

# A Geometry Dependent Texture Generation Framework for Simulating Surface Imperfections

Tien-Tsin Wong<sup>1</sup>

Wai-Yin Ng<sup>2</sup>

Pheng-Ann Heng<sup>1</sup>

<sup>1</sup>Department. of Computer Science & Engineering

<sup>2</sup>Department of Information Engineering

The Chinese University of Hong Kong

**Abstract:** To model surface imperfections and weathering, we propose a two-step texture generation framework in between manual texture synthesis and automatic physical simulation. Although the pattern of blemishes looks random, the systematic and geometry dependent nature of the underlying distribution is still observable. A distribution of *tendency* (potential to contain blemishes) is modeled in the first step, which includes user control and geometric information. The second generates and distributes an irregular blemish pattern according to the modeled tendency distribution. As examples we model three common surface imperfections; dust accumulation, patina and peeling.

## 1 Introduction

The modeling of natural imperfections is a technique to introduce imperfection in order to achieve realism. It is especially important when computer generated objects are overlaid with the live-action scenes. Unblemished surfaces make the composited scenes look unnatural and dissonant. Imperfection is a very general concept. In this paper, we consider *surface imperfections* or weathering.

A simple way to mimic weathering is to use traditional texture mapping. By compositing the texture map (either 2D or 3D), the user has full control of the whole weathering appearance. However, a professional artist may need to spend a lot of time to fine tune the texture details. On the other hand, physical simulation can automate the generation of the details and give a realistic appearance, but is less controllable due to the complexity of the simulation. Users have only limited control, by tuning some non-intuitive parameters. A more serious problem is that sometimes we do not know enough about the formation process of some surface imperfections. Here, we propose an empirical approach to modeling surface imperfections, in between manual texture synthesis and automatic physical simulation.

Even though the patterns of blemishes look irregular, the underlying geometry dependent blemish distribution is still observable in most cases. Therefore, we model this *tendency distribution*. *Tendency* is a scalar expressing the potential for occurrence of surface imperfections. The proposed framework allows intuitive control of the overall distribution of blemishes by positioning some abstract imperfection sources. The details of the distribution can then be automatically generated by utilizing geometric information. Once the tendency distribution is modeled, an undistorted blemish pattern can be generated with the guidance of this distribution. This framework is a generalization of the work of Hsu and Wong [17].

In Section 2 we discuss some related work, and in Section 3 overview the framework. The modeling of the tendency distribution is described in Section 4. Section 5 discusses how the modeled distribution guides the texture generation. Section 6 gives

three examples of imperfections modeled with the proposed framework, and their results. Conclusions and future directions are discussed in Section 7.

## 2 Related Work

Texture mapping [16] is a simple and powerful technique to increase the visual richness on an object surface. It is natural to use it to simulate surface imperfections, but it has a few drawbacks.

Firstly, it requires much human intervention during the composition of a realistic texture. Becket and Badler [3] proposed a system dedicated to the generation of a 2D blemish texture, which is then mapped to the 3D object surface. However, mapping 2D texture to 3D surface is also a well-known problem. If the object has a simple coordinate system (like a plane, quadric surface or parametric surface), a rectangular texture can simply be mapped by using it [4]. On the other hand, there is no canonical mapping to map a 2D texture to the surface of an irregular and aggregate object. Moreover, texture mapping usually suffers from distortion if the target surface is undevelopable<sup>1</sup>. Researchers [20] usually have to make trade off between the distortion and discontinuity during mapping. Peachey [23] and Perlin [24] considered that the fundamental source of this dilemma is that the dimension of the mapping domain does not match that of the range. They used the 3D texture domain to texture the surface. This approach is the well-known solid texturing.

Two seldom addressed characters of texturing are its dynamic nature and its environment dependency. The pattern on the skin of wild animal changes with time, and surface scratches grow as an object dilapidates. Another important feature is that the pattern changes according to the environment, knowledge of which is thus critical to modeling. Only a few papers have addressed these two texture features. Turk [28] divided the object surface into a mesh of cells and performed the reaction-diffusion process among these cells in order to simulate the formation process of the pattern on an animal skin. Geometric factors can be included in this process, to affect the pattern formed. Miller [22] determined surface accessibility which is closely related to the distribution of the patina on tarnished surface. Hsu and Wong [17] simulated the distribution of dust accumulation by calculating a dust amount, which is extended to the more general *tendency* in this paper. Dorsey and Hanrahan [10] modeled the appearance of patinas on object surfaces. Dorsey *et al.* [11] also simulated the formation process of washes and stains using a particle system. These particles interact with the environment factors, such as gravity, friction wind and roughness, during the formation process and hence produce realistic results.

