

Context-Aware Textures

JIANYE LU, ATHINODOROS S. GEORGHIADES, ANDREAS GLASER, and HONGZHI WU

Yale University

LI-YI WEI, and BAINING GUO

Microsoft Research Asia

JULIE DORSEY and HOLLY RUSHMEIER

Yale University

Interesting textures form on the surfaces of objects as the result of external chemical, mechanical, and biological agents. Simulating these textures is necessary to generate models for realistic image synthesis. The textures formed are progressively variant, with the variations depending on the global and local geometric context. We present a method for capturing progressively varying textures and the relevant context parameters that control them. By relating textures and context parameters, we are able to transfer the textures to novel synthetic objects. We present examples of capturing chemical effects, such as rusting; mechanical effects, such as paint cracking; and biological effects, such as the growth of mold on a surface. We demonstrate a user interface that provides a method for specifying where an object is exposed to external agents. We show the results of complex, geometry-dependent textures evolving on synthetic objects.

Categories and Subject Descriptors: I.3.7 [**Computing Methodologies**]: Computer Graphics—*Three-dimensional graphics and realism; Color, shading, shadowing, and texture.*

General Terms: Algorithms, Experimentation, Measurement.

Additional Key Words and Phrases: Aging, data capture, natural phenomenon, realistic rendering, texture synthesis, weathering.

1. INTRODUCTION

Weathering and aging produce complex texture variations on objects. The extent and nature of the changes depend on the object’s global context—where in the world it is sitting, and to what it has been exposed. The variation of texture at any point on the object depends on its local geometric context—parameters such as its curvature, which way it is oriented, or the thickness of the surface. The nature of how texture varies with local geometry depends on the specific type of weathering or aging. In this paper, we present a methodology for capturing and transferring geometry-dependent texture variations for a wide variety of weathering and aging processes.

Previous approaches to producing the appearance of weathering or aging consist primarily of painting effects by hand or performing numerical simulations. These are respectively labor and computationally intensive approaches. Recent work in all areas of graphics has shown the power and effectiveness of using captured data to produce complicated, natural and subtle effects. The key issue in the capture of physical data is putting it in a form that allows its reuse in novel ways. Our new contribution to the specific area of texture capture is measuring the dependence of texture on geometry for different weathering and aging processes.

We begin by reviewing previous work in weathering and aging, shape and texture capture, and texture transfer. We next describe our pipeline for capture, analysis and texture synthesis. We describe in detail a series of experiments designed to capture process-dependent geometric context parameters. We present the analysis of the captured data, and the method we use for transferring the textures augmented with context

parameters. We describe a GUI that can optionally be used to control where aging processes are applied to an object. We close with the presentation of a variety of examples of applying context-aware textures to new objects.

2. PREVIOUS WORK

The methodology we present for capturing and transferring textures with context builds on previous work on modeling the weathering and aging of materials, 3D shape and appearance capture, and texture synthesis techniques. In this section, we describe previous work in weathering in terms of a taxonomy of effects, previous work in 3D scanning as it extends measurements and models from other fields, and texture synthesis techniques based on captured data. We cite examples of previous work in these areas; comprehensive surveys are beyond the scope of this paper.

2.1 Material Aging and Weathering Taxonomy

Modeling texture variations associated with aging and weathering is critical to generating input for realistic scenes for feature films and games. In practice, such textures are produced by skilled combinations of processed digital photographs and artistic digital painting, using methods such as those described by Demers [2001]. To provide more powerful tools for modeling texture beyond pixel-by-pixel painting and manipulation of texture coordinates, many researchers have sought methods to systematically produce complex textures associated with specific physical processes. In addition to providing tools to the feature film and game industries, such tools would make realistic material modeling accessible in applications in which labor intensive processes are not financially feasible.

A wide variety of weathering simulation methods are summarized in Table (I), which includes information on effects, model parameters, performance, and validation. Only a small subset of possible weathering and aging effects has been modeled, indicating that there is a great deal of work left to be done in this area. The times to simulate phenomena on objects are still substantial. We have not adjusted timings to account for advances in hardware performance, primarily because such advances are more than offset by the increased size of the geometric models that have become common. The quality of existing methods is difficult to assess because of the rather limited validation data. Many methods have not been validated at all. Where validation *has* been performed, it has generally not been applied to reproducing the full appearance caused by aging effects, but has been focused on particular aspects of the phenomenon being studied, such as the time of onset of cracks, or the general shape of patterns being formed.

Aging effects can be classified by the type of process:

Chemical processes transform the original material into a substance with a different composition and different physical characteristics. Examples of models of chemical compounds changing include patination [Dorsey and Hanrahan 1996; Chang and Shih 2000], corrosion [Mérillou et al. 2001a; Chang and Shih 2003] and stone weathering [Dorsey et al. 1999]. An example of phase change is the work on drying by [Lu et al. 2005].

Mechanical processes involve the disintegration of a material into fragments or the transport and deposition of material, such as dirt and salts. Examples of mechanical stress and fracture include cracking and peeling [Hirota et al. 1998; Hirota et al. 2000; Gobron and Chiba 2001a; Gobron and Chiba 2001b; Aoki et al. 2002; Paquette et al. 2002], scratches [Mérillou et al. 2001b; Bosch et al. 2004] and impacts [Paquette et al. 2001]. Examples of transport processes include dust accumulation [Hsu and Wong 1995] and flow [Dorsey et al. 1996].

Biological processes involve the development of patterns on surfaces due to living organisms, such as algae, mold, and fungi. An example of material appearance affected by biological growth is the simulation of lichen colonization [Desbenoit et al. 2004].

paper	effect	parameters [†]	time	data size	validation
[Dorsey and Hanrahan 1996]	patina	accessibility, surface inclination & orientation	n/a	n/a	rendering only
[Chang and Shih 2000]	patina	accessibility, gravity, curvatures, moisture in soil	n/a	n/a	rendering only
[Mérillou et al. 2001a]	rust & patina	“imperfection factor,” layer protection, object collision, aeration	n/a	n/a	prediction of rate and spread of corrosion
[Chang and Shih 2003]	rust	curvatures, accessibility, orientation, current, salts	n/a	n/a	rendering only
[Dorsey et al. 1999]	erosion, efflorescence, discoloration	mineral concentration & solubility, decay index, exposure map, max. saturation, permeability, water pressure & density, fluid velocity, porosity, stone density, viscosity	24 hr.	2.2M tri.	rendering only
			3 hr.	1.3M tri.	
[Lu et al. 2005]	drying	accessibility, distance to wet/dry boundary	n/a	n/a	ground-truth comparison
[Hirota et al. 1998]	cracks	spring constant, mean and var of max strain, surface layer depth & contraction ratio, material density, time scale	24 hr.	20480 tri.	rendering only
[Hirota et al. 2000]	3D cracks		8 hr.	1000 cu.	density of cracks; speed of formation
[Gobron and Chiba 2001a]	cracks	resistance, stress	3 hr.	2017 tri.	rendering only
[Gobron and Chiba 2001b]	peeling	n/a	11 min.	1.8M cells	rendering only
[Aoki et al. 2002]	cracks	cell size, spring constant & max strain; moisture content & diffusion constant	8 hr.	8000 cu.	temporal development of cracks
			1 hr.	700 cu.	
[Paquette et al. 2002]	crack & peeling	tensile stress, break strength, crack strength, deformation; elastic relaxation distance, shearing stress, adhesion strength; crack width, adhesion width	78 min.	400k poly.	visual quality
			23 min.	150k poly.	
			6 min.	130k poly.	
[Mérillou et al. 2001b]	scratch	surface type	n/a	n/a	scratch appearance
[Bosch et al. 2004]	scratch	material hardness; tool shape, orientation, force	n/a	500 scratches	scratch appearance
[Paquette et al. 2001]	impacts	tool shape, hit path	2 hr. ‡	n/a	impact size and density
[Hsu and Wong 1995]	dust	surface slope, stickiness, exposure; dust source	n/a	n/a	rendering only
[Dorsey et al. 1996]	flow	material roughness, rate & capacity of absorption; deposits adhesion rate, solubility rate; water particle mass, position, velocity, soluble materials; rain, sunlight	3 hr.	450 poly.	rendering only
[Desbenoit et al. 2004]	lichen	accessibility, light and moisture (from simulation)	n/a	n/a	rendering only
[Wong et al. 1997]	general	sources; surface exposure, accessibility, curvature	n/a	n/a	rendering only

Table I. Survey of previous work on synthesizing aging effects, including information on parameters, performance and validation. ([†]For more information about the model parameters, please refer to the original publications. [‡]Time spent on user interaction, not on simulation or rendering.)

Most of this work applied first principles simulation to model appearance history due to individual aging effects, using context parameters based on underlying processes. (An exception is the work of [Wong et al. 1997], which developed a framework for surface imperfections, such as dust and peeling.) In this paper, we present a general approach to model a wide range of weathering effects. We refer to previous work for possible control parameters, verify them, and use them to direct appearance transfer, rather than running first-principles simulations. Section 3.2 will give a detailed analysis on the context parameters we choose for different aging effects. In Section 5, we will validate our approach and compare its performance with that in previous work.

2.2 3D Scanning of Shape and Appearance Capture

As part of our proposed pipeline, we perform extensive physical capture, which includes measuring the shape and other environmental factors along with the time-varying appearance of objects subjected to different physical phenomena. From this captured data, we determine “unlit” textures, which vary both spatially and temporally, and relate them to quantifiable context parameters determined from the measured shape and environmental factors.

Since both the modeling of shape and appearance are labor intensive processes, the automated scanning of existing objects has been pursued as a method of obtaining detailed digital models for computer graphics applications. Laser triangulation and time-of-flight instruments [Blais 2004] have developed over the past twenty years from a tool for reverse engineering and high-end engineering applications to being used for visual applications, such as cultural heritage presentation. The processing of 3D scanned data has become an active area of computer graphics research. We make extensive use of shape capture with the express purpose of relating the spatiotemporal change of appearance of an object to its geometric shape.

While surface reflectance is of interest in other fields, and had been studied extensively for decades in optics, new techniques needed to be developed for measuring reflectance for computer graphics. The methods for measuring reflectance, and the catalogs of existing data developed for other fields, were oriented to different needs. Data was collected for specific wavelengths or angular configurations, and the full range of data needed for rendering realistic materials was not available. In parallel to developments in shape capture, methods specific to computer graphics were developed to capture surface properties, such as BRDF [Marschner et al. 1999; Matusik et al. 2003] and BTF [Dana et al. 1999], [Mueller et al. 2005]. The need for modeling the complete object shape and appearance has led to the development of systems that capture both. A survey of techniques is presented in [Bernardini and Rushmeier 2002], and an example of a complete shape and appearance capture method is given in [Lensch et al. 2003]. Capturing time-varying appearance and relating it to shape and other parameters distinguishes our approach from previous capture techniques. Unlike them, we are capturing a new class of textures, which incorporate their relationship to geometry and environmental factors.

