Image-Guided Weathering: A New Approach Applied to Flow Phenomena

CARLES BOSCH
Yale University and REVES/INRIA Sophia-Antipolis
PIERRE-YVES LAFFONT
REVES/INRIA Sophia-Antipolis
HOLLY RUSHMEIER and JULIE DORSEY
Yale University
and
GEORGE DRETTAKIS
REVES/INRIA Sophia-Antipolis

The simulation of weathered appearance is essential in the realistic modeling of urban environments. A representative and particularly difficult effect to produce on a large scale is the effect of fluid flow. Changes in appearance due to flow are the result of both the global effect of large-scale shape, and local effects, such as the detailed roughness of a surface. With digital photography and Internet image collections, visual examples of flow effects are readily available. These images, however, mix the appearance of flows with the specific local context. We present a methodology to extract parameters and detail maps from existing imagery in a form that allows new target-specific flow effects to be produced, with natural variations in the effects as they are applied in different locations in a new scene. In this article, we focus on producing a library of parameters and detail maps for generating flow patterns; and this methodology can be used to extend the library with additional image exemplars. To illustrate our methodology, we show a rich collection of patterns applied to urban models.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*; I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.6.3 [**Simulation and Modeling**]: Applications

General Terms: Algorithms, Design

Additional Key Words and Phrases: Appearance modeling, weathering, rendering

ACM Reference Format:

Bosch, C., Laffont, P.-Y., Rushmeier, H., Dorsey, J., and Drettakis, G. 2011. Image-guided weathering: A new approach applied to flow phenomena. ACM Trans. Graph. 30, 3, Article 20 (May 2011), 13 pages.
DOI = 10.1145/1966394.1966399
http://doi.acm.org/10.1145/1966394.1966399

1. INTRODUCTION

Real materials change in appearance as a result of their interaction with the surrounding environment, and nowhere are such changes more apparent than in urban scenes. Variations in appearance are due to the specific type of weathering, the exposure and shape of an object, and the material of which an object is composed. Specifying and generating weathering effects on large-scale scenes remains a challenge. In this article, we present an approach for extracting appearance change data from photographs, which is then used to guide a weathering simulation with novel effects in a complex synthetic scene. As a representative type of weathering crucial to urban modeling, we consider the characteristic washing and staining patterns due to the flow of water over surfaces.

Our contribution is a new approach to weathering which uses photographs to drive a simulation, rather than simply reusing the photograph or patches of pixels from the photograph directly. Our approach is illustrated in Figure 1, outlining the major components required for this new approach. In particular, we present a method to separate the effect of the weathering phenomena (in this case stains) from the original object material, allowing the effect to be applied to materials of different color. We then introduce a method to extract simulation parameters from photographs of stains by optimization, allowing the effect to be applied to surfaces of different geometry. Finally, we show how we can extract high-frequency details of the effect, enabling simulation of weathering effects with natural-looking small-scale variations.

C. Bosch acknowledges a visiting grant from the University of Girona and an ANR project (ANR-06-MDCA-004-01). This work was also carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. INRIA acknowledges the generous support of Autodesk (Software donation of Maya and 3DSMax).

Authors' addresses: C. Bosch (corresponding author) and P.-Y. Laffont, REVES/INRIA Sophia-Antipolis, France; email: carles.bosch@inria.fr; H. Rushmeier and J. Dorsey, Yale University, New Haven, CT; G. Drettakis, REVES/INRIA Sophia-Antipolis, France.