## 3 Overview

The formation of blemishes may be due to human factors, physical laws, chemical reactions, etc. It is usually complex and sometimes even unknown. Although the pattern of surface imperfections looks irregular, one can still observe the underlying systematic distribution. Hence, the first step in our framework is to model this underlying distribution. We model the potential of occurrence of the blemish pattern at a surface point as a

---

<sup>1</sup> A surface is developable if it can be unfolded or developed onto a plane without stretching or tearing. An undeveloped surface is a surface that is not developable.

scalar quantity, namely *tendency*. The underlying distribution being modeled is called the tendency distribution.

Instead of simulating the formation process, we can imagine that the formation of blemishes is due to different kinds of *abstract imperfection sources*. For instance, scratches are usually found near the handle of a leather handbag, due to frequent human contact. In this case, we can imagine an abstract scratch source being positioned near the handle. Dust particles are usually accumulating on an upward-facing surface; a dust source can be imagined above the surface. This abstraction of imperfection sources, analogous to the concept of light sources, provides control of the overall tendency distribution. The user can specify where to introduce blemishes by placing the imperfection source nearby. Such control is familiar to the user due to its similarity to that of light sources. We discuss five types of imperfection sources in the next section.

However, using abstract imperfection sources alone to model the detail tendency distribution is not adequate, due to the non-trivial pattern of the distribution. Since the distribution is usually geometry dependent, the geometric information can be utilized to automate the generation of the detail distribution. In this paper, we study two such geometric factors, namely surface exposure and curvature.

Once the tendency distribution is modeled, the second step of the framework generates the irregular blemish pattern according to this distribution.

## 4 Tendency Determination

### 4.1 Abstract Imperfection Sources

Consider the *flux density* incident on a surface point  $P$  due to a specific imperfection source  $S_i$ . It represents the effect of an imperfection source on that surface point. The value of the flux density is determined by the geometric relationship between the surface point and the imperfection source. It takes value in the range  $[0, 1]$ . A value of 0 means the surface point is free of blemishes while a value of 1 means it is full of blemishes.

Similarly to light sources, we model various forms of imperfection sources; ambient, point, directional, spotlight, and slide projector. The flux densities of these five forms are given in Equation 1.  $F_{S_i}$  denotes the flux density due to the  $i^{\text{th}}$  imperfection source located at  $S_i$ . Figure 1 (see Appendix) shows the visual effect of these five kinds of sources. The region with high flux density is indicated in red.

$$\begin{aligned}
\text{(a): } & F_{S_i} = \text{constant}, \\
\text{(b): } & F_{S_i} = \frac{1}{\rho_i^c} \cos^{\eta_c}(\theta_i) \quad \text{or} \quad \frac{1}{\rho_i^c} \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right), \\
\text{(c): } & F_{S_i} = \cos^{\eta_c}(\theta_i) \quad \text{or} \quad \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right), \\
\text{(d): } & F_{S_i} = \frac{s}{\rho_i^c} \cos^{\eta_c}(\theta_i) \quad \text{or} \quad \frac{s}{\rho_i^c} \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right), \\
& \text{where } s = \text{smoothstep}(c_{out}, c_{in}, \mathbf{S}_i \mathbf{P} \cdot \mathbf{A}), \\
\text{(e): } & F_{S_i} = \frac{m}{\rho_i^c} \cos^{\eta_c}(\theta_i) \quad \text{or} \quad \frac{m}{\rho_i^c} \exp\left(\frac{-\theta_i^2}{2\eta_g^2}\right), \\
& \text{where } m = \text{map}(\mathbf{S}_i \mathbf{P}).
\end{aligned} \tag{1}$$

where  $P$  is the surface point of interest.

$S_i$  is the position of the  $i^{\text{th}}$  imperfection source.

$\rho_i$  is the distance between  $P$  and  $S_i$ .

$\theta_i$  is the angle between surface normal at  $P$  and vector  $\mathbf{PS}_i$ .

$\eta_c$  controls the size of the bump of the cosine function.

$\eta_g$  is the standard deviation which controls the size of the bump of Gaussian function.

$\mathbf{A}$  is the direction where the spotlight source pointing to.

$c_{in}$  is the cosine of half the angle at the apex of the inner spotlight cone. Surface points within this cone will be fully illuminated.

$c_{out}$  is the cosine of half the angle at the apex of the outer spotlight cone. Surface point outside this cone will not be affected by the source. Note that

$c_{out} \leq c_{in}$ .