2.3 Texture Synthesis

The work by Heeger and Bergen [1995] and de Bonet [1997] introduced one of the first approaches of synthesizing compelling textures using pyramid-based/multi-resolution approaches. These generally work on highly stochastic textures. Inspired by the more recent work by Efros and Leung [1999] in generating extended fields of texture from small samples, there has been a great deal of activity in computer graphics in efficiently generating textures from small captured data sets (e.g. [Wei and Levoy 2000]), and applying them to 3D objects [Turk 2001; Wei and Levoy 2001; Gorla et al. 2003]. Such algorithms are broadly classified as either pixel-based [Turk 2001; Wei and Levoy 2001; Tong et al. 2002; Wei and Levoy 2002; Zhang et al. 2003; Lefebvre and Hoppe 2005], or as patch-based [Soler et al. 2002; Kwatra et al. 2003; Zhou et al. 2005].

As proposed by Wong et al. [1997], the textures representing imperfections from weathering and aging should depend on geometry. Rather than just correcting for perspective distortions when transferring

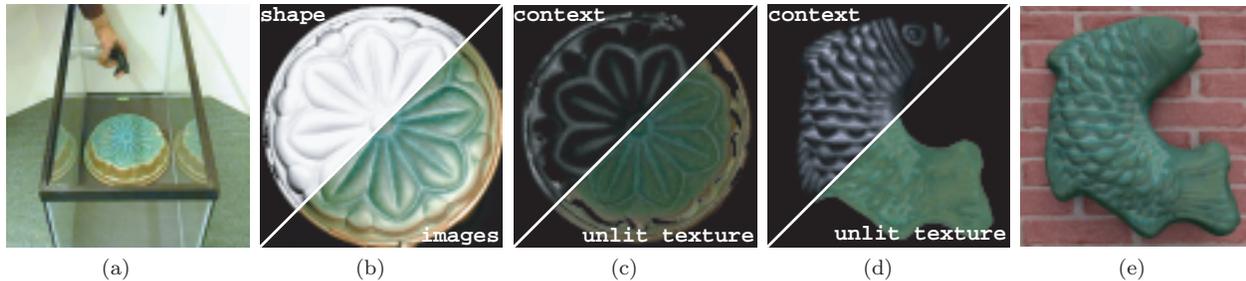


Fig. 1. (a) We conduct controlled experiments to capture how the texture of an object changes as the result of external agents; (b) We capture the shape and the texture history under controlled illumination; (c) We calculate the unlit textures and relate them to the context parameters; (d) and (e) Guided by the context parameters, we transfer the texture history onto a new object.

texture, additional geometric effects need to be considered. Zhang et al. [2003] presented a method for synthesizing textures that progressively vary across surfaces. To apply a progressively varying texture, a sample that relates the texture to a control function is needed. The distribution of that control function on the target object also needs to be specified. This approach has likewise been used for the image analogies in [Hertzmann et al. 2001].

In the methods presented here, we show how to capture the relationship between material changes and geometric properties for a variety of chemical, mechanical and biological effects. This relationship then defines the control variable on the surface of the target object, so that the transferred texture adapts to the specific local geometry.

3. THE PIPELINE

Our system combines techniques from very different areas. In this section, we will provide a short overview of its main components, which can be divided into two categories: procedures for generating a context-aware texture and tools and algorithms that are necessary to apply this texture to a new synthetic object. (See Fig. 1.)

The generation of a context-aware texture begins with a set of controlled experiments on real objects (Fig. 1a). As the desired effect develops, we capture the shape and the changing texture of the test object at regular time steps (Fig. 1b). In addition, we try to control and measure any other factors that may influence the effect. These procedures will be described in detail in Section 3.1.

In general, the texture change caused by a physical phenomenon will depend on the shape of the object and other experimental parameters. As a second step we analyze this relationship. We identify context parameters that describe shape and other factors, which can approximately model the systematic texture changes (Fig. 1c). A detailed description of the data analysis is presented in Section 3.2.

The steps above have to be performed just once for each texture, which models a certain phenomenon. To apply the texture to a new shape, we provide a transfer algorithm, which is based on texture synthesis techniques. The synthesis is guided by the context parameter in order to model the systematic variations of the texture (Fig. 1d and e). This algorithm is described in Section 3.3.

Context-aware textures can be generated fully automatically, assuming a full object is affected by a process. However, in many cases only part of an object may be exposed to an agent that causes aging. We therefore provide a GUI that allows the user to spatially control and modify the control parameters on the target object. Since the control parameters describe the physical reality, they provide an intuitive and natural tool for the user to modify the texture on the target shape. The GUI is described in Section 4.

category	experiment	source object	context parameters				
			occlu.	pdir.	curv.	sdir.	thick.
chemical	patina	round copper shape	+				
	rusting (a)	ripple pattern	+			+	
	rusting (b)	statue of Mozart	+			+	
	rusting (c)	iron pipe	+			+	
mechanical	paint cracking	flat slide					+
	cracking paste	wave pattern		+	+		+
	dirt accumulation	statue of Bach	+			+	
biological	mold on cheese	statue of Bach	+				

Table II. Table for experiments and their categories. For rusting, we used (a) iron metallic surfacer with rusting solution, (b) iron metallic surfacer in vinegar vapor, and (c) black iron pipe in vinegar vapor. Context parameters include ambient occlusion (occlu.), principal direction (pdir.), surface curvature (curv.), surface orientation w.r.t. source direction (sdir.), and coating thickness (thick.)

3.1 Procedure for Physical Capture

Our approach to modeling changes in texture as a result of chemical, mechanical, and biological processes is to prepare object samples, control the agents that initiate the processes, and capture the appearance of the object as the processes evolve over time. Table (II) summarizes the experiments conducted to produce the source data for the results shown in this paper. We have selected processes of each type in the taxonomy to demonstrate the common features in the experiments and subsequent analysis.

For each type of process, we need to select a material and an object composed of, or coated with, the material. Our goal is to find objects with appropriate geometric variations, so that we can relate change to geometric properties. Finding an appropriate test object is a matter of trial and error, dictated by the shapes and materials available, and informed by performing a number of pilot experiments to observe when and where noticeable changes in material texture occurred.

Exposure to the agents causing texture changes is controlled by performing experiments indoors, keeping objects in enclosures to minimize the effects of air flow, and controlling light exposure and temperature levels. Although the control on light and temperature was not precise, we controlled the conditions to the extent that we could to make sure that the effects observed were not the result of spurious handling or disruption of the objects.

Prior to each experiment the object was scanned using the setup shown in Figure 2. A ShapeGrabber laser triangulation scanner is used to scan the object shape. An Olympus C8080WZ color camera, calibrated with respect to the ShapeGrabber coordinate system, captures images in each view of the object under lighting from five small, halogen lights, whose positions have been measured in the scanner coordinate system. A description of the design and calibration of this system is given in [Farouk et al. 2003]. The scan data is processed into a 3D shape using the scan processing system distributed by the Visual Computing Laboratory of ISTI-CNR [Callieri et al. 2002]. The color images are processed to eliminate the effects of illumination direction and to produce maps of the diffuse component of surface reflectance using methods described in [Bernardini et al. 2001]. Although our samples are not all Lambertian, the diffuse texture maps act as maps of where change has occurred on the surface. A texture map is computed for the entire object. A complete shape scan is subsequently performed at each time step, so that by aligning the scanned shapes the texture maps for all time steps are aligned and changes can be tracked at each texture-map pixel.

This general procedure was performed for each of these specific cases:

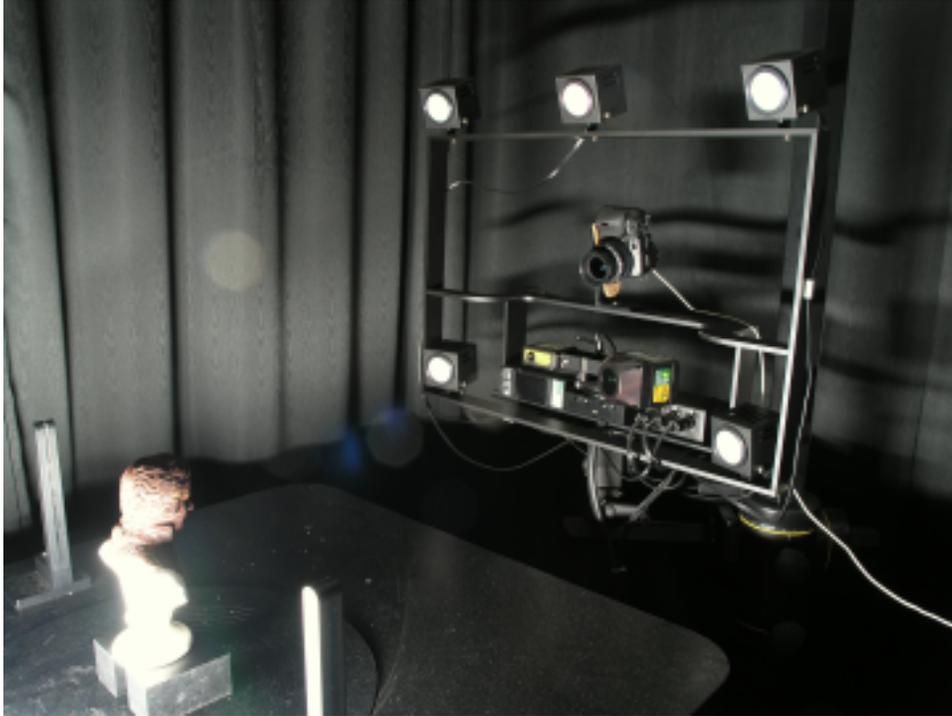


Fig. 2. The ShapeGrabber scanner and camera setup.

Patination. We captured patination effects using a copper kitchen pan as our test object. We uniformly sprayed a solution, consisting of 10 grams each of ammonium chloride and ammonium acetate dissolved in one liter of water, on the object every 12 hours for 11 days. During this period, the copper surface progressed to being covered with a blue-green patina.

Rusting. We captured the appearance of rusting using shapes covered with an iron surfacer solution and shapes of solid iron. Using the surfacer solution allowed us to examine more complex geometries than we could find in solid iron. In separate experiments, we used different agents to initiate rusting. In one case, we sprayed on the object a commercial chemical solution specifically formulated to accelerate the formation of rust. We treated the object periodically and always sprayed from the same direction. In another case, we enclosed the iron-coated objects in a bucket with vinegar vapor. The vinegar vapor source was consistently placed relative to the object. The third setup was quite similar, except that we instead enclosed a solid black iron pipe in vinegar vapor.

