EUROGRAPHICS 2004 / M.-P. Cani and M. Slater
(Guest Editors)

Volume 23 (2004), Number 3

A Physically-Based Model for Rendering Realistic Scratches

Carles Bosch,[1] Xavier Pueyo,[1†] Stéphane Mérillou[2] and Djamchid Ghazanfarpour[2‡]

[1] Institut d'Informàtica i Aplicacions, University of Girona, Spain
[2] MSI Laboratory, University of Limoges, France

Abstract

Individually visible scratches, also called isolated scratches, are very common in real world surfaces. Although their microgeometry is not visible, they are individually perceptible by the human eye, lying into a representation scale between BRDF and texture. In order to simulate this kind of scratches in synthetic images we need to know their position over the surface (texture scale), so we can determine where to use the specific scratch BRDF instead of the ordinary surface BRDF. Computing the BRDF of a scratch is difficult because it depends on the scratch's invisible microgeometry. In this paper, we propose a new physically based model to derive this microgeometry by simulating the formation process of scratches. We allow specifying intuitively the parameters involved in the process such as the scratching tool, the penetration forces, and the material properties of the object. From these parameters, we derive the microgeometries of the scratches by taking into account the real behaviour of the process. This behaviour has been determined by analysing existing models in the field of materials engineering and some "scratch tests" that we performed on metals. Our method has the advantages of easily simulating scratches with a wide range of microgeometries and taking into account the variability of their microgeometry along the scratch path. Another contribution is related to the location of the scratches over the surface. Instead of using an image of the paths as in previous work, we present a new representation based on curves defining the paths. This offers an independence on the image resolution or the distance from the observer and accurately provides the scratch direction in order to compute scratch BRDFs.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Color, shading, shadowing, and texture.

1. Introduction

The rendering of defects is an important key to achieve realism in computer-generated images. In most real situations, objects exhibit many defects, such as dust, corrosion, fracture, scratches or peeling. To simulate this defects there are already some works done in Computer Graphics [Bli82][TF88][BB90][DH96][WNH97], but there is still much room for improvement. With regard to scratches, we find two different types of scratches in real surfaces [MDG01]: microscratches and individually visible scratches. Microscratches provide an homogeneous aspect to the surface and are generally not individually perceptible due to their small size. Usually they are simulated with anisotropic BRDF models [Kaj85][PF90][War92]. On the other hand, individually visible scratches, also called isolated scratches, provide an individually perceptible behaviour, even if their geometry remains invisible (Figure 1). These scratches lie into a representation scale between texture and BRDF, so they can be simulated by using a texture to define their location and path along the surface (Figure 2), and a BRDF to model the specific light reflection on each scratch point [BL99]. This type of scratches has been little focused in Computer Graphics. Up to now, only our previous model [MDG01] deals with the realistic simulation of these scratches, where the real microgeometry of the scratches is taken into account in order to compute the scratch BRDFs. This microgeometry is specified for each scratch by means of its cross-section (Figure 2). However, this cross-section is hard to determine without previous knowledge of the scratch's (invisible) microgeometry. Furthermore, the model

† {carles.bosch|xavier.pueyo}@ima.udg.es
‡ {merillou|ghazanfarpour}@unilim.fr

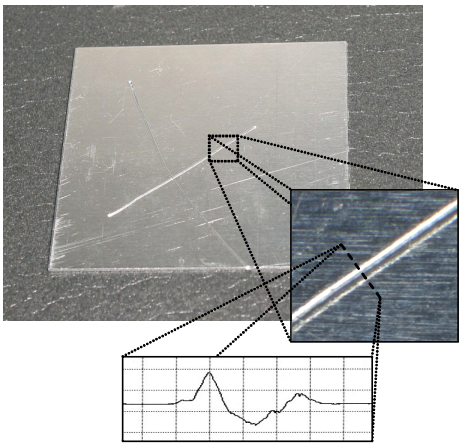

Figure 1: *Real scratched plate of aluminium with a close view on the microgeometry of a scratch and its measured cross-section.*

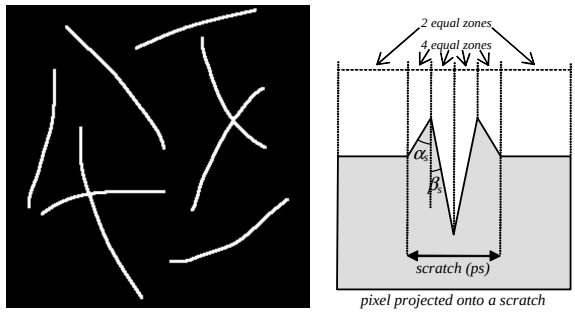

Figure 2: *Scratches are defined by their paths (texture) and their cross-section geometry (used to compute the BRDF).*

only accounts for some specific cross-section shapes, and the geometry of the scratches is considered constant along their paths. All these constraints are very significant because light reflection on a real world scratch may change drastically with the geometry.

In the field of materials engineering, many works focus on the scratching processes due to their importance when measuring the scratch resistance of materials, especially for polymers and thin coatings [BEPS96]. These state that the microgeometry of a scratch depends on the parameters involved in its formation process, like the material properties of the object, the scratching tool or the applied force. Some also quantify the contribution of these parameters to the final geometry [JZLM98][Buc01].