The function  $smoothstep(a, b, v)$  returns 1 if  $v \geq b$  and returns 0 if  $v \leq a$ . It interpolates value between 1 and 0 if  $a \leq v \leq b$ .

Function  $map()$  returns the texture value indexed by the vector  $\mathbf{S}_i\mathbf{P}$ .

All forms in Equation 1 except the ambient source provide two flux density distributions, namely cosine and Gaussian distributions. These two functions give similar visual effect. The difference becomes significant only when the angle is greater than or equal to  $\frac{\pi}{2}$ . Cosine function returns zero value whenever the input angle  $\geq \frac{\pi}{2}$ , while the Gaussian function still returns non-zero value. This means cosine function may produce sudden visual change. However, the cosine function is computationally more efficient than the Gaussian function. This is analogous to the empirical Phong's cosine lighting model [25] and the physical lighting model <sup>2</sup> [27]. The choice of these two functions is provided as an option.

If more than one source are given, flux densities due to all sources are summed up together, and the power of each source can be independently controlled by a scaling factor  $\gamma_i$ . Moreover, a source can be either positive or negative. The negative effect can be achieved by multiplying a negative  $\gamma_i$  to the corresponding imperfection source. The overall flux density  $F_S$  of a surface point due to all imperfection sources is given by

$$F_S = \sum_{i=1}^m \gamma_i F_{S_i}, \quad (2)$$

where  $m$  is the total number of imperfection sources,

$F_{S_i}$  is any one of the flux density functions in Equation 1,

$\gamma_i$  is a scaling factor  $\in \mathfrak{R}$ .  $\gamma_i$  is positive if  $S_i$  is a positive source.  $\gamma_i$  is negative if  $S_i$  is a negative source.

Figure 1(f) (see Appendix) shows the visual effect of applying a positive directional source and a negative slide projector source on a sphere.

Up to now, we only discuss the modeling of the static distribution of the flux density. In fact, the growth of the flux density can also be simulated by gradually changing the value of each scaling factor  $\gamma_i$ . The upper row of Figure 9 (see Appendix) demonstrates the growth of the underlying tendency distribution which is modeled by gradually increasing the scaling factor associated with the point source.

<sup>2</sup> The physical lighting model is not exactly a Gaussian function, but the distribution of the orientation of microfacets is believed to be a Gaussian function. Gard [13] explicitly used the Gaussian function to model the surface reflectance.

## 4.2 Geometric Factors

Apart from the incident flux density, the local geometry affects the actual distribution of blemishes. Different kinds of blemish may behave differently towards the same geometry. Surface exposure and surface curvature are two significant geometric factors; others can also be incorporated into the model below. The final tendency  $T$  at a surface point of interest,  $P$ , is the overall flux density  $F_S$  perturbed by a function  $\alpha$  of all geometric factors  $G_j$ .

$$T' = F_S \cdot \alpha(G_1, G_2, G_3, \dots),$$

$$T = \begin{cases} 0 & \text{if } T' < 0, \\ T' & \text{if } 0 \leq T' \leq 1, \\ 1 & \text{if } T' > 1. \end{cases} \quad (3)$$

where  $G_j$  is the  $j^{\text{th}}$  geometric factor.

The perturbation function  $\alpha()$  is a function of all geometric factors and returns a positive real value. It returns 1 if there is no perturbation.

**Surface Exposure and Accessibility** The surface exposure at a surface point  $P$  is a measure of exposure to air, defined as follows. Consider a ray  $R_i$  from  $P$ , intersecting an obstacle at distance  $d_i$  (Figure 2(a)). As  $d_i$  increases (*i.e.*, the obstacle moves away from  $P$ ), the effect of that obstacle on the surface exposure  $G_\xi$  decreases, hence it is more exposed. We define a monotonic increasing weight function  $\omega$  of  $d_i$  that returns value in  $[0, 1]$  s.t.  $\omega(d_1) \leq \omega(d_2)$  if  $d_1 \leq d_2$ ,  $\omega(d_i) = 0$  when  $d_i = 0$ , and  $\omega(d_i) \rightarrow 1$  when  $d_i \rightarrow +\infty$  (*i.e.* no intersection). The following function satisfies our requirements and is easy to compute:

$$\omega(d_i) = \frac{d_i}{d_h + d_i} \quad \text{where } d_h > 0 \quad (4)$$

The constant  $d_h$  is the *half-exposure distance*, which is roughly the average spacing between  $P$  and other obstacles that would reduce the exposure to 0.5. It is an indirect parameter to define how far is far.