Paint cracking. We produced paint cracking by spraying layers of two different color paints on a transparent film. The paint in the second layer was a commercial product specifically formulated to crack rapidly as it dried. The first layer of paint was allowed to dry before the application of the second. A digital camera was set to capture images at high-speed as the cracks formed. Since we focused only on a flat surface, we calibrated for the camera distortion and roughly removed the illumination, rather than using the ShapeGrabber set-up. It was suggested in [Shorlin et al. 2000] that paint thickness controls the speed of crack formation, the cracking patterns, and their relative scales. The factor we varied then was paint thickness, rather than surface shape. We estimated the paint thickness distribution on the sample by

measuring its transparency using a back-light projector.

Cracking Paste. We captured the effect of thicker coatings cracking using a mixture of water, powdered corn starch, and a commercial cracking paste. In this case, a 3D shape was covered by this mixture and dried indoors in still air. We can estimate the coating thickness by registering and comparing shapes before and after the mixture application.

Dirt Accumulation. We captured the effect of dirt accumulation by collecting fine-particle dirt from outdoors and dispersing it in a cloud by passing it through a sieve held in the air. The sample object was set in the midst of the dispersed particles.

Biological Mold Growth. In order to have a solid shape that provided nutrients for mold growth, we coated a small statue with a thin coat of cream cheese. The coated statue was then sealed in a container with pieces of moldy bread providing spores, and a sponge with water providing moisture. The container was kept in a box with a heat source, but in shadow. The object with cream cheese was taken out periodically to capture its shape and appearance.

In general, preparing a source dataset requires a significant amount of time, but it has to be done only once. The time required for one experiment varies between 48 hours and 14 days. During the experiment the object is captured 10 to 20 times. Each scan and texture acquisition takes 1 to 2 hours, including the post-processing (scan merging, image correction, illumination removal, texture mapping, and creation of the final VRML model).

3.2 Data Analysis

3.2.1 General Procedure. Predicting the detailed position where a fleck of rust will form or a spore of mold will settle and grow requires far more detailed microscopic data than we can measure, or that a user would want to specify in a graphics application. On a larger scale, however, the texture sequences which we have observed display systematic variations that depend on physically meaningful and measurable parameters. Rust for example is more likely to form at surface points that were more exposed to the rust causing agent. The appearance change of the objects in our experiments depends on their geometry and on external factors of the experimental setup. In order to transfer the observed texture changes in a physically plausible and visually appealing way, we have to identify these relationships.

For this purpose, we describe the factors which may have a systematic influence by a set of physically meaningful and measurable parameters, which we call “context parameters.” We look to previous work in material simulation (see Table I) to identify common geometric parameters that control change. For each experiment, we describe the geometry by the following set of local features:

Ambient occlusion measures the exposure of a surface point to the surroundings. The surroundings may act as the source of light, oxygen, or dispersed materials that are necessary for the material change process. The surroundings may also act as a sink for material leaving the surface by evaporative processes. The technical definition of the ambient occlusion at a given surface point is the fraction of the surrounding hemisphere that is occluded by other parts of the surface. See [Kontkanen and Laine 2005] for a more detailed discussion of the definition of ambient occlusion and its efficient computation. Also, note that the ambient occlusion measures similar properties as the “accessibility” defined in [Miller 1994].

Source direction indicates the orientation of the surface to the source of an outside agent, with, or without, accounting for cast shadows. This is also relevant to measure the surface orientation relative to the direction of gravity. When cast shadows are accounted for, the source direction is calculated by a simple simulation.

Signed mean curvature is positive on convex and negative on concave surface areas. It can be calculated as the inner product of the mean curvature normal and the normal at every vertex. Please refer to

[Meyer et al. 2003] for the calculation procedures. The curvature of a surface can influence the stress and tear of surface layers such as paint.

Principal directions at every vertex are the directions of minimal and maximal curvature. Please refer to [Meyer et al. 2003] for a detailed definition and calculation procedures. We noted that the principal directions are helpful to describe the alignment of texture features with the geometry. For example, the alignment of paint cracks along curves of constant curvature.

Coating thickness can vary across an object’s surface. We can estimate the coating thickness by registering and comparing shapes before and after a cracking mixture application. In the case of paint, we can estimate the thickness distribution by measuring its transparency using a back-light projector.

The physical background knowledge about the investigated effects as outlined in the taxonomy, and the experience from the experiments, usually provide a good intuition about what the relevant factors are. The “rusting on the statue of Mozart,” for example, was caused by the vinegar vapor to which it was exposed. The surface orientation with respect to the vinegar source and the ambient occlusion are both factors that determine how exposed one point on the surface was to the vapor. Nevertheless, to understand the complex appearance data we capture in the experiments, we use a non-linear dimensionality reduction tool, called “Diffusion Maps” [Coifman et al. 2005], to describe the whole texture time sequence by a small number of parameters. This dimensionality reduction enables us to estimate how much of the variations in the texture time sequence can be explained by an individual context parameter. This can be done by simply calculating the correlation between the context parameter and the texture.

The detailed procedure is as follows: For each vertex of the 3D mesh of the given object, we consider a square patch in the texture map around the vertex. We take the values of the pixels in that patch at every time instance in the texture time sequence and attach them all together, yielding a high-dimensional vector for each vertex. We calculate the diffusion map of this dataset as described in [Coifman et al. 2005]. The diffusion map yields a set of orthogonal functions on the vertices. Depending on the regularity of the texture, 5 to 40 of these functions are sufficient to approximately reconstruct the texture, if it is projected onto the subspace spanned by the functions. (A complete basis would require a number of functions equal to the number of vertices.) The crucial feature of this set of functions is that on vertices with similar texture structure the values of each function are close as well. Furthermore, the orthogonality guarantees that the functions are not correlated among each other.

The context parameters and the texture parameters resulting from the diffusion map are functions on the vertices and can be represented as n -dimensional vectors, where n is the number of vertices. In order to just capture the variations, we subtract the mean and normalize each context parameter vector. We measure the correlation of a specific context parameter with the texture parameters by taking the inner product of the context parameter vector with each of the texture parameter vectors and then summing up the results. Depending on the data size, we had to subsample the vertices or the time instances to make the calculations feasible.

The computation time depends on the number of subsampled vertices, the size of the square patch, and the number of time instances. The main steps in the computation of the diffusion map are a fast nearest-neighbor search on the texture sequence patches, and the calculation of the first eigenvectors of an $n \times n$ matrix, where n is the number of vertices. A typical example is the computation for the rusting Mozart statue: 28,000 subsampled vertices, 5×5 patches, and 9 time instances. The computations were performed in Matlab. On a Pentium 4 with a 2.4 GHz processor and 1.0 GB RAM, they took 41 minutes.

The procedure above enables us to select the context parameters which best correlate with the texture variations, and gauge roughly how good the correspondence is in comparison to other parameters. The precise numerical value of the correlation is not meaningful, since it depends on the number of sample vertices and texture parameters. The relative strength of the correlations of different parameters in comparison to each

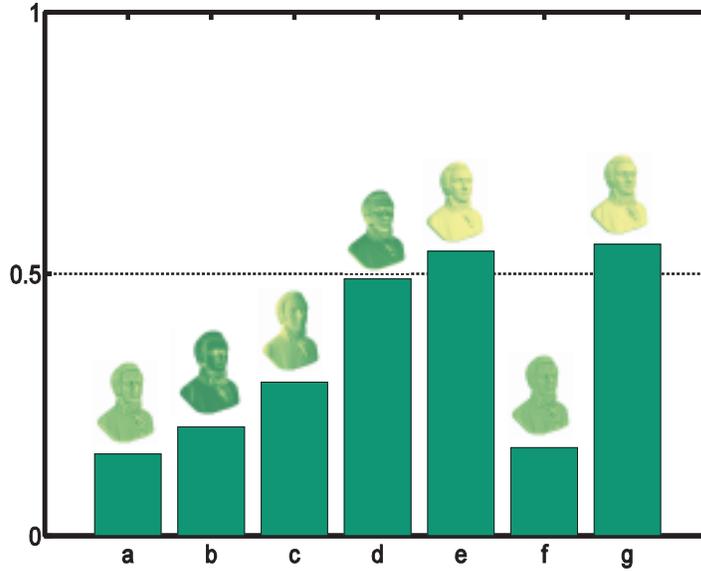


Fig. 3. The correlations between texture parameters and a selection of context parameters for the rusting Mozart statue: (a) Signed Mean Curvature, (b) Ambient Occlusion, (c)-(d) the normal component in different (non-optimal) directions, (e) the normal component in the optimal direction, (f) Signed Mean Curvature \times the optimal direction parameter, (g) The final control parameter: $(1 - \text{ambient occlusion}) \times \text{optimal direction}$.

other is, however, extremely robust under these choices.

For most of our experiments, the correlation between the selected context parameters and texture sequence is pretty obvious and can also be seen just by visual examination. However, in cases where the context parameters are not obvious, this analytical tool is helpful in determining and validating the right parameters. An example is the rusting statue of Mozart, which will be presented below.

We also try combinations of these parameters by taking the per vertex product. The parameters that are finally selected are the control parameters that will guide the texture transfer. The concrete results for each experiment will be detailed below.

3.2.2 Analysis Results

Patination. The patination on the surface of the copper kitchen pan is caused by the chemical solution we sprayed on it. Therefore, the patination at a specific point on the surface depends on the exposure to the solution. The captured texture sequence shows a high correlation with the ambient occlusion, which is obvious from simple inspection. Our correlation calculation confirms that the ambient occlusion dominates all other geometric parameters. A reasonable explanation is that the chemical solution accumulated in the concave indents and took longer to evaporate. Therefore, we observe a stronger patination effect at points with high ambient occlusion. We conclude that ambient occlusion is the right parameter to guide this patination effect.

Rusting. In the rusting experiments, spraying and exposure to vapor lead to very different rusting patterns, the first smooth and uniform and the second spotty and freckled. For the uniform pattern, there is again an obvious correlation with the ambient occlusion. In addition, there is a clearly visible correlation with the surface normal component in the spraying direction. These two factors influence how much of

the chemical solution was deposited at a surface point. We have chosen both as the control parameters for this effect. For the vapor setup, the texture sequence is more complex and the best parameters are not obvious. Calculation of the correlations with the standard set of context parameters shows that the normal component in the direction of the vinegar source plays the dominant role. We choose the optimal direction by picking the direction with the highest correlation from a set that samples the hemisphere at space angular intervals of 0.1 rad. Figure 3 shows the correlations with the usual suspects and with some directions. The correlation with the optimal direction parameter d can be slightly improved by combining it with a measure for the exposure based on the ambient occlusion o to form a new context parameter: $P = d(1 - o)$. Since taking into account the ambient occlusion will also make the texture variations on the target object more interesting, we choose this parameter as the control parameter.

Paint cracking. For the paint cracking example on the flat slide, there are no geometric variations on the source object. The high correlation of the crack pattern with the paint thickness, which was measured in the experiment, can clearly be seen. For thin paint, the cracks are small and the crack density is high. At points with thicker paint, there are fewer but significantly bigger cracks. Therefore, the paint thickness is the right control parameter.