In this paper, we introduce a new physically-based model for simulating scratches by taking into account the scratch formation process. The objective is to derive the complex microgeometry of the scratches from the parameters of the process, such as the tool and its orientation, the force, and the material properties. This allows the simulation of scratches with a wide range of geometries by simply specifying the parameters involved in their generation. Furthermore, the proposed model is physically-based, because it considers the real behaviour of the scratching process. This is achieved by analysing some existing models in the field of materials engineering and by performing several "scratch tests" and measurements. Here our work mainly focuses on the scratching processes over metals and alloys because their behaviour is more common than for other materials, like ceramics (glass, porcelain, ...) or polymers (plastic, rubber, ...) [Cal94], but it could be extended to incorporate those types of materials as well. Another improvement we propose here is the parameterisation of the scratch process along the paths of the scratches, so we can take into account the variability of geometry along their paths. Finally, a last improvement is related to the definition of the paths over the surface. Instead of using an image like in previous models (Figure 2), we introduce a new representation based on curves defining the paths. This representation offers the advantage of being independent on the image resolution or the observer's distance, and accurately provides the scratch direction and other parameters in order to compute scratch BRDFs.

This proposed model has many important applications. It can be used to study the real appearance of manufactured products and materials when scratched under certain conditions, or to study their scratch resistance. It can be also employed to train computer vision systems for accurately detecting scratched objects. Furthermore, in the artistic engraving of metals such as gold or silver, it can serve to test different designs or tools before the final engraving operation, avoiding possible mistakes and reducing important costs.

The rest of the paper is organised as follows. In section 2 we discuss the previous work. Next, section 3 explains the performed scratch tests and how we derive the geometry of the scratches from the parameters of the scratch process. In section 4 we compute the BRDF of a scratch from its specific geometry. Then, section 5 presents the new representation of scratch paths using curves. Results are finally given in section 6. We conclude and give future research directions in section 7.

2. Previous work

All the literature concerning the simulation of individually visible scratches is based on the same principle. The location of the scratches over an object's surface is defined by a scratch pattern, represented by a 2D image with the scratch paths painted on it. The pattern is applied onto the surface using 2D texture mapping techniques and it serves to indicate if a point (projected pixel) on the surface contains or not a scratch. If it contains a scratch then its light reflection is computed using a specific BRDF instead of the common surface BRDF.

The very first authors that considered the rendering of isolated scratches were Becket and Badler [BB90]. They developed a system to simulate many types of surface defects using 2D texture generation techniques. Scratches were defined as straight lines and placed on the pattern with random lengths and directions. Their reflection behaviour was then simulated simply by assigning a random intensity to each one, without taking into account their anisotropic behaviour (i.e. the reflection did not depend on the position of light sources and viewer).

Buchanan and Lalonde [BL99] later proposed a model taking into account the anisotropic behaviour. Scratches were randomly placed on the pattern also as straight lines, but saving on each texel a list of the scratches passing through it. In the rendering pass, the BRDF of each surface point was then computed by adding to the surface base reflection the maximum highlight of all the scratches on it. This model has the disadvantage that is a Phong like model and does not consider the microgeometry involved in a scratch. In addition, all scratches behave in the same way (i.e. same geometry and reflection properties).

Finally, our previous model [MDG01] proposed to consider the scratch microgeometry to compute the BRDF of the scratches. By analysing different cross-section measurements of real scratches, we determined a common geometry for scratches: a groove surrounded by two peaks. This geometry was simplified by considering two equal peaks and the same widths for all the zones of a cross-section, so the geometry could be easily defined by means of two angles (Figure 2). On the other hand, scratches were located (painted) in the pattern as desired, and the pattern was used to determine the scratch direction for each point of a scratch. This direction was important to adequately locate the cross-section geometry and then compute the BRDF, accounting for the shadowing and masking effects among the different zones of the cross-section of a scratch. This model respects the anisotropic behaviour of the scratches and has many other additional advantages: it can be used with any model of reflection (Phong, Cook & Torrance, ... [FvDFH90]), a different BRDF can be specified for each zone of a cross-section (i.e. to simulate scratched multilayered surfaces), and the resulting BRDF is physically correct if the specified BRDFs are physically correct too. However, this model has many drawbacks as stated before. First, the microgeometry of the scratches is very limited, not only for the cross-sections (Figure 2) but also along the paths, because the geometry remains constant. On the other hand, it is difficult to specify a cross-section with two angles if the real profile of the scratch cannot be measured. Concerning the pattern, the use of an image makes imprecise the determination of the scratch direction, given that for each point of a scratch the direction is determined by analysing the neighbouring texels. This is especially true if the pattern contains close scratches, intersections or image noise.

3. Scratch geometry

To determine the geometry of a scratch from its generation process we have considered the next parameters (Figure 3): the material properties of the surface, the geometry of the tool used to scratch, the orientation (θ_t, ϕ_t) of the tool relative to the surface normal, and the force F_N applied with the tool. As stated before, here we focus our work on metals because its behaviour under a scratching process is simpler than for other materials.

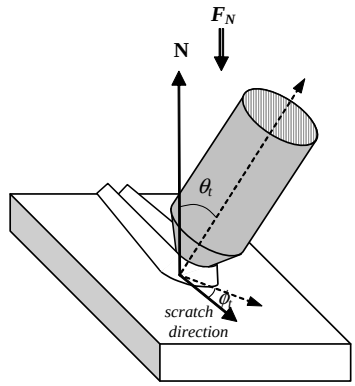

Figure 3: *Scheme of the scratching process.*

3.1. Measuring real-world scratches

We performed many tests to understand the behaviour of scratches on metals. Those tests can be separated in two classes: using a scratch tester or made by hand. A scratch tester is an instrument that offers the possibility to perform controlled and accurate scratch tests over a material sample. Usually it uses some specific tool (in this case a Rockwell diamond cone, see Figure 4) and allows the precise specification of some of the test parameters, like load (force). However, some other parameters are fixed, like the scratching direction or the orientation of the tool. In our case, the scratch tester was used to study the effect of different forces onto different metals. We used four samples: aluminium, brass, steel, and titanium. Then for each one we made different scratches applying different loads from 0.5 to 4 Kg.