DOI 10.1145/1966394.1966399
http://doi.acm.org/10.1145/1966394.1966399

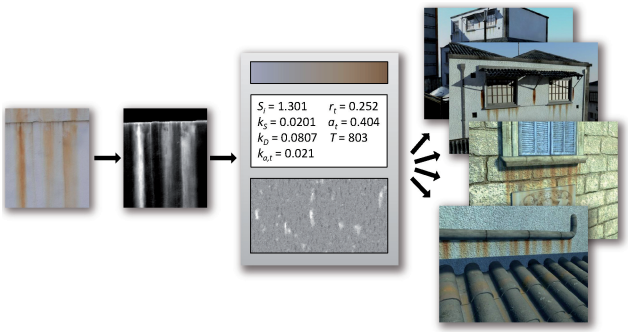

Fig. 1. From left to right: we start with a photograph of a weathering phenomenon, such the stain shown in the leftmost image: we want to obtain a similar weathering effect on new geometry. After some simple manual processing, we extract a *stain degree map* which indicates the degree of staining at each pixel of the original image (second block). The third block represents the main functionality of our solution, which first separates the stain from the original material with a nonlinear mapping of color (top box). We then estimate a set of parameters for a phenomenological flow model, resulting in a simulation that matches the stain of the input (middle box). Finally we extract details from the input image and synthesize a *detail map* (bottom box). Given the nonlinear color mapping, the simulation parameters, and the detail map, we can run simulations on a variety of different geometries, resulting in a wide variety of similar-looking stains in new scenes (rightmost images).

We begin with a brief review of work in weathered appearance that provides a context for our new approach, and, in particular, the motivation for a new method to generate flow stains.

2. PREVIOUS WORK

A variety of approaches have been brought to bear on the problem of generating aging effects for realistic image synthesis. The wide range of biological, chemical, and mechanical aging effects and the various methods applied to simulate them are described in surveys by Merillou et al. [2008] and in the book by Dorsey et al. [2007]. One approach is to have skilled artists assemble combinations of processed digital photographs and digital painting, applying image processing methods as described by Demers [2001] and demonstrated in work such as that by Clement et al. [2007]. In an alternative approach, researchers have developed physically-based and phenomenological simulation models to systematically generate effects that are tedious to produce manually, or to achieve a more realistic look. Specifically in the area of stains produced by flow, Dorsey et al. [1996] introduced a phenomenological simulation using a particle system to create flow effects. This approach was generalized to general transfer phenomena in the "gamma-ton" method by Chen et al. [2005]. Flow accessibility has been studied for simulating pollution as well [Mérillou et al. 2010].

Recently, methods to produce aged appearance based on photographic exemplars have been developed. Wang et al. [2006] introduced a technique based on capturing examples of different "degrees" of weathering due to a single effect in a single image. These examples form the basis of an *appearance manifold* that approximates an underlying subspace of weathered surface point appearances for a material. Xue et al. [2008] further developed this idea, focusing on the interactive editing of weathering effects in images with an eye to transferring effects from one image to another. Beyond using individual images, more sophisticated capture systems [Weyrich et al. 2009] have been used to obtain aged appearance data. This includes systems that capture time-varying reflectance data for flat surfaces [Koudelka 2004; Gu et al. 2006; Sun et al. 2007], aged appearance correlated to geometric effects [Mertens et al. 2006], and a system that captures time-varying effects as a function of geometry [Lu et al. 2007].

Systems that use captured data produce either pixel values or texture patches that are correlated to a control variable, such as time, weathering degree, and/or local geometric properties, such as curvature or orientation. Ambient occlusion, which is closely related to accessibility [Miller 1994] and surface exposure [Hsu and Wong 1995], has also been found to be highly correlated with weathering concentration. Fast texture synthesis methods can be used to map the captured values or patches to an object for which the control variable can be computed on the surface. When the control variable is a geometric property, it can be computed locally at each point on the surface. However, when the control variable is time or magnitude of exposure, such as solar irradiation, a simulation is needed. In the case of solar exposure, the full scene geometry needs to be considered, but the direct exposure calculation is independent of material type. A time- or degree-dependent control variable can be computed on a structure, and subsequently used with data acquired from a flat surface or image degree map to obtain spatial variations. For variables that depend on weathering degree, an additional problem is that an appropriate dataset or image with multiple weathering degrees is not available for an effect to be modeled. A more fundamental problem is when a control variable cannot be computed locally and/or independently of material. This fundamental problem occurs with flow phenomena (as well as with other phenomena such as cracking and erosion). Stains due to flows depend on global geometry: the full history along the path from where water first hit a surface to where stain particles are deposited. Furthermore, that history depends on the nature of the stain particle and the surface properties. For this reason, the capture-plus-control-variable paradigm fails for creating stains. This leaves manual application and phenomenological simulations as the methods available for stained surfaces.