Cracking Paste. As for the previous example, the crack size and density were directly related to the paint thickness. Since we applied the cracking mixture to a 3D shape—a wave form—we could also observe the influence of the geometric variations. The cracks first started to develop in the convex regions of the surface and the crack density in these regions was higher than in the concave regions. The correlating geometric parameter for this effect is the signed mean curvature. Furthermore, we observed that the cracks tend to align with the geometric features of the object; the cracks either propagate in the direction of the biggest curvature or in the direction orthogonal to it, as described by the field of principal directions. To transfer the alignment effect, we have conducted our final experiment on a shape with a constant principal directions field (the wave form). On the target object, for the cracking, we use the direction field of maximum curvature to guide the synthesis. Summarizing, there are three control factors for the paste cracking: Alignment with the principal directions and the regular parameters: paint thickness and signed mean curvature.

Dirt Accumulation. For the texture sequence of this example, we have observed a high correlation with the ambient occlusion and the normal component in the direction of gravity. The fine dirt, which was deposited on the convex regions of our test object, tended to slide into nearby concavities and would not stick on surface areas which were too steep. The final parameter is the ambient occlusion combined with the direction of gravity. Note, that there can be other dirt accumulation scenarios, where this relationship does not hold, or is even reversed. For example, if the surface of the test object is very sticky, the dirt will first accumulate on the more exposed surface areas.

Biological Mold Growth. The mold started to grow and grew most exuberantly in the regions that were less accessible. We believe that this is due to the fact that these regions were less exposed to airflow and therefore more moist. The context parameter for this effect is the ambient occlusion.

3.3 Data Transfer via Texture Synthesis

In this section, we describe the algorithm on how we transfer appearance data from input to output based on context information. Since the appearance data consists of primarily repetitive texture patterns, we utilize texture synthesis techniques for data transfer.

The input of our system consists of a 3D spatiotemporal texture recording the time-varying appearance history of certain materials plus the associated context parameters, and an output mesh with assigned context information. Our goal is to synthesize a new spatiotemporal texture onto a target mesh so that it conforms to the assigned context information, while maintaining the input texture patterns. We would

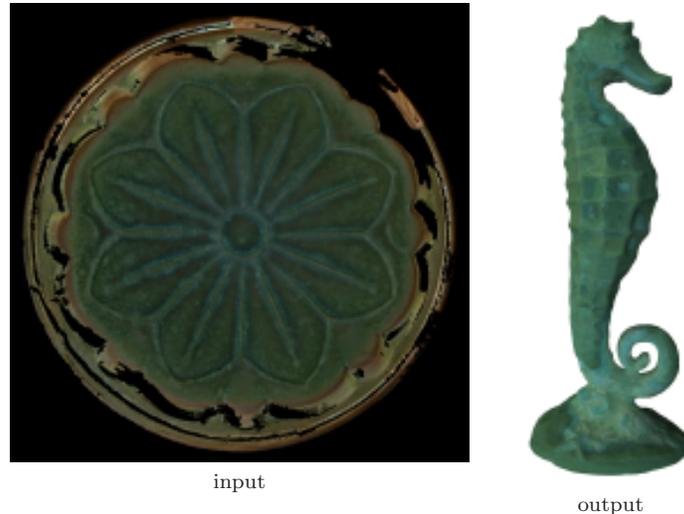


Fig. 4. The input data and output model can have very different local geometric, or contextual information, such as the round copper shape and the seahorse models shown here.

also like our algorithm to be efficient enough to allow direct user interactions with the output model. In addition, since the user might interact with any portion of the model, we would like the synthesis result to be order-independent [Wei and Levoy 2002] to maintain pattern consistency.

To meet these requirements, we have chosen [Turk 2001; Wei and Levoy 2001] as our base algorithm with modifications. We chose this pixel-based algorithm over patch-based ones [Soler et al. 2002; Kwatra et al. 2003; Zhou et al. 2005] since these algorithms tend to gloss over local geometry, especially when the input data and the output mesh have very different local geometric details (Figure 4). To avoid this problem, we have to utilize small patches, but this would diminish the benefit of patch-based algorithms in terms of quality and speed compared to pixel-based ones [Turk 2001; Wei and Levoy 2001; Tong et al. 2002; Zhang et al. 2003].

For clarity, we first provide a brief review of [Turk 2001; Wei and Levoy 2001], followed by the necessary modifications for our application. Basically, [Turk 2001; Wei and Levoy 2001] synthesizes textures over arbitrary 3D mesh surfaces via vertex coloring. As a preprocess, they first build multi-resolution pyramids for both the input image and the output mesh. The synthesis then progresses from lower to higher resolutions of the output mesh. At each resolution, the output vertices are assigned colors one by one via neighborhood search and comparison; in particular, [Turk 2001; Wei and Levoy 2001] provide a mechanism for sample output mesh neighborhoods, so that they can be compared with input pixel neighborhoods via a simple ℓ^2 -norm.

We now describe the necessary modifications over [Turk 2001; Wei and Levoy 2001] to satisfy our specific application needs. Note that all of these modifications come from previous work; we simply combine them in a novel way. For easy reference, we have summarized our algorithm in Table III.

High dimensional pixels. Unlike [Wei and Levoy 2001], which deals with RGB color textures, our datasets are 3D spatiotemporal volumes, which means we treat each pixel as a huge vector containing concatenated RGB values across all time frames at the same spatial pixel location. Due to this high dimensionality, we opt for a CPU, rather than a GPU implementation, to avoid the texture memory limit problem.

```

function Preprocess
   $I_a \leftarrow \text{PCA}(I_a)$ ;
   $G_a \leftarrow \text{BuildImagePyramid}(I_a)$ ;
   $G_a \leftarrow \text{K-CoherenceAnalysis}(G_a)$ ;
   $G_s \leftarrow \text{BuildMeshPyramid}(M_s)$ ;
  RetimeMeshes( $G_s$ );
  AssignTextureOrientation( $G_s$ );

function  $G_s \leftarrow \text{ContextAwareTextureSynthesis}(G_a, G_s)$ 
  foreach level  $L$  from lower to higher resolutions of  $G_s$ 
    foreach iteration from 0 to  $N$ 
      //  $N == 2$  in our implementation
      loop through all pixels  $p$  of  $G_s(L)$ 
         $C \leftarrow \text{FindBestMatch}(G_a, G_s, L, p)$ ;
         $G'_s(L, p) \leftarrow C$ ;
       $G_s \leftarrow G'_s$ ;
      //  $G'_s$  double-buffers of  $G_s$  for order-independent synthesis
  return  $G_s$ ;

function  $C \leftarrow \text{FindBestMatch}(G_a, G_s, L, p)$ 
   $N_s \leftarrow \text{BuildMeshNeighborhood}(G_s, L, p)$ ;
  // via [Wei and Levoy 2001]
   $N_a^{best} \leftarrow \text{null}$ ;  $C \leftarrow \text{null}$ ;
  loop through all pixels  $p_i \in \text{KCoherentSet}(G_s, L, p)$ 
    //  $k$ -coherence search via [Tong et al. 2002]
     $N_a \leftarrow \text{BuildImageNeighborhood}(G_a, L, p_i)$ ;
    if  $\text{Match}(N_a, N_s) > \text{Match}(N_a^{best}, N_s)$ 
       $N_a^{best} \leftarrow N_a$ ;  $C \leftarrow G_a(L, p_i)$ ;
    // context information serves as constraints
    // similar to [Hertzmann et al. 2001]
  return  $C$ ;

```

Table III. Pseudocode of our algorithm. Our context information serves as constraint during neighborhood matching as shown in function **FindBestMatch**. Please refer to Table IV for the meaning of symbols.

Symbol	Meaning
I_a	Input spatial-temporal image
M_s	Output spatial-temporal mesh
G_a	Gaussian pyramid built from I_a
G_s	Gaussian pyramid built from M_s
p_i	An input pixel in G_a
p	An output vertex in G_s
$P_s(p)$	Flattened patches around p
$N(p)$	Neighborhood around the pixel p
$G(L)$	L th level of pyramid G
$G(L, p)$	Pixel p at level $G(L)$

Table IV. Table of symbols

However, for large input datasets, their total data size may still exceed CPU system memory. In this case, we perform SVD (singular value decomposition) to reduce the dimensionality of the input data size as in [Liu et al. 2001]. We have found our CPU implementation provides interactive speed for typical user interactions.

Constrained synthesis. In addition to RGB colors, our texture pixels also contain context information, serving as a constraint during synthesis, similar to the methodology in [Hertzmann et al. 2001]. This is to ensure that the synthesized texture over a target mesh model conforms to the given context. Furthermore, for textures with distinctive patterns, such as paint cracks, we also utilize texton masks [Zhang et al. 2003] to improve the synthesis quality. In this scenario, both context information and texton masks serve as synthesis constraints.

Order-independent synthesis. In our GUI system, we would like to allow the user to be able to change context parameters and perform re-synthesis over any arbitrary sub-regions of the output mesh. In order to maintain an invariant result with respect to all possible sub-regions, we adopt order-independent synthesis [Wei and Levoy 2002; Lefebvre and Hoppe 2005]. This can be easily achieved by using two copies of the output vertex colors and perform synthesis in a “ping-pong” fashion. In addition, by using k -coherence search [Tong et al. 2002], as in [Lefebvre and Hoppe 2005], we are able to achieve faster synthesis than [Turk 2001; Wei and Levoy 2001].

4. USER INTERFACE

An important part of the whole pipeline is to provide some graphical user interaction to help end users better utilize our techniques. Among previous approaches for aging effect synthesis, 3D drawing requires tedious user input for all details, while conventional texture transfer methods do not provide any direct control, or interactive-time visual feedback, to the end users. We design the following user interface with an attempt to find a good balance between “no-control-at-all” and “more-than-enough-control”; the goal is to provide high-level, intuitive control for the end users. The users can synthesize the aging effects by modifying the context, changing the time-axis, and combining two different effects in a cascading way.

Below, we provide more details about our GUI system:

```

function AttributePainting( $G_a, G_s$ )
  assign user painted context parameters into  $G_s$ 
  foreach level  $L$  from lower to higher resolutions of  $G_s$ 
    foreach iteration from 0 to  $N$ 
      //  $N == 2$  in our implementation
      loop through pixels  $p$  of  $G_s(L)$  where
         $N(p)$  touched by user painting
         $C \leftarrow \text{FindBestMatch}(G_a, G_s, L, p)$ ;
         $G'_s(L, p) \leftarrow C$ ;
       $G_s \leftarrow G'_s$ ;

function CascadeSynthesis( $G_a^1, G_a^2, G_s$ )
  //  $G_a^1$  are  $G_a^2$  two inputs for different phenomena
   $G_s \leftarrow \text{ContextAwareTextureSynthesis}(G_a^1, G_s)$ 
  // for effect 1
  transform new color result into context parameters for  $G_s$ 
   $G_s \leftarrow \text{ContextAwareTextureSynthesis}(G_a^2, G_s)$ 
  // for effect 2

```

Table V. Pseudocode of our algorithm for GUI/user-interaction.