Manual tests were made to study the effect of different tool orientations, because the scratch tester does not allow this. In this case, we used a sample of aluminium alloy, and a nail as the tool. Different scratches were made onto the alloy changing the orientation of the nail (i.e. changing θ_t and ϕ_t).

After performing the tests, we measured each scratch with a Hommelwerke T2000 profilometer, which allows the measuring of scratch cross-sections (Figure 1 and Figure 4). For each profile we measured the depth of the groove, *p*, the height of the peaks, *h*, and their angles, α and β (Figure 4).

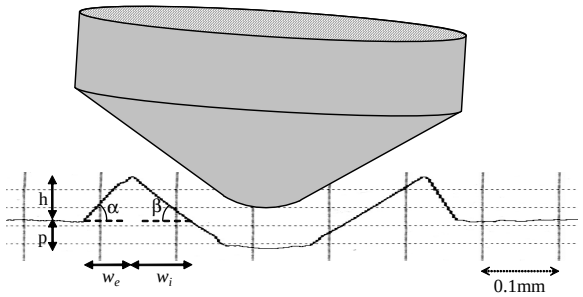

Figure 4: *Tool of the scratch tester and real scratch cross-section measured with the profilometer. It shows the dependence between the groove of the scratch and the geometry of the tool, and also the measured values for each scratch.*

For each material sample, we also measured hardness, using a static hardness tester. This hardness value will be useful to relate the behaviour of the scratches to the properties of the material, as will be shown in section 3.2. Hardness was measured on the Vickers scale, which is one of the common hardness scales [Cal94].

3.2. Deriving the geometry of a scratch

As stated in [MDG01] the cross-section geometry of a scratch is composed of a groove and two peaks (Figure 2). The groove is due to the penetration of the tool into the material, and the peaks are due to the flow and pile-up of the material around the tool during the scratch process [JZLM98]. Consequently, the shape of the groove and the internal zones of the peaks depend on the geometry of the tool (Figure 4). In the case of metals moreover, this dependence can be assumed as direct, because metals have no significant shape recovery after a scratch [BEPS96].

Apart from the shape, we need to know the depth of the groove from the surface base-line, p. The depth is directly related to the force applied with the tool and the material properties of the object, especially with hardness. For most metals the wear volume is inversely proportional to hardness [JZLM98]. From the measurements obtained by the scratch tester, we have derived a function that relates the depth of penetration, p (in mm), with the applied force, F_N (in kg), and the Vickers hardness of the material, H_V (in kg/mm^2):

$$p = 0.182\sqrt{\frac{F_N}{H_V} + 0.0055} - 0.014\,.$$

Although this relation may vary for different geometries of tools [Buc01], here we assume that force and hardness will be usually specified as approximate values (see section 3.3), so a loss of accuracy can be neglected for depth. With regard to the geometry of the peaks, from our measurements and [Buc01], we have found a linear relation between the internal angles of the peaks, α, and the external ones, β. For each peak, this relation is the following:

$$\tan\beta = -0.56 + 2.54\tan\alpha\,.$$

Here we assume that there is no loss of material during the scratch process. For metals, this is accomplished if the scratching tool is not too sharp [Buc01]. This allows us to consider that the area of the groove of a scratch, A, is equivalent to the sum of the areas of the peaks. Thus, the height of each peak, h, can be easily obtained using the next expression:

$$h = \sqrt{\frac{A}{\cot\alpha + \cot\beta}}\,.$$

Finally, the width of each zone of the peaks, w_i and w_e (Figure 4), is obtained from its height and angle:

$$w_i = h\cot\alpha\,,$$
$$w_e = h\cot\beta\,.$$

Summarising, the geometry of a scratch is derived from the process parameters using the following relations:

1. The geometry of the tool relates to the shape of the groove and the angles of the peaks.
2. The hardness of the object and the applied force relates to the depth of the groove (and the height of peaks).

From our measurements, we have seen that the orientation of the tool from the object's surface is also related to the profile; although we have not found any work that study this parameter during a scratch process. The tool is usually considered as being oriented parallel to the surface normal. From the measurements obtained with the manual tests, although less precise than the ones made with the tester, we have concluded that the orientation only supposes a rotation on the geometry of the tool. Consequently, the resulting scratch cross-section depends on the rotated geometry as if it was a different tool, so the last exposed assumptions and equations are also valid for this rotated geometry.

3.3. Parameters specification

Different considerations must be taken into account when specifying each of the parameters of the scratch process. Concerning the hardness property for example, it will be defined for the entire object or surface usually, and as an exact or an approximate value depending on the user's knowledge. Nevertheless, it can be defined to change its value over the surface by using a texture if desired. Concerning the tool, we suppose that it is previously modelled as a 3D model and in real world coordinates (μm, mm, ...), so the geometry of the scratch's groove can be obtained directly from its geometry. The scale used to model the tool will depend on the size of the scratches we need to simulate, and if desired, a different tool can be used for each scratch. Finally, the orientation of

the tool and the force can be specified for each scratch along its entire path. If these are specified as non constant along a path then the geometry of the scratch (and its reflection behaviour) will change along the path. In the case of the orientation of the tool, it is defined by its two angles, θ_t and ϕ_t, as mentioned, and the penetration force as a single value. In addition, the force can be specified visually, because it is closely related to the size of the resulting scratch, as well as hardness. So, these may be specified interactively.

Our purpose is to ease the task of specifying the parameters of the scratches by means of using approximate values, because the knowledge of the exact values in a scratch process is very difficult and not necessary for many applications. Nevertheless, the proposed model works just as well with exact values as with approximate values.