Our focus here is generating weathering effects in urban scenes. The modeling of urban scenes has been the subject of great interest in the computer graphics and computer vision communities for decades. A major focus of these modeling methods has been in procedurally generating geometry for cities; a recent comprehensive survey can be found in Watson et al. [2008]. There has been increasing interest in generating associated textures for these methods, especially in the context of computer vision (e.g., Koutsourakis et al. [2009]). In all these methods, however, texture is reused "as-is" from the photograph or with minor preprocessing for lighting; aging is not treated. The texturing demands posed by today's procedurally generated urban-scale models call for efficient, automated approaches. For local effects or solar exposure, fast texture synthesis with control variables is becoming viable. However, flow effects on a large scale are still problematic.

A recent study by Endo et al. [2010] considered manual application and Dorsey et al.'s particle flow system as methods for creating realistic flow effects in imagery. They designed a new variant of the particle system that allows the user to interactively adjust the parameters to get the desired stain appearance. They found that adjusting the parameters was substantially faster than trying to paint in the stains in a photo-editing system. However, even with the new system, the trial-and-error user adjustment of parameters to get the stain correct took minutes per stain. Such an approach is not feasible for urban-scale scenes.

The problem then is how can we determine the parameters to achieve a particular effect? Can't we just look up the appropriate physical values and run the simulation? The major reason that parameters need to be estimated is that phenomenological models are physically inspired, not physically accurate. A physically accurate flow model would need to be more complex, such as the model for droplet flow by Wang et al. [2005]. Even that model is too simple, in that it does not include evaporation rates that are needed to determine the deposition of the solid particles. Detailed simulations would have to be run for extended periods, since physical stains are generally the result of many instances of water flowing over time. Where it is even possible to construct the accurate fluid simulation, the physical data to use in it is not available. Engineering efforts on weathering are focused on inhibiting the onset of aging, for example, on preventing rust, not on determining the properties of rust particles and how they are deposited on various building materials. The great success of the phenomenological models is that they can produce visually compelling results, without the full detail of an accurate physical flow model. It is of course important that physical principles inspire these methods.

An inspiration for resolving the parameter estimation problem is work in deriving procedural models for textures based on image exemplars [Bourque and Dudek 2004; Lefebvre and Poulin 2000]. Our basic idea is to expand the previous work to estimate the parameters of a process, rather than the parameters of a function that produces a texture. The link we need to do this is to identify data from an image that we can directly relate to data produced by the simulated process. This is a basic principle that could be applied to model many difficult aging processes, such as erosion or wear. In this article we focus on flow stains. The data we can obtain from stain images are degree maps, which we extract building on the work in Wang et al. [2006]. This is linked to the output of Dorsey et al. [1996], and in particular the map of deposited material that method creates. The link between these two approaches in previous work is the inspiration for the method presented here.

3. OVERVIEW

Our approach to overcome the problems of using either texture synthesis or procedural models alone is to extract parameters and details from captured images that can then be used in a phenomenological simulation. The simulation can then produce results that are similar to the captured image, but which are adapted to the specific target and that have natural variations in each instance.

