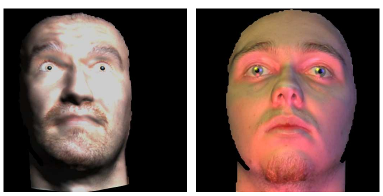
wet (s=8.0) dry (s=2.6) matte (s=1.7)

Figure 8. Comparison of the shininess exponent under various skin conditions (mate skin has been obtained with foundation cream)



(a) (b) (c)

Figure 9. Comparison between the input photograph (a) and result of the optimization (b); (c): difference between (a) and (b).



(a) (b)

Figure 10. (a) Stronger contrast thanks to overexposure (exaggerated for illustration purpose). The details disappear in the highlight. (b) Relighting with 3 light sources.

(a) (b)

Figure 11. (a) Without shadow mapping (b) With shadow mapping

(a)

(b)

Figure 12. (a) Without highlight in the eyes (b) With highlight in the eyes

(a) (b)

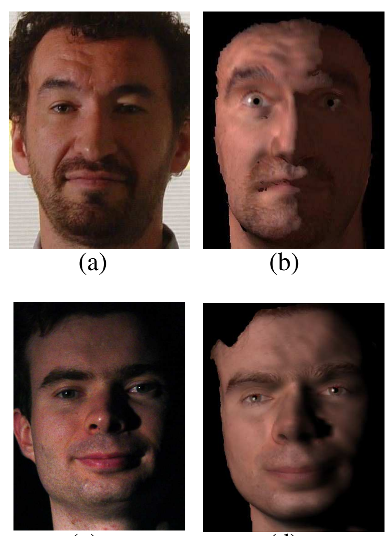
Figure 13. Comparison between (a) fine mesh (*scanner*) and (b) coarse mesh (*simple*)

(a) (b)

Figure 14. Sample face pictures rendered with the latest graphics cards (pictures courtesy of NVIDIA). (a) With a NVIDIA GeForce 3 card (b) With a NVIDIA GeForce FX card.

(a) (b)

Figure 15. (a) A short beard with visible skin. (b) Using a Computer Vision 3D model



(c) (d)

Figure 16. Comparison between real photographs (a,c) and relighted images (b,d).