3.4. Particular cases

Two particular cases especially affect the geometry of a scratch: intersections and hard ends. Intersections between two (or more) scratches exhibit a particular geometry in the intersection point. As observed by measurements that we performed onto many scratch intersections, the geometry in these points exhibit an X form (Figure 5). In the case of scratch ends, when a scratch finishes (begins) softly the material pile-up in front of (behind) the tool is not very high. However, when a scratch finishes (begins) abruptly, with a considerable depth, the pile-up material which appears in front of (behind) the tool remains also considerable.

Like in [MDG01], we consider that the 3D geometry of a scratch can be represented by its 2D cross-section, so the light scattering into the scratch can be easily computed by the BRDF (see section 4). This can be assumed if within a pixel the geometry does not change (see the close view of Figure 1). However, to simulate intersections or hard ends correctly, we cannot use a 2D cross-section; it must be done by using the 3D geometry. Anyway, because these particular cases occur in a few points on some scratches and they are rarely visually perceptible, we will not consider them.

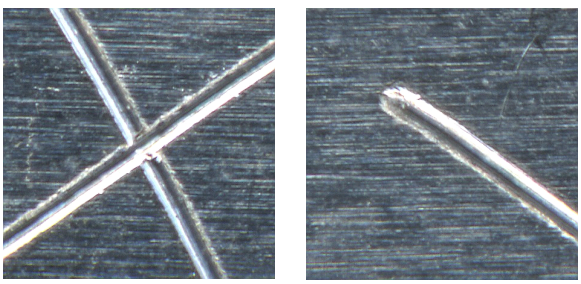

Figure 5: *Close view of the intersection of two real scratches (left); the end of a real scratch (right).*

4. Light scattering model

Once we know how to derive the geometry on each point of a scratch, we need to know how to simulate the light scattering into this virtual geometry. Like in our previous model, we assume that the scratch is always centred on the current pixel (as in the close view of Figure 1), so the mathematics of our light scattering model can be simplified without any important loss of accuracy. In addition, the relative width of the scratch over the pixel size, named scratch proportion or *ps* (Figure 2), is at most half of the pixel ($ps \in [0, 0.5]$). This serves to accomplish that the geometry of the scratch is never visible but only its reflection behaviour [MDG01]. The *ps* value will depend on the real width of the scratch (usually in μm), which is derived from the scratch process, and the surface size contained by the pixel once projected onto the surface (usually in mm). The contained surface will naturally increase with the distance to the viewer, the viewing angle, and the resolution.

4.1. Scratch BRDF

When a pixel projected onto the surface contains a scratch, the light reflected at this area will be described by the scratch BRDF, $f_{r,scratch}$. In order to compute this BRDF, in our previous model we divided the pixel in six zones (Figure 2): four central zones for the scratch and two external zones for the surrounding surface. Here however, we do not know *a priori* which geometry will be derived for a given scratch (Figure 6), due to the number and variability of the parameters involved. Thus, each scratch can have a different number of zones m, and consequently, the pixel is divided in $n = m + 2$ zones. The scratch BRDF is computed as the sum of the light reflected from each zone k [MDG01]:

$$f_{r,scratch} = \sum_{k=1}^{n} f_{r,k} R_k G_k \,,$$

where $f_{r,k}$ is the base BRDF associated to each zone, R_k is the contribution of each zone to the total reflection, and G_k the geometrical attenuation factor, described in section 4.2.

For each zone, the reflection contribution, $R_k \in [0,1]$, will be:

$$R_k = \frac{l_k \cos\theta_{i,k} \cos\theta_{r,k}}{\cos\alpha_k \cos\theta_i \cos\theta_r},$$

where l_k is the relative area of zone k over the total area of the projected pixel, α_k is the angle of this zone from the surface base line, $\theta_{i,k}$ and $\theta_{r,k}$ are the directions of the incident light and the observer relative to the k zone, and θ_i and θ_r are the directions of the incident light and observer relative to the surface normal (Figure 6). Note that $\cos\theta_{i,k} = N_k \times \omega_i$ and $\cos\theta_{r,k} = N_k \times \omega_r$, where N_k is the normal of the zone k, ω_i the incident vector, and ω_r the viewing vector.

As in [MDG01], a different base BRDF, $f_{r,k}$, can be associated to each cross-section zone of a scratch. This is espe-

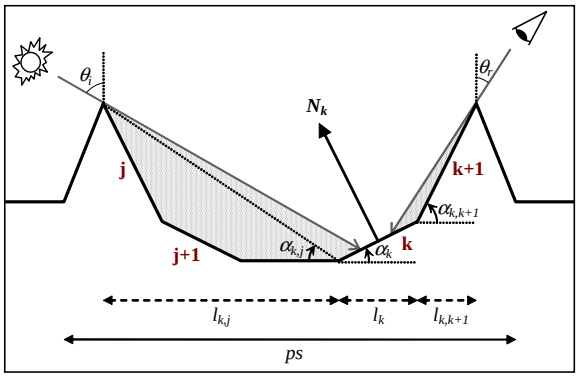

Figure 6: *Cross-section of a pixel containing a scratch and the parameters of each zone k needed to compute the BRDF.*

cially useful when the reflection properties inside a scratch differs from the outside, which is usually the case, because the roughness inside the scratch differs from the surface roughness due to the friction during the scratch process [JZLM98]. This serves also to simulate scratches on multilayered surfaces, like painted or dusted surfaces. In addition, any model of reflection can be used for each $f_{r,k}$, such as Cook & Torrance or Phong [FvDFH90].