To determine  $G_\xi$ , infinite rays are fired from  $P$  and distributed evenly on the upper hemisphere of  $P$  (Figure 2(b)). Then  $G_\xi$  is defined as,

$$G_\xi = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \omega(d_i) \quad (5)$$

Exposure  $G_\xi$  will be 1 when  $P$  is completely exposed, since all  $\omega(d_i)$  equal 1. It will be 0 when  $P$  is touched by any object since all the  $\omega(d_i)$  are zero.



**Fig. 2.** (a): Ray  $R_i$  intersects with an obstacle. (b):  $n$  evenly distributed rays emitted from point  $P$ .

In practice,  $G_\xi$  is determined using ray casting [15, 14]. We cast only a finite number of sample rays, evenly distribute them on the upper hemisphere of  $P$  to detect the exposure. The same ray distribution pattern can be reused for other surface points on the object. Hence only one ray distribution pattern is needed to compute. The surface exposure is then approximated by the average of samples,

$$G_\xi \approx \frac{1}{n_0} \sum_{i=1}^{n_0} \omega(d_i) \quad (6)$$

where  $n_0$ , the number of rays emitted, is a finite constant.

$d_i$  is the distance between  $P$  and the point of intersection.

$\omega(d_i)$  is the weight function defined in Equation 4.

The more rays are fired, the more accurate the approximation is. However, this method inherits the same problems of ray casting, namely aliasing and expensive computation cost of intersection tests. The aliasing problem can be partially solved by supersampling, distribution ray-tracing [7] or cone-tracing [1]. The ray-object intersection tests can be sped up by object partitioning [19], spatial partitioning [12], and ray classification [2] techniques.

A more economical approach to determining  $G_\xi$  is described in [17]. Instead of determining an accurate  $G_\xi$  at each surface point of interest, we only need a very rough approximation of  $G_\xi$  by emitting a few number of rays in random directions (for anti-aliasing). Then  $G_\xi$  is evaluated using Equation 6 at each surface point. Although  $G_\xi$  is inaccurate at the microscopic level (at each surface point), it is accurate in the macroscopic level (*i.e.*, on average). The distribution of the surface exposure value in Figures 6 and 7 (see Appendix) are determined using this rough approximation with five random rays at each surface point. The drawback is that the resultant distribution becomes noisy due to the randomness.

Miller [22] defined two types of accessibilities which are two different approaches to measure surface exposure. These are tangent-sphere accessibility and offset-distance accessibility. Tangent-sphere accessibility at  $P$  is the radius of the largest sphere that touches  $P$  without intersecting any other object. Offset-distance accessibility is the distance to the nearest point on the offset surface minus the offset radius. To determine tangent-sphere accessibility, sphere-object intersection tests must be performed for all objects. However, not every kind of object representations has an efficient intersection algorithm with sphere. For example, sphere-metaball intersection test is expensive. Miller also demonstrated the determination of offset-distance accessibility on heightfield and voxel-based objects. Unfortunately, offset-distance accessibility is difficult to determine for objects with other representations. On the other hand, ray-object intersections are well studied [14]. Hence, the surface exposure by ray casting is preferred if sphere-object intersection test is inefficient. However, it is impossible to compute the *global* surface exposure by ray casting. Therefore, when global nature of the accessibility is a main concern, global tangent-sphere accessibility is preferred.

Determining surface exposure is closely related to the problem of visibility, on which a rich source of techniques can be found in the radiosity literatures [5, 26]. The surface exposure can be determined by obtaining six depth maps surrounding the surface point of interest (similar to the hemi-cube [6]). These depth maps are then used to compute  $G_\xi$  by averaging all  $s_i \omega(d_i)$  where  $d_i$  is the depth value at the  $i^{\text{th}}$  pixel, and  $s_i$  is the solid angle subtended by the  $i^{\text{th}}$  pixel element. Similarly to ray casting, this method can determine *local* exposure only.

For simplicity, we define the geometric function  $\alpha$  as a linear function,

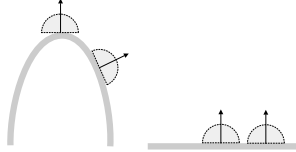
$$\alpha = 1 + r_0 G_\xi \quad (7)$$

where  $r_0$  is a scaling factor of surface exposure and  $r_0 \in \mathbb{R}$ .

$G_\xi$  is the surface exposure,  $0 \leq G_\xi \leq 1$ .