We begin with a photograph of a weathering effect on a building, in particular a stained area produced by flows. We use approximate frontal images of flat surfaces or surfaces that can be approximated with a simple geometric proxy (e.g., a box or a cylinder). For future reference, we label the image by geometry of the source of the effect (e.g., point source such as a small pipe or pole, or line source such as a ledge or window sill), the source material (e.g., rust, chalked paint, biological growth, dirt) and the target of the material that is stained (e.g., concrete, stucco, bricks). We use simple user input followed by an automated process to segment out the weathered material in the image, resulting in a *stain degree map*. If exactly the same parameters and geometries were used in the simulated scenes as were found in the segmented images, the segmented images could just simply be copied onto the modeled geometry. However, even in this case the effects would need to have variations introduced to be realistic; an observer would notice the same stain repeated over and over again. Texture synthesis is not an appropriate solution in this case, since no control is provided to take the physical properties of the underlying phenomena into account. In addition, such methods have difficulty reproducing the regular structure of downward flows, for example. To account for the differences between the source photographs and target surfaces being stained, and to introduce natural variations in the stains we produce, we extract data from the photographs rather than reuse them directly.

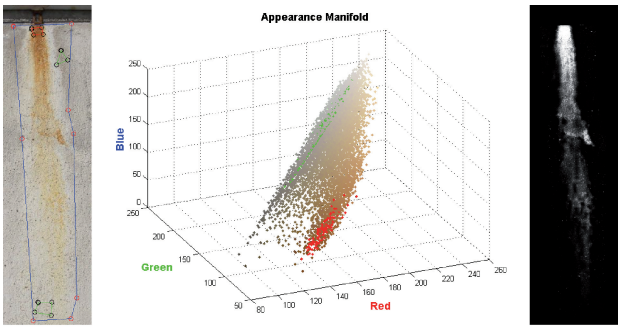

Fig. 2. Overview of stain extraction. Through a simple user interface (a), an appearance manifold (b) and weathering degree map (c) are constructed. The blue polyline in (a) corresponds to the approximate stain region, while the red and green regions contain the most and least stained points, respectively.

Three types of data are extracted from the segmented photograph; the nonlinear relationship between degree of staining and color, estimated parameters for a simulation, and maps of high spatial frequency details. These data are extracted from the photographs starting with a stain degree map. The nonlinear color variation in stains is estimated from a histogram of stain color as a function of stain degree. The parameters for a simple phenomenological method for flow are estimated using nonlinear optimization to find parameters that form a flow pattern similar to the stain degree map. Finally the difference between the flow pattern from the optimized parameters and the original stain degree map is used as a source for synthesizing a map of high spatial frequency details for stains.

To apply the extracted data in a particular scene, we run the phenomenological simulation with the parameters we have estimated. In the simulation, the properties of the target surface are modulated using the high spatial frequency details. Finally the flow is combined with the target surface background using the nonlinear color mixing procedure.

The extracted data is independent of the target scene. A user can collect many stain photographs and create a library of flow parameters and detail maps that can be used in their own scene modeling toolkit, or which can be shared with other users.

In the following sections we detail each of the steps in extracting reusable flow parameters and detail maps from a photograph of a stained surface.

4. STAIN IMAGE PROCESSING

We begin with an image of a stained surface. Stain processing involves two main steps: (i) manual selection of the approximate stain region and the most and least stained points and (ii) automatic construction of the appearance manifold and the weathering degree map. An overview of the process is illustrated in Figure 2.

Initially, the original image is displayed, and the user is requested to select the region of the stain, by indicating a set of points which create a polyline around the stain. Once the stain region has been selected, the user indicates the set of pixels X_0 which are the "least-stained" and the set of pixels X_1 that are the "most-stained."

We follow the approach in Wang et al. [2006] to use the marked pixels to construct an appearance manifold. In our case, the manifold is formed in RGB space. To reduce the number of pixels in the subsequent processing, we treat all pixels within some small distance in color space as being located in the same point in RGB space. As in Wang et al. [2006] we perform a dilation step to improve the initial user specification to a better estimate of the sets X_0 and X_1.