4.2. The shadowing/masking term

The geometrical attenuation factor, $G_k \in [0,1]$, represents the shadowing/masking term of each zone of a pixel. This serves to specify how much each zone is visible to the viewer and lit, i.e. not occluded or shadowed by the other zones. To compute this term we have generalised the method from [MDG01] and [ON94], where the term is first derived in 2D from the cross-section and then transformed to 3D. In our case, the main difference is that the groove normally will not be a perfect V shape (Figure 6), so we cannot use the same expressions. Given that we must consider a generic cross-section for the scratch, we have to solve the terms of each zone in a generic way. For each zone k we need to compute its self-shadowing/masking, GS_k, the shadowing/masking coming from the left, GL_k, and coming from the right, GR_k:

$$G_k = \max\left(0, GS_k\left(GR_k + GL_k - 1\right)\right).$$

Each of these terms will be computed by the following expressions:

$$GS_k = \max\left(0, \min\left(1, g_k\left(\omega_i\right), g_k\left(\omega_r\right)\right)\right),$$

$$GR_k = \max\left(0, \min\left(\begin{array}{c} 1, g_{k,k-1}\left(\omega_i\right), g_{k,k-1}\left(\omega_r\right), \cdots \\ \cdots, g_{k,0}\left(\omega_i\right), g_{k,0}\left(\omega_r\right) \end{array}\right)\right),$$

$$GL_k = \max\left(0, \min\left(\begin{array}{c} 1, g_{k,k+1}\left(\omega_i\right), g_{k,k+1}\left(\omega_r\right), \cdots \\ \cdots, g_{k,n}\left(\omega_i\right), g_{k,n}\left(\omega_r\right) \end{array}\right)\right),$$

where $g_k(\omega)$ is the self-shadowing or self-masking (depending on ω) of zone k, and $g_{k,j}(\omega)$ is the shadowing or masking of zone k due to j. Note that the two self-shadowing/masking terms g_k must be computed only once for the current zone, while the terms $g_{k,j}$ must be computed for all the zones to the right of the current zone (to the left in the case of GL_k). These terms are computed by the following expressions:

$$g_k(\omega) = \frac{1}{N_k \times \omega},$$

$$g_{k,j}\left(\omega\left(\theta,\phi\right)\right) = 1 + \frac{l_{k,j}}{l_k}\cos\alpha_k \frac{\cos\theta - \tan\alpha_{k,j}\sin\theta\cos\left(\phi - \phi_{scratch}\right)}{N_k \times \omega},$$

where $l_{k,j}$ is the distance between k and j, $\alpha_{k,j}$ is the angle between k and j, and $\phi_{scratch}$ is the azimuthal orientation of the scratch cross-section into the surface, obtained from the current scratch direction (see section 5). Note that $l_{k,j}$ and $\alpha_{k,j}$ are computed in a different way when the zone j is on the left or on the right from the zone k (Figure 6).

In order to avoid the computation of $g_{k,j}$ for each zone to all other zones of the cross-section, we first precompute which zones affect which ones. This can be easily done by considering for the current zone only the higher ones (j, $j+1$ and $k+1$ for the zone k in Figure 6), and neglecting the zones that are always occluded by others ($j+1$ is always occluded by j when k is occluded in Figure 6). In addition, if the geometry of a scratch does not change along its path most of the cross-section parameters are also precomputed.

5. Pattern based on curves

The main function of the scratch pattern is to specify the paths of the scratches over an object's surface. In [MDG01], this pattern was represented by a colour image, indicating the non-scratched surface in black and scratches with other colours (Figure 2). This image was then mapped onto the surface to locate the scratches appropriately.

Here we propose to define the pattern in a different way, by using curves. The concept is similar, but instead of 'painting' the scratches on an image we define each scratch path directly with a curve, saving its control points instead of the resulting image. During the rendering process, the curves are used to determine if the current pixel contains or not a scratch. This is easily done in texture space, by performing a 2D intersection test between the projected pixel and the curves. We have implemented a procedural texture [FvDFH90] that reads the data of the curves, represents them into the texture space, and performs the intersection tests. In order to represent the curves into the texture space, curves are projected onto the object during the modelling process (Figure 7). This allows the definition of the paths directly onto the surface of the object. Concerning the intersection tests, in order to avoid doing the tests for all

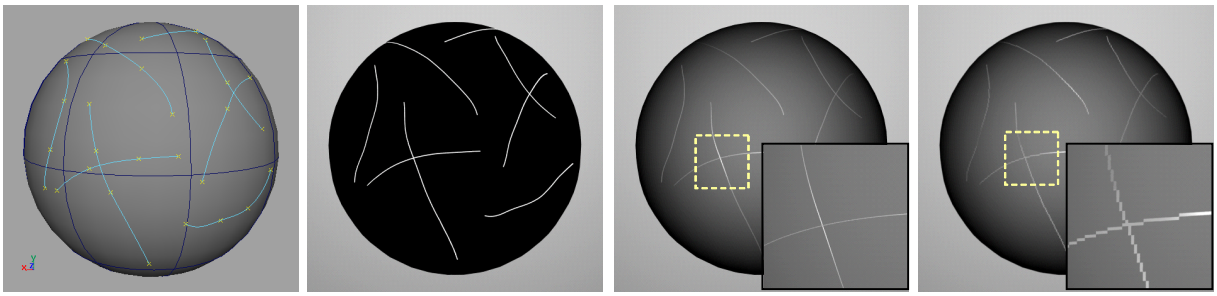

Figure 7: *Pattern based on curves. Left: modelling the curves directly onto the surface. Middle left: a procedural texture represents the curves into the texture space. Middle right: rendering (new method). Right: rendering (old method).*

the scratches of a pattern, we previously subdivide the pattern into a uniform grid, saving in each cell a list of all the scratches crossing it. This allows performing the tests only with the nearest scratches, like is done with the objects of a scene in accelerating raytracing [Gla89].