If the renderer is ray-tracing based, the surface exposure by ray casting can be determined on the fly. Only the visible area requires the surface exposure determination. For depth-buffer or scanline based renderers, where ray casting may not be available, surface exposure can be determined on the fly by the hemi-cube-like approach above. However, it is usually more efficient to precompute and store the surface exposure distribution for later use. Surface exposure of a polygonal object can be accurately determined and stored at each polygon vertex. The exposure value at an interior point of a surface polygon can be linearly interpolated among the values at its vertices. For other kinds of object representations, surface exposure can be determined at each surface point and stored in a texture map. Of course, a mapping should be defined before we know where in the 2D texture map (exposure map) to store the sampled exposure value. This approach does not suffer from distortion since the exposure is determined in the world space. The possible problem of this approach is aliasing due to the finite resolution of the 2D exposure map. Figure 6(b) shows an exposure map of the vertical plane behind the protrusive logo in Figure 6(a) (see Appendix).

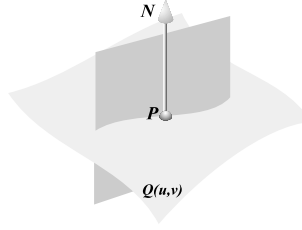
**Surface Curvature** Paint on protrusive surface is more likely to be peeled off than that on the flat one. Scratch or peeling usually appears on the protrusive area and then propagates to the surroundings. This is because the protrusive nature increases the chance of being attacked by external forces. Note that surface exposure alone cannot account for such character. Consider a convex surface and a flat surface (Figure 3). Both of them are completely exposed, but paint on the convex one has a higher potential of being peeled off.



**Fig. 3.** Paint on the convex surface has a larger tendency of being peeled off than that on a flat one, even though they have the same surface exposure.

For biparametric surface, the surface curvature can be analytically determined. For each plane containing the normal at a particular point  $P$  on the biparametric surface  $Q(u, v)$  (Figure 4), there exists a curvature  $\kappa$  of the intersection curve between the plane and the surface at  $P$ . As the plane is rotated about the normal, the curvature changes. The maximum ( $\kappa_{max}$ ) and the minimum ( $\kappa_{min}$ ) curvature are known as principal curvatures. Two combinations of the principal curvatures are useful (Equation 8). They are the average curvature  $G_{\kappa_a}$  and the Gaussian curvature  $G_{\kappa_g}$ .

$$G_{\kappa_a} = \frac{\kappa_{min} + \kappa_{max}}{2} \quad \text{and} \quad G_{\kappa_g} = \kappa_{min} \kappa_{max} \quad (8)$$



**Fig. 4.** The curvature at surface point  $P$  is defined as the curvature of the curve of intersection of the plane containing the normal and surface  $Q(u, v)$ .

$$G_{\kappa_a} = \frac{A|Q_v|^2 - 2BQ_u \cdot Q_v + C|Q_u|^2}{2|Q_u \times Q_v|^3} \quad \text{and} \quad G_{\kappa_g} = \frac{AC - B^2}{|Q_u \times Q_v|^4} \quad (9)$$

where

$$A = (Q_u \times Q_v) \cdot Q_{uu}, \quad B = (Q_u \times Q_v) \cdot Q_{uv}, \quad C = (Q_u \times Q_v) \cdot Q_{vv},$$

$$Q_u = \frac{\partial Q}{\partial u}, \quad Q_v = \frac{\partial Q}{\partial v}, \quad Q_{uu} = \frac{\partial^2 Q}{\partial u^2}, \quad Q_{vv} = \frac{\partial^2 Q}{\partial v^2}, \quad Q_{uv} = \frac{\partial^2 Q}{\partial u \partial v}$$

A detailed derivation can be found in [9].

Figures 8(a) and (b) (Appendix) show the magnitude of average and Gaussian curvatures on a Bézier patch teapot. High curvature is shown in red, low curvature in white. Note the difference between the two types of curvatures near the teapot handle.

For a mesh of polygons, the surface curvature can be approximated using the method suggested by Dehaemer and Zyda [8]. The curvature at one vertex of the polygon is approximated by the length of the vector resulting from the subtraction of the unit normal vectors of polygons sharing the vertex. Note that the adjacency among the polygons must be determined or provided before calculating the curvature.

Now we can use either the magnitude of average or Gaussian curvature or even both in the calculation of the geometric function  $\alpha$ . Function  $\alpha$  is now expanded as,

$$\alpha = 1 + r_0 G_\xi + r_1 |G_{\kappa_a}| + r_2 |G_{\kappa_g}| \quad (10)$$

where  $r_0, r_1$  and  $r_2 \in \mathfrak{R}$  are scaling constants.

$G_\xi \in [0, 1]$  is the surface exposure.

$G_{\kappa_a} \in \mathfrak{R}$  is the average curvature.

$G_{\kappa_g} \in \mathfrak{R}$  is the Gaussian curvature.

Whenever the surface curvature is determined, it can be stored for later use just as the case of surface exposure.