Context parameter painting (Figure 5). The user can manipulate a virtual brush to fine-tune the context parameters interactively. Once the user finishes editing, our tool will automatically perform texture

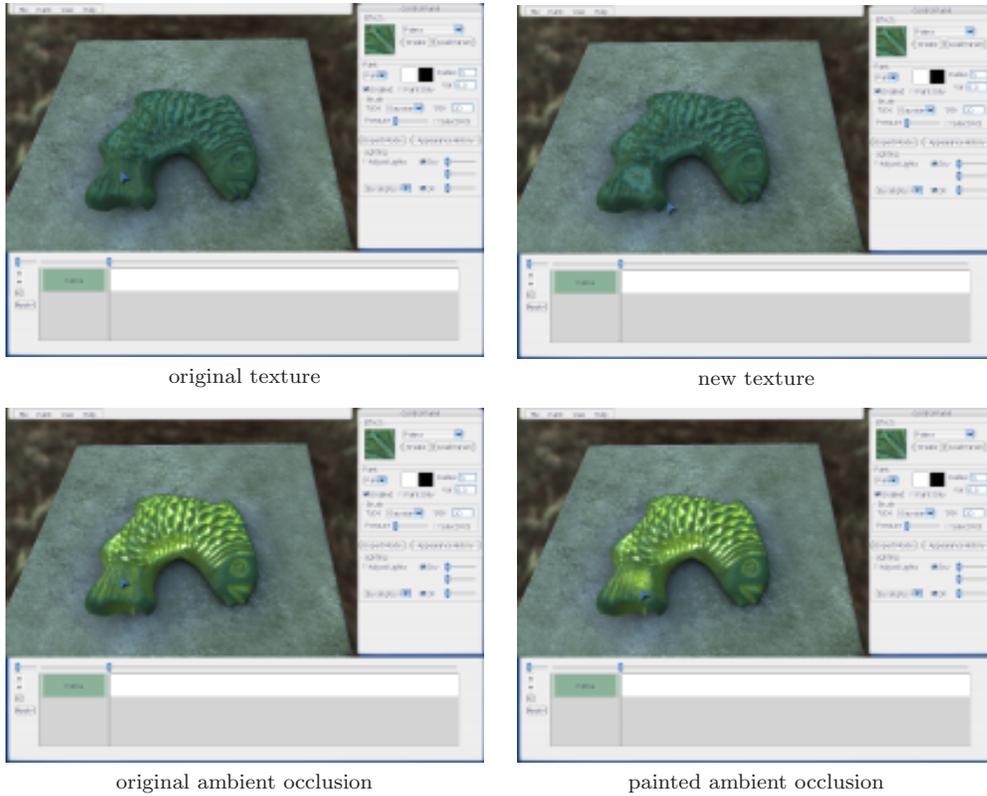


Fig. 5. Attribute painting. The user paints new ambient occlusion information, and the new patination is then automatically synthesized at an interactive speed.

synthesis over the user-painted region to give real-time feedback. Our current system supports various brush and sprayer effects similar to Adobe Photoshop. See function **AttributePainting** in Table V for details.

Cascade synthesis (Figure 6). In addition to synthesizing one effect, our system also allows cascade synthesis, where a later effect may grow depending on an earlier one, as shown in Figure 6. In this example, we first synthesize a paint-peeling pattern based on the geometry curvature as the context parameter; we then synthesize a rusting pattern over the paint-peeled regions by considering the synthesized paint-peeling pattern as the “ambient occlusion” context parameter. In our current system, we implement the cascade effect as follows. First, we synthesize an earlier effect via **ContextAwareTextureSynthesis**. We then transform the newly synthesized color values into context parameters for the next effect; for example, in the paint-peeling followed by rusting, we transform the degree of color darkness caused by peeling into the ambient occlusion context parameter for the subsequent rusting process. Note that since these two effects might not use the same context information, we have to transform the color results generated by an earlier effect into the context parameters for the later effect; obviously, this is a heuristic approach, rather than physically plausible, but it allows us to expand our method to novel effects. See **CascadeSynthesis** in Table V.

Appearance History Fine-Tuning (Figure 7). In some cases, it is helpful for users to control the progression

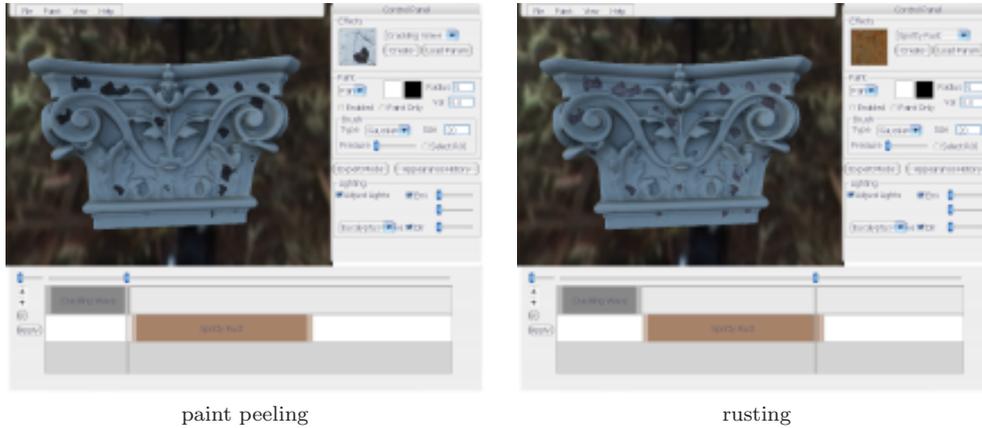


Fig. 6. Cascade synthesis effect. We first synthesize the paint peeling pattern, and utilize this as the context information for the subsequent iron rusting.

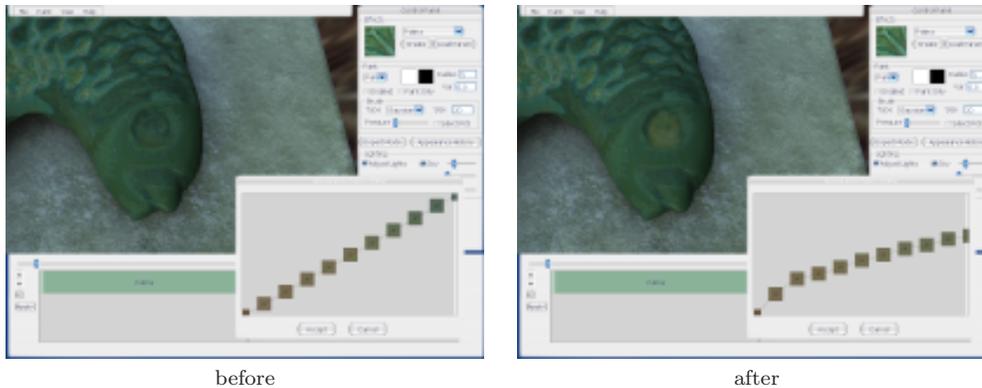


Fig. 7. Appearance history tuning. Here, the user selects a small region over the fish eye, and adjusts the appearance history diagram so that textures on the eye progress differently from the rest of the shape.

speed of aging effects. Our tool also allows local fine-tuning of the appearance history over a certain region of the entire mesh. To be precise, the user may first select a region of interest on the mesh, change the curve of appearance history of this region and then our tool will re-compute the appearance according to this modified history curve and immediately provide visual feedback. Note that this fine tuning does not require any re-synthesis, as we only need to determine which time instance to show for each output vertex sample. We perform linear blending between adjacent time samples for smooth transition.

5. RESULTS

To validate our approach, we present the source data we captured and processed from the experiments we conducted, and the temporal and spatial appearance variations transferred to new objects. All demos in this section are rendered in Cinema4D (except Figure 18). Background and tabletop in the figures are rendered with simple texture mapping, with texture samples from <http://www.mayang.com/textures/>.

Most of the previous papers in weathering effects synthesis only presented final renderings, possibly next

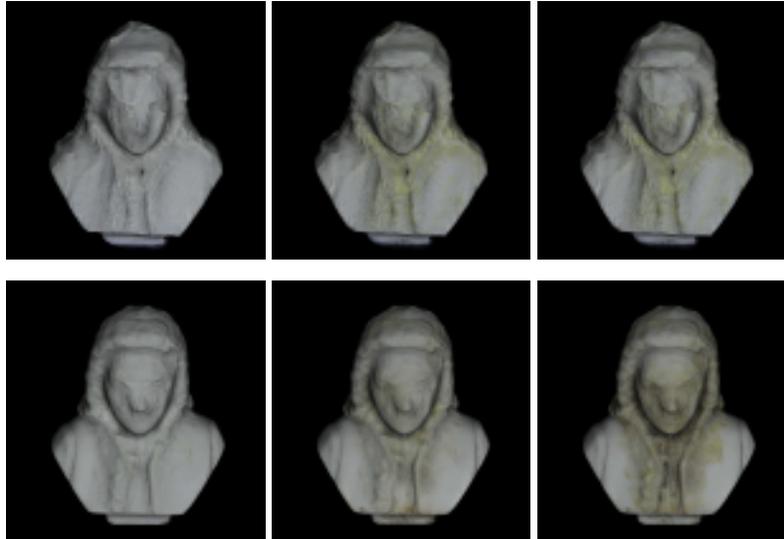


Fig. 8. Comparison between two experiments of biological mold growing on a statue of Bach. Top row: three instances captured during experiment 1; Bottom row: corresponding instances captured during experiment 2. The context parameter is the ambient occlusion, which favors the concavities.

to an image from the real world, but did not provide much detailed discussion about the overall comparison. In this section, we discuss several experiments to validate the concept of “context-aware textures.” In these experiments, we show for each example: the natural variation in real-world aged appearance; the appearance history transferred back to the source objects; comparison between appearance transfers with and without context constraints; comparison between synthesized results and ground truth; and, finally, a wide range of effects transferred to new target objects.

5.1 Comparison between Two Experiments of the Same Aging Effect

In order to illustrate the natural variation for the same aging effect, we conducted the same aging experiment twice on the same object and with the same context parameter values. Fig. 8 shows the comparison between the appearance of biological mold growing on cream cheese, which was applied on a statue of Bach. Although the appearances of corresponding instances are not pixel-by-pixel identical, the biological mold seems to settle down first in the grooves, and grow from these seeds. Accordingly, the color of the regions occupied by the biological mold changes from milky white to yellow, and gets darker and darker as more and more biological mold grows up. From this example, the aged appearance could be roughly predicted by ambient occlusion, the context parameter in this specific effect. Also, we will expect to see some temporal or spatial variation due to natural randomness. The combination of predictability and randomness is the basis for our main concept of “context-aware texture.”