This method, although slower than using an image, allows a fine and continuous representation of the scratches at any image resolution or distance from the viewer. Figure 7 shows a comparison between scratches rendered using curves (middle right) and rendered using an image (right). When the camera is approached to the surface (see close view of each image) the differences are obvious. On the other hand, curves allow to easily computing some parameters of the scratches, like the scratch direction and sense on each point of a scratch, which can be obtained by means of the curve tangent. In [MDG01] only the direction was determined, not its sense, because the cross-section geometry was considered as symmetric. However, in this new model, asymmetric cross-section geometries are very common, and the knowledge of the direction sense is important. Another advantage of the new representation is that we can precisely determine if the current pixel is in an intersection or an end of a scratch. Finally, we can also determine for each point of a scratch its distance or position along its path. This property is especially important to associate different scratch process parameters along the path of a scratch, and correctly use them during the derivation of its geometry (see section 3.3).

In general, this new representation could be useful to render any type of surface annotation (geometric lines) [SKHL00].

6. Results

We implemented our method as two plug-ins for Maya®, the reflection model as a shader, and the scratch pattern as a 2D procedural texture. The pattern curves and the tools were also modelled with Maya (Figure 8).

First, Figure 9 shows different synthetic scratches obtained by changing the parameters of the formation process. Each scratch is accompanied below by the derived cross-section geometry. Tested tools (upper left) are nail, screwdriver, scratch tester's cone, and pyramid. Hardness (upper right) and forces (bottom left) increase from left to right. Finally, orientations (bottom right) from left to right are (0,0), (40,0), (60,90), and (45,45) for (θ_t, ϕ_t). To generate these images, light was located facing the camera in the opposite side of the scratched plate. As shown, the geometry and reflection of the scratches depend a lot on the specified parameters. In the case of scratches with high peaks, the shadowing/masking effect plays an important role, thus becoming darker. These results also prove that it is important to take into account the specific geometry of scratches, instead of simplifying it as done in [MDG01]. With our previous model, only the scratches obtained with the pyramid (without orientation) could be well represented.

In Figure 10 we compare some pictures of a real scratched surface with the images obtained using our model. The object is the titanium plate used with the scratch tester and the tool is the tester's conical tip, without any specific orientation. In this case we performed five parallel scratches with different forces (decreasing from left to right on the first im-

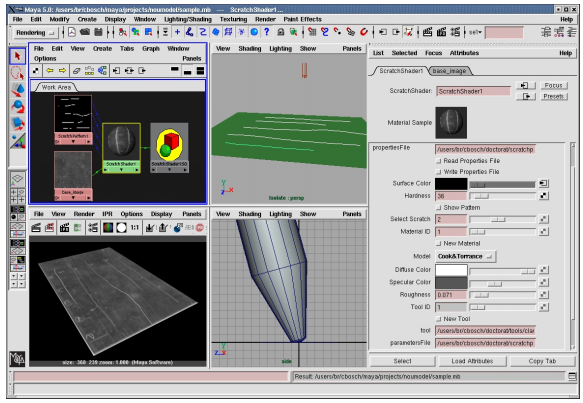

Figure 8: *Simulating scratches with Maya's plug-ins.*

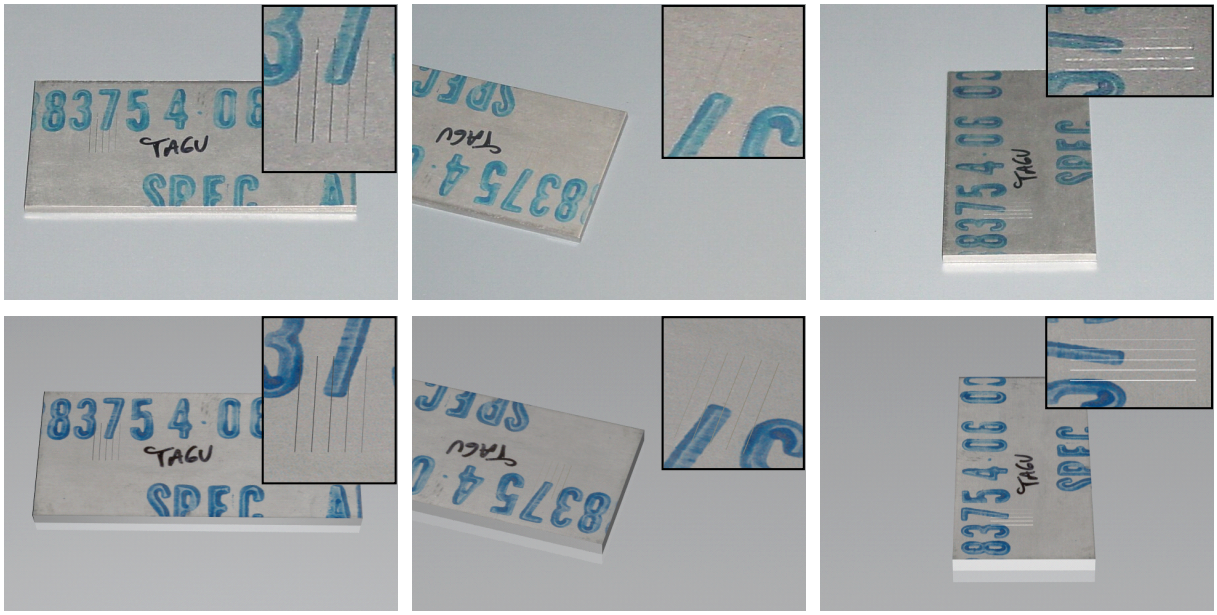

Figure 10: *Real titanium plate scratched using the scratch tester and seen from different viewing angles (top); the corresponding images synthesised with our method (bottom).*

age, from right to left on the second, and from bottom to upper on the third). Titanium hardness was taken into account. Light was positioned beside the camera because pictures were made with flash. As can be seen, our model allows an accurate simulation of the real behaviour of the scratches. When rotating the camera (and the light) around the plate, the simulation of the scratches' reflection closely matches the real reflection without changing neither the process parameters nor the reflection properties.