## 5 Tendency-Guided Texture Generation

Once the tendency value  $T$  is determined, texture can be generated according to the tendency distribution. More blemishes should be generated at a place with higher tendency. In order to prevent texture distortion, solid texturing [23, 24] is used. A stochastic texture synthesis method is required to generate an irregular blemish pattern. The 3D



texture can be obtained by two methods: scanning a real material and fitting a mathematical model. Materials like wood and marble can be scanned in a slice-by-slice basis. However, a large volume of data must be scanned in order to reduce periodicity in the resultant texture pattern. An alternative is to evaluate a mathematical function  $f : \mathbb{R}^3 \mapsto \mathbb{R}$  which generates a scalar value given the positional vector. Two popular models are fractional Brownian motion (fBm) [21] and Perlin's noise and turbulence functions [24]. Blemish pattern in Figures 6(a) and 7(b) (see Appendix) are generated using Perlin's noise function while Figures 9 and 10 in the Appendix are generated by fBm function.

## 6 Results

### 6.1 Dust Accumulation

Dust accumulation takes place in any environment which has not been cleaned for a while. A less inclined surface accumulates more dust particles than a steeper one, hence a vertical directional dust source usually suffices to simulate the tendency distribution of dust accumulation. Figure 5 (Appendix) shows a sphere with more dust particles on the North pole than on the equator. Another observation is that more dust particles accumulate on surfaces with less exposure, which provide the dust particles with shelter where they have less chance of being removed by wind or other external forces. We mimic this by scaling up the effect of the surface exposure  $G_\xi$  when modeling the tendency distribution. Figure 6(a) (see Appendix) shows a plane with the protrusive logo surrounded by dust particles. This is done by applying a vertical directional dust source and exaggerating the surface exposure effect. To model a scraping off effect on dusty surface, we use a negative slide projector dust source. Figure 5 in the Appendix shows the scraping off effect on a dusty sphere. The fuzzy appearance of a dust layer is simulated by the following modulation of the surface properties:

$$\begin{aligned} T'' &= T \cdot \text{noise}(P) \\ S_{\text{fuzzy}} &= T''S_{\text{dust}} + (1 - T'')S_{\text{object}} \end{aligned} \quad (11)$$

where  $S_{\text{fuzzy}}$  is the resultant surface property.

$S_{\text{dust}}$  is the surface property when fully covered with dust, it is a user-defined constant.

$S_{\text{object}}$  is the surface property of the object.

$T$  is the final tendency evaluated using Equation 3.

Function  $\text{noise}()$  is Perlin's noise function, which returns a value in  $[0, 1]$ .

$P$  is the positional vector.

### 6.2 Patina

As patina forms on a tarnished surface, the less exposed region provides more shelter for it to grow and accumulate. Since there is less directional preference in forming patina, an ambient patina source is used. Like dust accumulation, the effect of surface exposure is scaled up to visualize the sheltering effect. Figure 7(b) (see Appendix) shows a Beethoven statue full of patina.

### 6.3 Peeling

Peeling often appears on the surfaces of multi-layered objects, such as gilded and leather objects. It is usually seen at places with high surface curvature. There is less directional preference in the formation of peeling, hence an ambient peeling source is used. Since peeling is highly affected by surface curvature, the effect of surface curvature is scaled up. Figure 9 (see Appendix) shows a peeled teapot (lower row) and its underlying tendency distribution (upper row). Besides the ambient source, other types of source can also be used when the peeling is formed by external force. We positioned a point source near the cover of the teapot in Figure 9 to model the effect of frequent contact with external force. The irregular peeling pattern is simulated in the following way:

$$T'' = T \cdot fBm(P)$$

$$S_{\text{final}} = \begin{cases} S_{\text{layer}_0} & \text{if } 0 \leq T'' < \lambda_0, \\ S_{\text{layer}_1} & \text{if } \lambda_0 \leq T'' < \lambda_1, \\ \dots & \dots \\ S_{\text{layer}_i} & \text{if } \lambda_{i-1} \leq T'' < \lambda_i, \\ \dots & \dots \\ S_{\text{layer}_m} & \text{if } \lambda_{m-1} \leq T'' < \lambda_m, \end{cases} \quad (12)$$

where  $m + 1$  is the total number of surface layers.

$S_{\text{final}}$  is the final surface property.

$S_{\text{layer}_i}$  is the surface property of the  $i^{\text{th}}$  layer below the outermost surface  $S_{\text{layer}_0}$ .

$\lambda_i \in [\lambda_{i-1}, 1]$  is a user-defined threshold for the  $i^{\text{th}}$  layer, where by convention  $\lambda_{-1} = 0$ .