5.2 Aging Effects Transferred Back to Source Objects

Another way to see how well the appearance history is directed by some context parameters is to transfer the captured data back to the source objects and study the appearance difference. Fig. 9 shows one instance of results transferred back to the source objects for two different aging effects. We can see that the dominant appearance changes—color change in the patina example, occurrence of rusting spots in the rusting



Fig. 9. Aging effects transferred back to source objects, as unlit textures rendered on the shapes. Top row: patina on kitchen pan, with context of ambient occlusion; (from left to right: the source appearance, the appearance transferred back to the same copper object, and the visualized context parameter); bottom row: iron rusting on a statue of Mozart with contexts of ambient occlusion and source direction. (from left to right: the source appearance, the appearance transferred back to the same statue, and the visualized context parameters).



Fig. 10. Patination synthesis result with and without complying with the ambient occlusion context information. Notice the unnatural spot patterns on the left image.

example—have been largely captured and faithfully copied back to the source objects. Such appearance changes also conform to the underlying context parameter values. The appearance of the synthesized results is not identical to the original appearance. There are several reasons. First, minor temporal or spatial differences come from natural variation in aging effects as shown in Fig. 8 in Section 5.1. Second, dimensionality reduction during texture synthesis introduces some blurriness. Finally, imperfect correlation between context and appearance patterns can result in regions of difference, such as the rust patterns on the chest of the Mozart statue. However, our system captures the important features of weathering appearance with respect to the context.



Fig. 11. Comparison between synthesized results and ground truth of rusting on black iron pipes. Left: rusting on source T-junction pipe; Middle: transferred result on an elbow-shaped pipe; Right: elbow-shaped pipe undergoing the same rusting process. Contexts are ambient occlusion and source direction.

5.3 Appearance Transfer with and without Context Constraints

One important question for our system is, does the use of context control parameters have a noticeable effect on the resulting transferred texture? To answer this question, we transfer textures from the original copper kitchen pan to a fish shape. (Please refer to Fig. 12 for the original appearance of the source object.) Images in Fig. 10a and b were generated with and without the context parameter. Without the use of the context parameter, the texture is relatively uniform, with a couple of spots of lighter texture that are uncorrelated to geometry. Using the context parameter, a clearly different result is obtained. Areas in the ridges between the fish scales are clearly lighter than the tips of the scales.

5.4 Comparison Between Synthesized Results and Ground Truth

Another question may be, to what extent are the transferred results predictive of the appearance of an object undergoing the same physical process? To answer this question, we performed a physical, ground-truth experiment. The fish shape, made of copper, was treated with the same solution as the original round kitchen shape and under the same conditions. A rendering of the shape with the captured texture from the end of the experiment is shown in Fig. 10c. The result is not pixel-by-pixel identical to our simulated result in Fig. 10b. However, the general trends of lightness in the ridges and darker scales appear, rather than the more uniform texture obtained without the context parameter.

One more example is provided in Fig. 11, showing iron rusting effects transferred from one pipe to another pipe of different shape. As we expected, rusting patterns appear differently in concavities and convexities, as illustrated in the ground truth. (See Fig. 14 for the full rusting pipe series of the source and transferred results.)

5.5 Aging Effects on New Target Objects

In this section, we show different aging effects transferred to different targets, demonstrating the applicability of our system to a wide range of target object geometries. For better comparison, appearance histories from source objects are illuminated and placed on the left columns.

Fig. 12 shows patina development on copper objects. The source object is a copper kitchen pan, and the targets are a copper seahorse and a copper fairy statue. The context parameter is ambient occlusion. From the figures, we can see that the global color changes at different time instances are captured and faithfully transferred. Moreover, from the closeup figures, we can also see patina patterns appearing sooner and faster in the concavities, which conform to the correlation between the appearance and shape we captured in the source data.

Fig. 13 shows rusting effects—denoted as “rusting (b)” in Table (II)—transferred to two target objects:



Fig. 12. Patina developing on copper objects, with ambient occlusion as the context. Left column: original appearance history from the copper kitchen shape; Center column: patina transferred to the seahorse; Right column: patina transferred to the fairy statue.

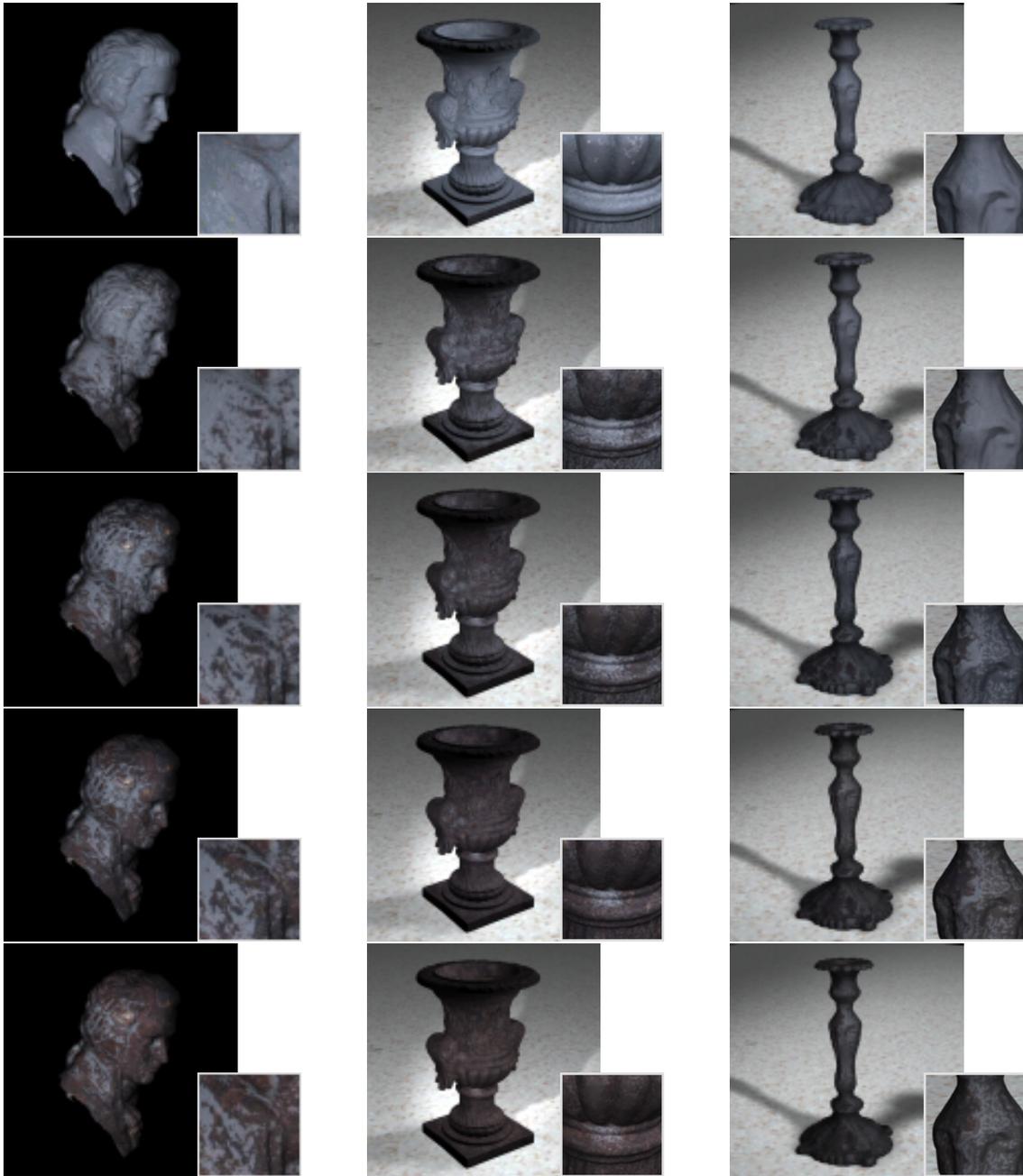


Fig. 13. Rusting on iron surface in vinegar vapor, with ambient occlusion and source direction as the control parameters. Left column: original appearance history from the Mozart statue; Center column: a rusting urn, with omni-directional vinegar source; Right column: a rusting candlestick, with the vinegar source on its left.

an urn and a candlestick. The source object for this effect was a Mozart statue, which was covered with iron surfacer, a commercial product, and exposed to vinegar vapor from below his nose. The context parameters are the ambient occlusion and the source direction. Therefore, we can see the rust develops first on the more exposed regions, with the frontal side being easier to rust. For the target objects, the direction from which the rust causing agent is emanating can be chosen by the user. For the urn in the center column, we have chosen a uniform exposure from all directions, such that the rusting effect on the target object depends only on the ambient occlusion. For the candlestick in the right column, the vinegar source direction is from the left, as explicitly specified by the user. We can see the left half of the candlestick gets rusty sooner and faster. The flexibility of the source direction in these cases provides the user with a very convenient and intuitive way to influence the transferred effects. While the control parameters for the source objects are fixed, the user can choose different settings for the target objects. Since our control parameters have a physical meaning, they provide an intuitive and natural tool to manipulate the effect.

Fig. 14 shows two other rusting examples. For the one denoted as “rusting (a)” in Table (II), the source object was a “ripple” pattern, which was covered with iron surfacer and sprayed with a chemical solution. For the target iron pot, no specific direction of the emanating rusting agent was specified, so that the effect is only controlled by the ambient occlusion. For another example, denoted as “rusting (c)” in Table (II), the source object is a black iron pipe enclosed in a bucket filled with vinegar vapor. The appearance history depends on the ambient occlusion and the source direction of vinegar vapor. Compared to Fig. 13, we demonstrate nicely how the same chemical effect can lead to very different visual impressions.

Fig. 15 shows results for biological mold growing on cheese. The source object is a statue of Bach covered with cream cheese. Targets include a string cheese and a piece of Swiss cheese. Mold tends to develop from the concavities and change the appearance to furry dark, which can be viewed as directed by ambient occlusion. In general, more complex control factors are involved in the biological world, which makes it even more difficult to simulate the aging effects cause by biological growth. As shown in Table (I), not much work has been done to model and synthesize—not only to render—those weathering effects. This example illustrates that, under certain conditions, such complex phenomena can be modeled and transferred to arbitrary shapes based on one simple geometric feature, which derives its power from our basic concept of “context-aware texture” and the pipeline we built.

Fig. 16 shows the paint cracking effect, a mechanical process. The source object was a flat slide, and the targets are a frog and a lion-shaped “sippy” cup. The size of the cracks varies with the paint thickness, which has been painted on the target objects with our GUI. Unlike previous examples, paint thickness, used as a context parameter here, does not directly depend on the shape. We rendered the paint thickness maps—with brighter color representing thicker paint—on the shapes from the same viewing angle, and put them in the bottom row of the figure for better visualization. Directly computing the neighborhood vectors of all pixels in the original texture image, even after SVD decomposition, would make excessive demands on memory. Therefore, we use a low-spatial-resolution version of the paint crack image shown in the left column of Fig. 16 for synthesis in addition to the dimensionality reduction of the input data as indicated in Section 3.3. The blurriness from source texture resampling and dimensionality reduction inevitably degrades the texture synthesis quality, especially for crackling patterns.