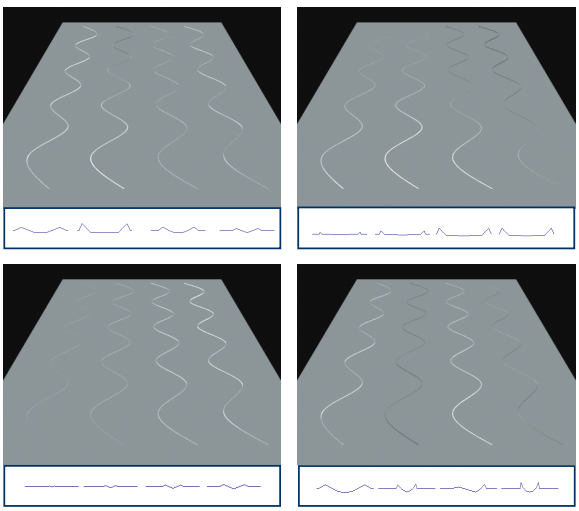

Figure 9: *Simulating scratches with different parameters for: tools (upper left), hardness (upper right), forces (bottom left), and tool orientations (bottom right).*

Figure 11 shows another comparison between a real scratched surface and a synthesised one using our model. In this case, the object is an aluminium plate manually scratched with a nail, without specific orientation. This presents five scratches, which are numbered onto the real plate (top). For the first and fourth scratches, the force is nearly constant and not high. For the second and fifth scratches, the force is stronger and diminishes at the end of their paths. Finally, the third presents different forces along the path. The generated synthetic image demonstrates how our model simulates these scratches by taking into account the variability of the force parameter along the paths. In this case, force was specified manually by visually inspecting the real scratched plate.

Next, Figure 12 and Figure 13 show application examples of our model. Figure 12 shows a real scratched metallic component from a car and the corresponding synthetic image. Images generated with our model can be used to train computer vision systems for correctly detecting scratched parts in manufacturing and inspection processes. For engraving processes, our model can be also used to test different designs, tools, or other parameters over metals before engraving them. This can avoid possible mistakes and reduce important costs. In Figure 13, we show an example of a synthesised gold ring with an inscription. Generally, our model can serve to study the appearance of metallic objects scratched

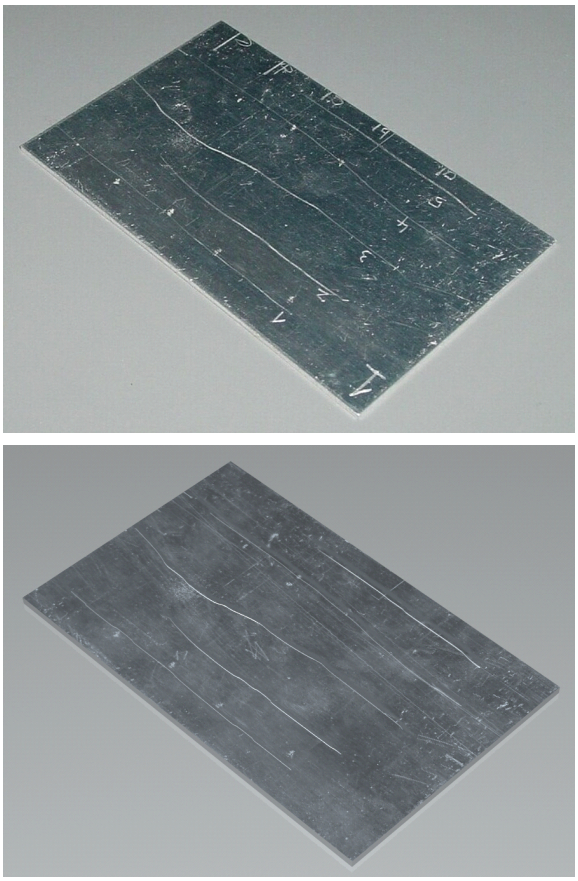

Figure 11: *Real aluminium plate scratched with a nail and applying different forces along the paths of scratches (top); the corresponding synthetic image (bottom).*

under certain conditions and parameters, and to study their scratch resistance.

Finally, in Table 1 we compare the performance between the new model and our previous one [MDG01]. For this, we have tested three scenes, each containing an object with a different number of scratches on it: 5, 50, and 500. Each scene has been rendered with the same quality parameters and on a Pentium 4 (1.8 GHz) with 1Gb RAM. The same scene without scratches rendered in 4 sec. As can be seen, the rendering times for the two models slowly increase with

Scratches:	5	50	500
New model	10 sec	19 sec	67 sec
[MDG01]	8 (10) sec	15 (17) sec	41 (55) sec

Table 1: *Rendering times for different scenes.*

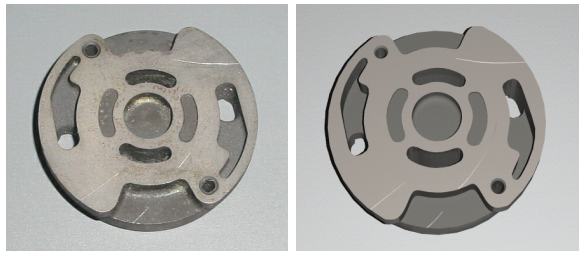

Figure 12: *A real scratched metallic part (left) and its corresponding synthetic image (right).*

Figure 13: *Synthetic ring with an inscription.*

the number of scratches, because more computations are needed. The performance differences between the two models are mainly due to the computation of the intersections between the curves and the current projected pixel for the new model. However, the previous model needs more computation time to obtain a similar quality than using curves. These are shown in brackets. In addition, when the number of scratches increases, the previous model has many problems to compute the scratch directions from the pattern image.