$T$  is the final tendency evaluated using Equation 3.

$fBm()$ , the fractional Brownian motion function, returns a value in  $[0, 1]$ .

$P$  is the positional vector.

Figure 9 in the Appendix shows four frames from two animation sequences. The lower ones show the appearance of the growth of peeling, while the upper demonstrates the growth of the underlying tendency distribution. The growth is simulated by gradually increasing the power of each peeling source. Figure 10 (Appendix) shows a synthesized image of a peeled teapot, a brand new spoon, and a teacup.

## 7 Conclusions and Future Work

The use of traditional texture mapping to create the appearance of blemish surface requires a lot of human intervention and is difficult to apply to irregularly shaped objects. The distortion during the mapping process further complicates the composition of the desired texture. Moreover, traditional methods do not model the underlying distribution of blemishes while such distribution is critical for the synthetic imperfect surface to look real.

The procedural framework proposed in this paper allows the user to intuitively control the overall tendency distribution by positioning various imperfection sources around the object. This control ability is lacking for the physical approaches due to the complexity of simulation processes. Small details are automatically generated by utilizing the geometric information. Hence the user is freed from detailed adjustments. The separation of the tendency determination from the generation of blemish pattern gives extra freedom to switch between different pattern generation methods. Solid

texturing is used in our work to generate undistorted blemish patterns due to its simplicity and efficiency. Fascinating texturing schemes like texture with volume [18] can also be used. We model dust accumulation, patina and peeling by modeling the underlying tendency distribution, producing some convincing images.

As the number of abstract imperfection sources increases, so does the calculation cost of tendency values at each surface point. This may not cause serious problems, as each object usually owns only a few private (or local) imperfection sources. Imperfection sources of one object cannot affect another object. This is quite different from the case of light sources, which are global to all objects in the environment.

If a library of abstract imperfection sources and a library of geometric factors are built up, a wide range of blemish distribution can be modeled. In this paper, we already demonstrate five kinds of sources and two kinds of geometric factors.

The proposed framework can only model the simple growth behavior of the underlying tendency by scaling the parameters  $\gamma_i$ . More complex growing behavior like fluid flow appearance of some rusty surfaces cannot be conveniently simulated. A cellular simulation approach [28] and particle system [11] can be used to solve the problem. However, the computation cost will increase tremendously. Nevertheless, our approach can still be used as the initial tendency distribution.

Currently the tendency is expressed as a scalar quantity. If it is generalized to a tensor quantity, we may be able to model the anisotropy of the texture patterns found in some real life surfaces. Patterns like scratch are usually anisotropic, since the external force which introduces the pattern is usually directional. This effect is less convenient to model with the current scalar tendency.

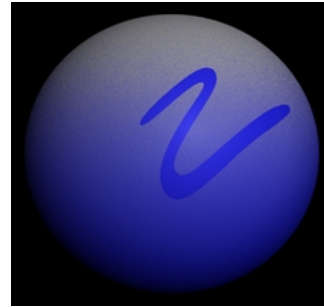
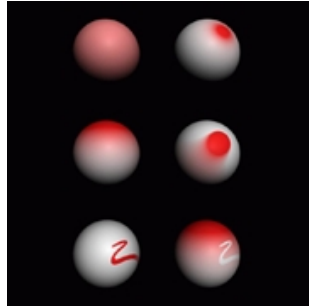
## Acknowledgements

We would like to thank Prof. Tim Poston of Institute of Systems Science, National University of Singapore and Dr. Wai-Shing Luk of Katholieke Universiteit Leuven for their careful proofread and helpful suggestions. Thanks also to the reviewers for their valuable suggestions and careful review. This work was supported by the CUHK Postgraduate Student Grants for Overseas Academic Activities.

## References

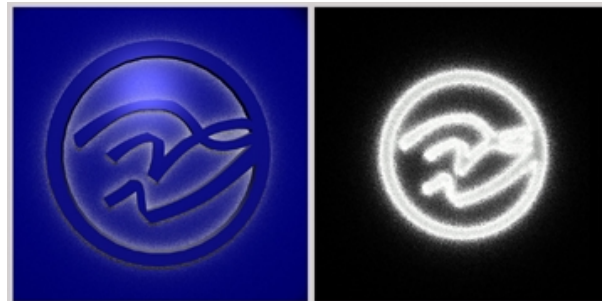
1. J. Amanatides. Ray tracing with cones. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 129–135, July 1984.
2. J. Arvo and D. Kirk. Fast ray tracing by ray classification. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 55–64, July 1987.
3. W. Becket and N. I. Badler. Imperfection for realistic image synthesis. *Journal of Visualization and Computer Animation*, 1(1):26–32, August 1990.
4. E. Catmull. *Subdivision Algorithm for the Display of Curved Surfaces*. PhD thesis, University of Utah, 1974.
5. M. Cohen and J. Wallace. *Radiosity and Realistic Image Synthesis*. AP Professional, 1993.
6. Michael Cohen and Donald P. Greenberg. The hemi-cube: a radiosity solution for complex environments. In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 31–40, August 1985.
7. R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. In Hank Christiansen, editor, *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 137–145, July 1984.