Fig. 17 shows two more aging effects due to mechanical processes. Dirt accumulation is transferred based on the contexts of ambient occlusion and source direction. Although some of the dirt patterns have been transferred from the Bach statue to the corresponding parts of Chopin, the overall appearance history seems not quite satisfactory. This may be due to the global transport of dirt that cannot be fully accounted for by local geometric features. For the paste cracking and peeling effect, some of the cracks do follow the grooves, which have been observed in the source wave pattern. One of the possible reasons is that crack patterns aligned with shape curvature are highly anisotropic. We need to use a much larger neighborhood in



Fig. 14. Two more rusting effects. Two columns on the left: rusting caused by spraying a chemical solution on iron surfacer is transferred from the “ripple” pattern to the iron pot; Two columns on the right: rusting of black iron in vinegar vapor is transferred from the T-junction pipe to the elbow-shaped pipe.



Fig. 15. Biological mold growing on cream cheese, with ambient occlusion as context. Left column: original appearance history from a statue of Bach; Center column: biological mold growing on a string cheese; Right column: biological mold growing on a piece of Swiss cheese.

effect	target	in figure	triangles	offline (min.)	online (min.)
patina	round copper shape	9	238,381	30	10
rust(b) †	Mozart	9	217,484	11	7
patina	fish shape	10	377,057	10	6
patina	seahorse	12	365,421	7	5
patina	fairy	12	478,004	25	5
rust(b) †	urn	13	280,281	8	6
rust(b) †	candlestick	13	334,736	20	12
rust(a) †	iron pot	14	211,018	7	5
rust(c) †	elbow-shaped pipe	14	67,176	6	2
biological mold	string cheese	15	123,530	9	5
biological mold	swiss cheese	15	67,698	4	3
paint cracking	frog	16	87,732	30	26
paint cracking	“sippy lion”	16	511,232	30	40
dirt accumulation	Chopin	17	794,135	5	4
cracking paste	granite pot	17	589,382	16	13
paint crack & rust(b)	iron pot	18	211,018	40	47
peeling & rust(b)	capital	18	303,558	21	41

Table VI. Table for texture transfer timings. Offline time consists of the SVD computation, the k -coherence building process, and the mesh neighborhood finding process; online time is the texture synthesis process itself. (†See Table (II) for difference between these rusting effects.)

the texture synthesis algorithm to capture and preserve such visual features, which requires not only more computational resources, but also much more source data to be captured.

5.6 Cascade Effects

In Figure 18, we show results of the cascaded synthesis: two different cracking and peeling effects, followed by rusting of the underlying iron. The source object for the column capital was the wave form shown in Fig. 16, which was covered by the cornstarch mixture. The thickness of the cracking layer on the target object is constant. The control parameters were the signed mean curvature and the principal direction. We used the field of principal directions on the target object to guide the synthesis. The source object for the iron pot was the flat slide shown in Fig. 16. The golden color behind the cracks was replaced by iron on which we could then apply the rusting effect. These examples demonstrate how our approach offers new possibilities to combine different effects on the same object within one consistent framework.

5.7 Running Time Compared with Previous Approaches

The texture synthesis is the only part that has to be performed for each new target object and it significantly outperforms first-principles simulations and digital painting, as noted in Table (I).

The timing for the texture synthesis algorithm consists of an offline part, which itself consists of the SVD computation, the k -coherence building process, and the mesh neighborhood finding process, and an online part, which is the texture synthesis process itself. Note that each synthesized texture includes all time steps. All computations were performed on a 3.2G Pentium 4 PC with 1GB RAM, and the running times are summarized in Table (VI). For the interactive local adjustment, typically less than 1 second is needed for the result at the coarsest level (to provide instant visual feedback), while 3 to 10 seconds are needed for the

result at the finest level. These speeds are for all effects, except paint cracking, which requires more time to process due to the large neighborhood needed to preserve the shape of cracks.

6. CONCLUSIONS

We have presented a method for capturing progressively varying textures that are formed by chemical, mechanical and biological processes. The unique feature of our framework is that we capture and identify parameters that control the texture variation on the material. Using the control parameters, we are able to generate spatially complex textures that evolve with time on different synthetic objects. While the transfer of texture from our experimental objects to new objects can be fully automatic, we have also presented a graphical user interface that allows a user to apply effects and alter their progression to achieve a particular desired appearance.

We are continuing this work by capturing more effects over longer time periods. Our goal is to produce an extended library of textures with control parameters that can be used in computer graphics modeling systems.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0528204. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- AOKI, K., DONG, N. H., KANEKO, T., AND KURIYAMA, S. 2002. Physically-based simulation of cracks on drying 3d solid. In *10th Pacific Graphics Conference on Computer Graphics and Applications*. IEEE, Beijing China, 467–468.
- BERNARDINI, F., MARTIN, I. M., AND RUSHMEIER, H. E. 2001. High-quality texture reconstruction from multiple scans. *IEEE Trans. Vis. Comput. Graph.* 7, 4, 318–332.
- BERNARDINI, F. AND RUSHMEIER, H. 2002. The 3D model acquisition pipeline. *Computer Graphics Forum* 21, 2, 149–149.
- BLAIS, F. 2004. Review of 20 years of range sensor development. *Journal of Electronic Imaging* 13, 1, 231–240.
- BONET, J. S. D. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Computer Graphics*. ACM SIGGRAPH, 361–368.
- BOSCH, C., PUEYO, X., MÉRILLOU, S., AND GHAZANFARPOUR, D. 2004. A physically-based model for rendering realistic scratches. *Comput. Graph. Forum* 23, 3, 361–370.
- CALLIERI, M., CIGNONI, P., AND PINGI, P. 2002. An end-to-end software suite for 3d scanning. *ERCIM News* 50.
- CHANG, Y.-X. AND SHIH, Z.-C. 2000. Physically-based patination for underground objects. *Computer Graphics Forum* 19, 3 (August). ISSN 1067-7055.
- CHANG, Y.-X. AND SHIH, Z.-C. 2003. The synthesis of rust in seawater. *The Visual Computer* 19, 1, 50–66.
- COIFMAN, R., LAFON, S., LEE, A., MAGGIONI, M., NADLER, B., WARNER, F., AND ZUCKER, S. 2005. Geometric diffusions as a tool for harmonic analysis and structure definition of data, part i: Diffusion maps. *Proceedings of the National Academy of Sciences* 102, 21, 7426–7431.
- DANA, K. J., VAN GINNEKEN, B., NAYAR, S. K., AND KOENDERINK, J. J. 1999. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics* 18, 1 (Jan.), 1–34.
- DEMERS, O. 2001. *Digital Texturing and Painting*. New Riders Publishing, Indianapolis, IN.
- DESBENOIT, B., GALIN, E., AND AKKOCHE, S. 2004. Simulating and modeling lichen growth. *Computer Graphics Forum* 23, 3, 341–350.
- DORSEY, J., EDELMAN, A., JENSEN, H. W., LEGAKIS, J., AND PEDERSEN, H. K. 1999. Modeling and rendering of weathered stone. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 225–234.
- DORSEY, J. AND HANRAHAN, P. 1996. Modeling and rendering of metallic patinas. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM Press, New York, NY, USA, 387–396.
- DORSEY, J., PEDERSEN, H. K., AND HANRAHAN, P. 1996. Flow and changes in appearance. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM Press, New York, NY, USA, 411–420.

- EFROS, A. A. AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*. IEEE Computer Society, Washington, DC, USA, 1033.
- FAROUK, M., EL-RIFAI, I., EL-TAYAR, S., EL-SHISHINY, H., HOSNY, M., EL-RAYES, M., GOMES, J., GIORDANO, F., RUSHMEIER, H. E., BERNARDINI, F., AND MAGERLEIN, K. A. 2003. Scanning and processing 3d objects for web display. In *4th International Conference on 3D Digital Imaging and Modeling (3DIM 2003)*. 310–317.
- GOBRON, S. AND CHIBA, N. 2001a. Crack pattern simulation based on 3d surface cellular automata. *The Visual Computer* 17, 5, 287–309.
- GOBRON, S. AND CHIBA, N. 2001b. Simulation of peeling using 3d-surface cellular automata. In *9th Pacific Graphics Conference on Computer Graphics and Applications*. IEEE, Tokyo Japan, 338–347.
- GORLA, G., INTERRANTE, V., AND SAPIRO, G. 2003. Texture synthesis for 3D shape representation. *IEEE Transactions on Visualization and Computer Graphics* 9, 4 (Dec.), 512–524.
- HEEGER, D. J. AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *SIGGRAPH*. 229–238.
- HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 327–340.
- HIROTA, K., TANOUÉ, Y., AND KANEKO, T. 1998. Generation of crack patterns with a physical model. *The Visual Computer* 14, 3, 126–137.
- HIROTA, K., TANOUÉ, Y., AND KANEKO, T. 2000. Simulation of three-dimensional cracks. *The Visual Computer* 16, 7, 371–378.
- HSU, S.-C. AND WONG, T.-T. 1995. Simulating dust accumulation. *IEEE Comput. Graph. Appl.* 15, 1, 18–22.
- KONTKANEN, J. AND LAINE, S. 2005. Ambient occlusion fields. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*. ACM Press, New York, NY, USA, 41–48.
- KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics* 22, 3 (July), 277–286.
- LEFEBVRE, S. AND HOPPE, H. 2005. Parallel controllable texture synthesis. *ACM Trans. Graph.* 24, 3, 777–786.
- LENSCH, H. P. A., KAUTZ, J., GOESELE, M., HEIDRICH, W., AND SEIDEL, H.-P. 2003. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics* 22, 2 (Apr.), 234–257.
- LIU, X., YU, Y., AND SHUM, H.-Y. 2001. Synthesizing bidirectional texture functions for real-world surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 97–106.
- LU, J., GEORGIHADES, A. S., RUSHMEIER, H., DORSEY, J., AND XU, C. 2005. Synthesis of material drying history: Phenomenon modeling, transferring and rendering. In *proceedings of Eurographics Workshop on Natural Phenomena*. 7–16.
- MARSHNER, S. R., WESTIN, S. H., LAFORTUNE, E. P. F., TORRANCE, K. E., AND GREENBERG, D. P. 1999. Image-based brdf measurement including human skin. In *10th Eurographics Workshop on Rendering*. 139–152.
- MATUSIK, W., PFISTER, H., BRAND, M., AND McMILLAN, L. 2003. A data-driven reflectance model. *ACM Trans. Graph.* 22, 3, 759–769.
- MÉRILLOU, S., DISCHLER, J.-M., AND GHAZANFARPOUR, D. 2001a. Corrosion: Simulating and rendering. In *Proceedings of Graphics Interface 2001*, B. Watson and J. W. Buchanan, Eds. 167–174.
- MÉRILLOU, S., DISCHLER, J.-M., AND GHAZANFARPOUR, D. 2001b. Surface scratches: measuring, modeling and rendering. *The Visual Computer* 17, 1, 30–45.
- MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, Heidelberg, 35–57.
- MILLER, G. 1994. Efficient algorithms for local and global accessibility shading. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM Press, 319–326.
- MUELLER, G., MESETH, J., SATTLER, M., SARLETTE, R., AND KLEIN, R. 2005. Acquisition, synthesis, and rendering of bidirectional texture functions. *Computer Graphics Forum* 24, 1, 83–109.
- PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 2001. Surface aging by impacts. In *Graphics Interface 2001*. 175–182.
- PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 2002. The simulation of paint cracking and peeling. In *Graphics Interface 2002*. 59–68.
- SHORLIN, K. A., DE BRUYN, J. R., GRAHAM, M., AND MORRIS, S. W. 2000. Development and geometry of isotropic and directional shrinkage-crack patterns. *Physical Review E* 61, 6 (june), 6950–6957.
- SOLER, C., CANI, M.-P., AND ANGELIDIS, A. 2002. Hierarchical pattern mapping. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 673–680.