7. Conclusions and future work

In this paper, we have presented a new physically-based model to render realistic scratches. It improves our previous model [MDG01] by allowing an accurate modelling of the scratches' geometries, which is done from the definition of the parameters involved in the scratching process. These parameters offer interesting features: hardness is a real physical property, thus can be found in any material science book, the geometry of the tool can be easily designed, and the force and the orientation of tool are easy to test and intuitive enough. This is satisfying for both the design and the simulation. Furthermore, our method allows the modelling of complex scratches, like scratches with very specific cross-section geometries or scratches where their geometry changes along its path. Another important improvement of this new method is the accurate definition of the scratch paths over a surface

by using curves instead of an image. This improves the visualisation of the scratches due to its independence on the image resolution or viewer's distance, while allowing an easy and accurate computation of many scratch parameters for the BRDF, like the scratch direction.

There are still some future extensions that should be treated in the simulation of scratches. In this paper, we have considered the behaviour of scratches on metals; it remains the study of ceramics and polymer materials. The indirect scattering of light or inter-reflections among the scratch zones must be considered too. Currently, only direct light scattering is taken into account. Another improvement can be done by simulating close views of scratches, where the geometry will then be visible. On the other way, pixels containing more than one scratch should be examined too. Maybe a multiresolution model with smooth transitions among the different scales would be an interesting avenue to pursue.

Acknowledgements

This project has been funded with grant numbers TIC2001-2392-C03 and TIC2001-2226-C02 from the MCyT of the Spanish Government, and 2001-SGR-00296 from the DURSI of the Catalan Government. We would like to thank Christophe Le Niniven (ENSIL, Limoges) for letting us use the measurement devices and Frederic Pérez for its helpful comments and suggestions. Maya software has been used under agreement with Alias®.

References

[BB90] BECKET W., BADLER N. I.: Imperfection for realistic image synthesis. *Journal of Visualization and Computer Animation 1*, 1 (1990), 26–32.

[BEPS96] BRISCOE B. J., EVANS P. D., PELILLO E., SINHA S. K.: Scratching maps for polymers. *Wear 200*, 1 (1996), 137–147.

[BL99] BUCHANAN J. W., LALONDE P.: An observational model for illuminating isolated scratches. In *Proc. Western Computer Graphics Symposium 1999 (SKIGRAPH'99)* (March 1999).

[Bli82] BLINN J. F.: Light reflection functions for simulation of clouds and dusty surfaces. In *Proc. SIGGRAPH '82* (1982), vol. 16, pp. 21–29.

[Buc01] BUCAILLE J.-L.: *Simulation numérique de l'indentation et de la rayure des verres organiques*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2001.

[Cal94] CALLISTER W. D.: *Materials Science and Engineering, an Introduction*, 3rd ed. John Wiley & Sons, 1994.

[DH96] DORSEY J., HANRAHAN P.: Modeling and rendering of metallic patinas. In *Proc. SIGGRAPH '96* (1996), pp. 387–396.

[FvDFH90] FOLEY J. D., VAN DAM A., FEINER S. K., HUGHES J. F.: *Computer Graphics: Principles and Practice*, 2nd ed. Addison-Wesley, 1990.

[Gla89] GLASSNER A.: *An Introduction to Ray Tracing*. Academic Press, 1989.

[JZLM98] JARDRET V., ZAHOUANI H., LOUBET J. L., MATHIA T. G.: Understanding and quantification of elastic and plastic deformation during a scratch test. *Wear 218* (1998), 8–14.

[Kaj85] KAJIYA J. T.: Anisotropic reflection models. In *Proc. SIGGRAPH '85* (1985), vol. 19, pp. 15–21.

[MDG01] MERILLOU S., DISCHLER J.-M., GHAZANFARPOUR D.: Surface scratches: measuring, modeling and rendering. *The Visual Computer 17*, 1 (2001), 30–45.

[ON94] OREN M., NAYAR S. K.: Generalization of lambert's reflectance model. In *Proc. SIGGRAPH '94* (1994), pp. 239–246.

[PF90] POULIN P., FOURNIER A.: A model for anisotropic reflection. In *Proc. SIGGRAPH '90* (August 1990), vol. 24, pp. 273–282.

[SKHL00] SUITS F., KLOSOWSKI J. T., HORN W. P., LECINA G.: Simplification of surface annotations. In *Proc. IEEE Visualization '00* (October 2000), pp. 235–242.

[TF88] TERZOPOULOS D., FLEISCHER K.: Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In *Proc. SIGGRAPH '88* (1988), vol. 22, pp. 269–278.

[War92] WARD G. J.: Measuring and modeling anisotropic reflection. In *Proc. SIGGRAPH '92* (1992), vol. 26, pp. 265–272.

[WNH97] WONG T. T., NG W. Y., HENG P. A.: A geometry dependent texture generation framework for simulating surface imperfections. In *Proc. Eurographics Workshop on Rendering '97* (1997), pp. 139–150.