Once the appearance manifold has been created, the weathering (in our case, stain) degree of a point x is defined as a scalar function of appearance

$$D(x) = \frac{\phi(x, x_0)}{\phi(x, x_0) + \phi(x, x_1)}, \quad (1)$$

where x_0 and x_1 are respectively the closest points to x in the least and most stained sets, X_0 and X_1. $\phi(x, x_i)$, $i = 0, 1$ is the geodesic distance between two points in the appearance manifold, which we implement as the shortest path along connected nodes. With this definition, the stain degree D is defined within [0.0, 1.0], where 0.0 indicates the least stained points and 1.0 the most stained points.

5. THE FLOW MODEL

We use the stain degree map obtained in Section 4 as a spatial stain distribution map. We use this to fit flow simulation parameters to produce a deposition map that matches the stain distribution. In this section, we detail the flow simulation that we use.

The flow model is a "physically inspired" particle system using models for absorption and sedimentation modified from Dorsey et al. [1996]. In the system, water is modeled as particles with positions on a target surface. We denote time-varying quantities associated with individual water particles with the subscript i. Each water particle i has a position x_i, y_i on the target surface, a velocity v_i, mass m_i, and concentration of stain material S_i. The velocity of the particle depends on gravity g. The stain material carried by each water particle has an adhesion rate k_S and a solubility k_D. We omit the subscript i on the stain material properties, since they are the same for the stain material carried by all water particles. The values of k_S and k_D respectively model the likelihood that the stain material will stick to the target and be dissolved in water.

We denote quantities associated with the target surface with the subscript t. The target surface is characterized by its roughness r_t, maximum absorption a_t, absorptivity $k_{a,t}$, and coefficient of friction f_t. The roughness governs how much particles are deflected in the surface direction perpendicular to flow. The value of r_t is treated as the sine of the maximum angle of deflection of the flow from vertical. In the simulation a uniform random variable ξ is chosen to determine the actual deflection at each time-step. The absorptivity $k_{a,t}$ models the fraction of water absorbed at any point (x, y) as a function of the current surface wetness $w_t(x, y)$. The coefficient of friction f_t is used to decrease the velocity of water particles on the target. We use an empirical estimate of f_t as a function of wetness equal to $max(0.2, \frac{a_t - w_t(x,y)}{a_t})$. The value of 0.2 was determined empirically; values of less than 0.2 correspond to a completely wet surface, with essentially no friction.

At each x, y location on the target surface the $w_t(x, y)$ and deposited stain concentration $D_t(x, y)$ are stored and updated as water particles hit the location (x, y) during the simulation. The amount of water absorbed decreases the water particle mass m_i and increases the target wetness $w_t(x, y)$ unless the wetness at a target location has reached the saturation value of a_t. Similarly, the amount of material deposited decreases the particle stain concentration S_i and increases the target's value of $D_t(x, y)$.

Table I. Flow Model Parameters

Symbol	Parameter	Initial Guess	Bounds
$S_i(0)$	Initial stain concentration	1	[0.01, 5]
$m_i(0)$	Initial particle mass	1	fixed
k_S	Stain adhesion rate	0.04	[0.001, 0.5]
k_D	Stain solubility rate	0.04	[0.001, 0.5]
r_t	Target roughness	0.2	[0.001, 1]
a_t	Target maximum absorption	0.3	[0.001, 1]
$k_{a,t}$	Target absorptivity rate	0.05	[0.001, 0.5]
N	New particles per time step	15	fixed
$\Delta\tau$	Size of time step	1	fixed
T	Total time	300	[1, 5000]

The parameters not marked as *fixed* in the rightmost column are fit with our nonlinear optimization.

The particle simulation proceeds as follows, updating the target wetness and stain concentration as each particle is tracked through time.