8. M. J. Dehaemer and M. J. Zyda. Simplification of objects rendered by polygonal approximations. *Computer & Graphics*, 15(2):175–184, 1991.
9. J. C. Dill. An application of color graphics to the display of surface curvature. In *Computer Graphics (SIGGRAPH '81 Proceedings)*, volume 15, pages 153–161, August 1981.
10. J. Dorsey and P. Hanrahan. Modeling and rendering of metallic patinas. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 387–396, August 1996.
11. J. Dorsey, H. K. Pedersen, and P. Hanrahan. Flow and changes in appearance. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 411–420, August 1996.
12. A. Fujimoto, T. Tanaka, and K. Iwata. Arts: Accelerated ray-tracing system. *IEEE Computer Graphics and Applications*, 6(4):16–26, April 1986.
13. Gregory J. Gard. Measuring and modeling anisotropic reflection. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.
14. A. Glassner, editor. *An Introduction to Ray Tracing*. Academic Press, 1989.
15. R. A. Goldstein and R. Nagel. 3-D visual simulation. *Simulation*, 16(1):25–31, January 1971.
16. P. S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, Nov. 1986.
17. Siu-Chi Hsu and Tien-Tsin Wong. Simulating dust accumulation. *IEEE Computer Graphics and Applications*, 15(1):18–22, January 1995.
18. J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. In *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23, pages 271–280, July 1989.
19. T. L. Kay and J. T. Kajiya. Ray tracing complex scenes. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 269–278, August 1986.
20. J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 27–34, August 1993.
21. B. B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman and Co., New York, rev 1977.
22. G. Miller. Efficient algorithms for local and global accessibility shading. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 319–326, 1994.
23. D. R. Peachey. Solid texturing of complex surfaces. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 279–286, July 1985.
24. K. Perlin. An image synthesizer. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 287–296, July 1985.
25. B.-T. Phong. Illumination for computer-generated pictures. *Communication of ACM*, 18(6), June 1975.
26. F. X. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, 1994.
27. K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. In *Journal of Optical Society of America*, volume 57, pages 1105–1114, September 1967.
28. G. Turk. Generating textures for arbitrary surfaces using reaction-diffusion. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 289–298, July 1991.



Different forms of sources. (a)top-left, ambient form. (b)top-right, point form. (c)middle-left, directional form. (d)middle-right, spotlight form. (e)bottom-left, slide projector form. (f)bottom-right, a negative slide projector + a positive directional sources. (Wong, Ng & Heng, Fig. 1)

A dusty sphere with a scrap pattern. A vertical directional and a negative slide projector dust source are applied. (Wong, Ng & Heng, Fig. 5)



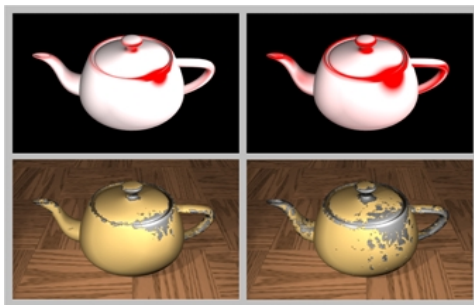
(a)left: Dusty logo. Simulated by a vertical directional dust source and exaggerating the effect of the surface exposure. (b)right: The corresponding exposure map of the vertical plane behind the protrusive area. (Wong, Ng & Heng, Fig. 6)



(a)left: A brand new Beethoven statue. (b)right: A few years later. Patina formed on less exposed regions. (Wong, Ng & Heng, Fig. 7)



(a)left: Average curvature of a teapot. (b)right: Gaussian curvature of a teapot. (Wong, Ng & Heng, Fig. 8)



Growth of peeling. From left to right, the scaling factors of the peeling sources are increased. The upper row shows the underlying tendency distribution, while the lower row shows the appearance of the peeling based on the corresponding tendency distribution. (Wong, Ng & Heng, Fig. 9)



A synthesized image of a peeled teapot and other brand new teacup and spoon. (Wong, Ng & Heng, Fig. 10)