- TONG, X., ZHANG, J., LIU, L., WANG, X., GUO, B., AND SHUM, H.-Y. 2002. Synthesis of bidirectional texture functions on arbitrary surfaces. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 665–672.
- TURK, G. 2001. Texture synthesis on surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 347–354.
- WEI, L.-Y. AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 479–488.
- WEI, L.-Y. AND LEVOY, M. 2001. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 355–360.
- WEI, L.-Y. AND LEVOY, M. 2002. Order-independent texture synthesis. Tech. Rep. TR-2002-01, Computer Science Department, Stanford University.
- WONG, T.-T., NG, W.-Y., AND HENG, P.-A. 1997. A geometry dependent texture generation framework for simulating surface imperfections. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97*. Springer-Verlag, London, UK, 139–150.
- ZHANG, J., ZHOU, K., VELHO, L., GUO, B., AND SHUM, H.-Y. 2003. Synthesis of progressively-variant textures on arbitrary surfaces. *ACM Trans. Graph.* 22, 3, 295–302.
- ZHOU, K., DU, P., WANG, L., MATSUSHITA, Y., SHI, J., GUO, B., AND SHUM, H.-Y. 2005. Decorating surfaces with bidirectional texture functions. *IEEE Trans. Vis. Comput. Graph.* 11, 5, 519–528.

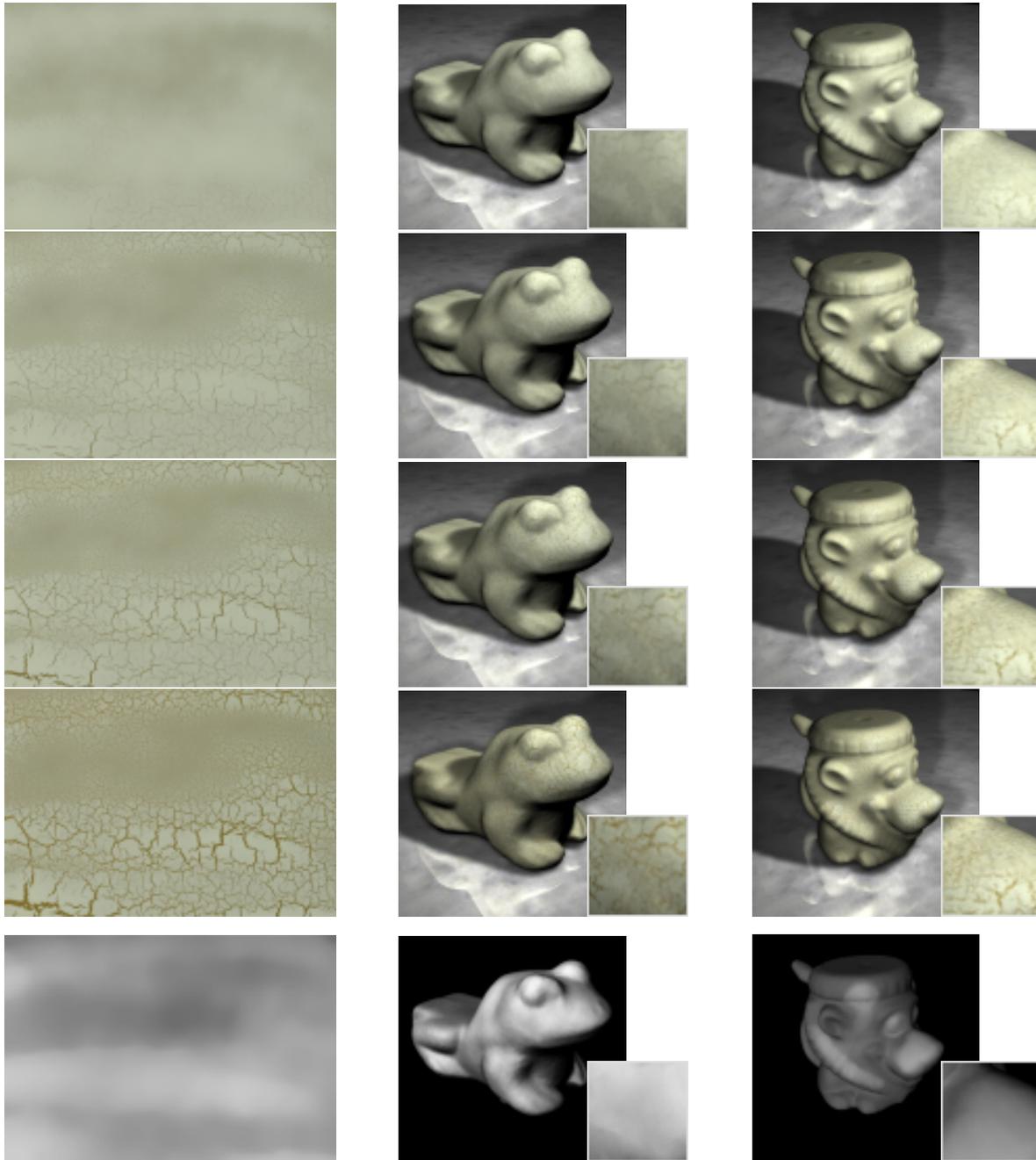


Fig. 16. Paint cracking, with context of paint thickness. Left column: original cracks on flat slide; Center column: paint cracking on a frog; Right column: paint cracking on a lion-shaped “sippy” cup. (Corresponding paint thickness maps are rendered in the bottom row.)



Fig. 17. Two more mechanical effects. Two columns on the left: dirt accumulation, with contexts of ambient occlusion and source direction (gravity), transferred from a statue of Bach to a statue of Chopin. Two columns on the right: paste cracking and peeling, with contexts of curvature and principal direction, transferred from a wave pattern to a granite pot.

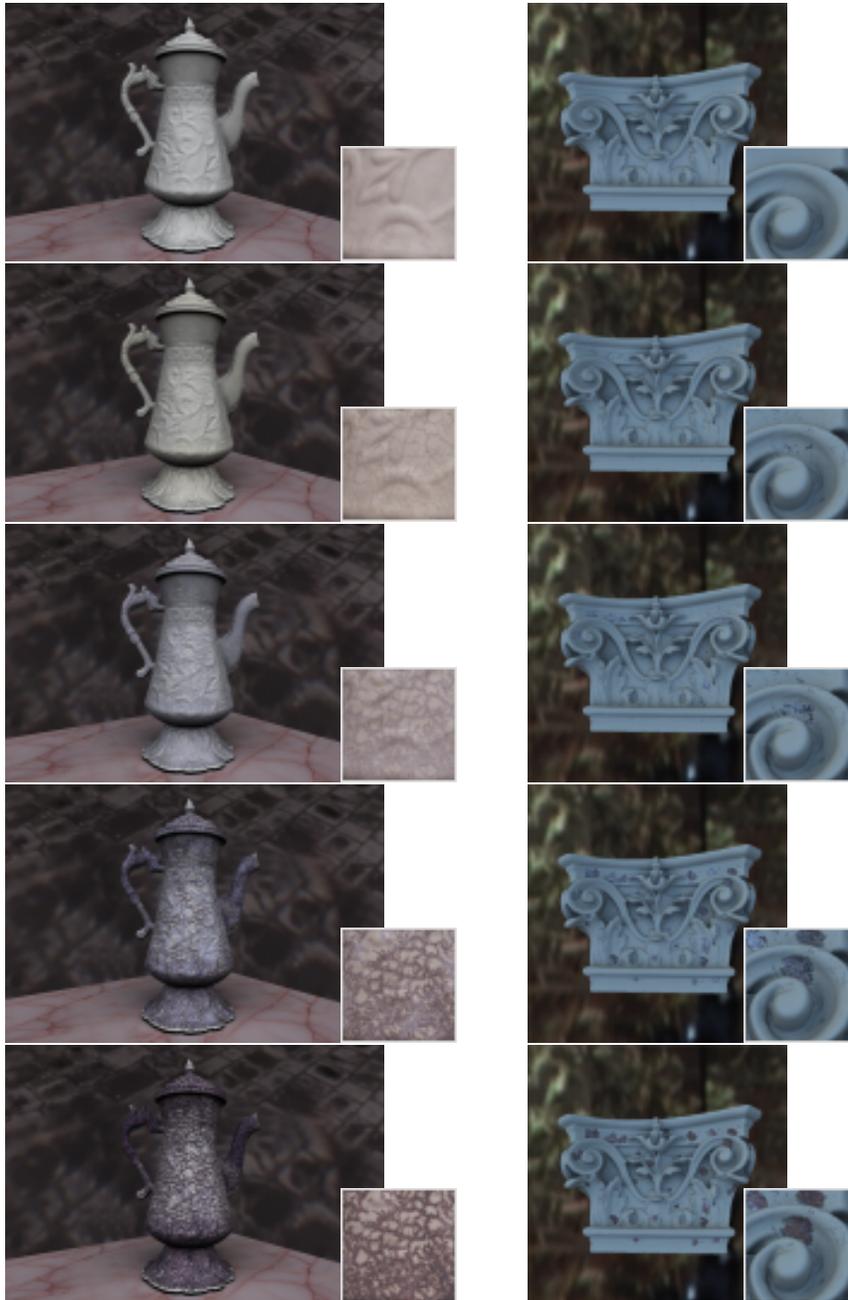


Fig. 18. Cascade effects. Left column: rusting effect (as illustrated in Fig. 13) following paint cracking (as illustrated in Fig. 16); Right column: the same rusting effect following paste cracking and peeling (as illustrated in Fig. 17).