Initialize $D_t(x, y)$ and $w_t(x, y)$ to 0 for all target locations
For each water particle i **do**
  Set position (x_i, y_i) at stain generator location
  Initialize particle attributes: $v_i = 0$, $m_i = 1$, $S_i = S_i(0)$
  For each time-step $\Delta\tau$ until T **do**
    $f_t(x_i, y_i) = max(0.2, \frac{a_t - w_t(x_i, y_i)}{a_t})$
    $v_{i,x} = v_{i,x} + (2\xi - 1) r_t |v_i| \Delta\tau$
    $v_{i,y} = v_{i,y} + g\Delta\tau$
    $x_i = x_i + (1 - f_t(x_i, y_i)) v_{i,x} \Delta\tau$
    $y_i = y_i + (1 - f_t(x_i, y_i)) v_{i,y} \Delta\tau$
    $m_i = m_i - (k_{a,t} \frac{a_t - w_t(x_i, y_i)}{a_t m_i}) \Delta\tau$
    $w_t(x_i, y_i) = w_t(x_i, y_i) + (k_{a,t} \frac{m_i (a_t - w(x_i, y_i))}{a_t}) \Delta\tau$
    $S_i = S_i + (-k_S S_i + k_D D_t(x_i, y_i) m_i) \Delta\tau$
    $D_t(x_i, y_i) = D_t(x_i, y_i) + (k_S S_i - k_D D_t(x_i, y_i)) \Delta\tau$
  end for
end for

In practice, rather than launching all water particles at once, N particles are produced at each new time-step $\Delta\tau$. This simulates the effect of water flowing down surfaces that have already had stains deposited and which are already partially wet. Each new generation of particles both washes away and deposits stain material.

Table I summarizes the parameters to be estimated. We note that the parameters are not independent. For example, adjusting the time-step $\Delta\tau$ would give a similar flow result with suitable scaling of the rate variables $k_S, k_D, k_{a,t}$ and the total time T. Similarly, a change in the particle mass would result in scalings of the stain concentrations and deposition values. Essentially choosing arbitrary values of mass and time establishes the base measurement units, for instance, similar to choosing ounces or grams for mass, seconds or hours for time, etc.

6. FITTING PARAMETERS

The core of our method is fitting the flow model parameters to a given stain degree map extracted from an image exemplar. This procedure is based on the assumption that the degree map obtained in Section 4 represents the stain concentration at every point in the image, so that a stain obtained by simulation can be directly compared to the input degree map. With this idea in mind, our basic approach is to first recreate the approximate geometry behind the given exemplar and then run the flow simulations until the obtained stain deposition matches the input stain degree map. The fitting step

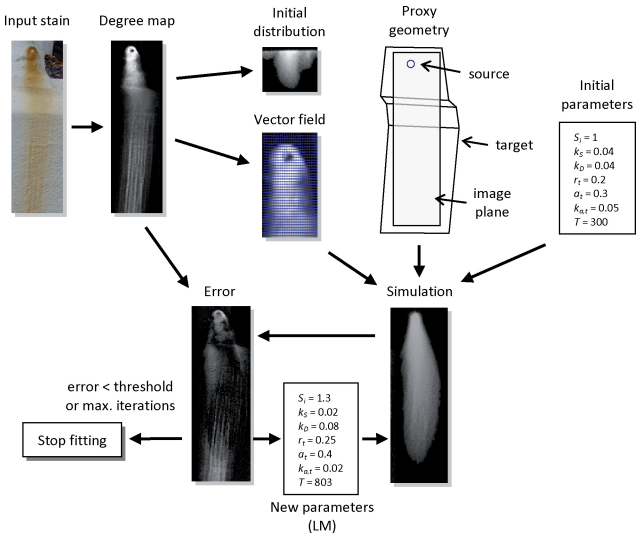

Fig. 3. Overview of the fitting procedure.

then consists on optimizing the set of flow parameters β (e.g., β represents the set $\{S_i(0), k_S, k_D, r_t, a_t, k_{a,t}, T\}$) so that the following expression is minimized. We have

$$E(\beta) = \sum_{j=1}^{n_p} [D_j - D_{t_j}(\beta)]^2 , \qquad (2)$$

where n_p is the number of pixels in the image, D is the degree map of the exemplar, and $D_t(\beta)$ is the simulated stain concentration obtained with the β parameters. We employ a constrained nonlinear optimization technique based on the Levenberg-Marquardt algorithm for minimizing this metric, using Lourakis's levmar library [Lourakis 2004].

The fitting procedure is depicted in Figure 3. First, the user specifies a *proxy* geometry representing the target surface where the stain appears, and the source geometry where the stained particles start. For stains appearing on facades, the proxy usually consists of a vertical plane, though alternate proxies can be used for other situations (e.g., stains over steps, slanted walls, curved tanks, etc.). The *source* geometry specifies the initial position of the particles on the model, and either a point, disk, or line projected on the proxy is used. For the simulation, an initial set of values is given to the different parameters and the simulation is started by generating the particles along the source, which flow onto the scene until they are absorbed or leave the proxy boundaries. Once the simulation is done (i.e., T has been reached), the generated stain pattern D_t and the input map D are compared by evaluating $E(\beta)$, and the fitting process is then repeated for a new combination of parameters chosen by the Levenberg-Marquardt algorithm.

The default initial values along with their bounds are included in Table I for reference. These values were determined from initial manual optimizations of particle flows for sample stains. These arbitrary, but specific, values are justified by the idea that we are looking for parameters to repeat the same stain pattern, not for physically accurate properties. Depending on the initial guess, different final parameter fits may be obtained. There are potentially many sets of parameters that would faithfully reproduce the stain pattern. For our application we need only one such set of parameters. As shown in the table, some of the parameters of the model are not estimated, but simply fixed. Fixed values can be used because not all the model parameters are independent, as noted in Section 5.

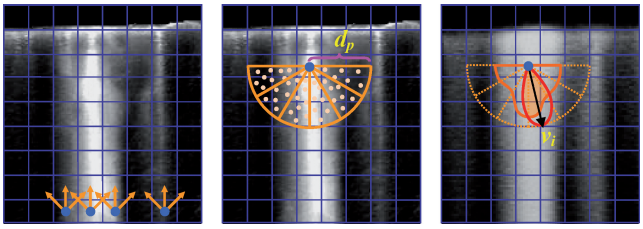

Fig. 4. Computation of the initial distribution of particles (left) and the flow vector distributions (middle) from the input degree map. During the fitting, the flow vector distributions are weighted by a Gaussian centered at the current particle direction (right).

To obtain a good match between the input and the simulated stains, the source should be placed as close as possible to the apparent source in the image, or some particles will not reach parts of the stain. This doesn't mean that the source should represent the real stain source, but just a simple geometry placed where the particles first appear on the target. For line sources such as window sills, the initial distribution of particles along the source needs to be estimated to obtain a stain shape similar to the exemplar. For simple vertical stains, the initial distribution is computed from the degree map by summing up the total weathering degree associated with every point on the source, following the vertical path of flow down the map. We discuss more complex configurations in the next section.

Our basic method fits parameters using one input image, however, it is possible to fit parameters using a set of images. We allow fitting parameters to multiple stains at the same time by sharing some of the properties between the exemplars. These properties are usually related to the surface or the stain material, and the rest of the parameters are independently fit for each stain.

7. ADJUSTING FOR GEOMETRIC VARIATIONS

The direction of the flow producing stains in an exemplar may be affected by obstacles on the target surface, such as pipes or fixtures, or by mesostructure variations in the target, such as bumps and dents. These changes in direction affect the stain degree map, so we must account for them to properly estimate the parameter set β. We need to address two issues: determining the initial source distribution, and adjusting the flow direction in the simulation.

7.1 Initial Distributions

Sources such as balconies or window sills tend to form stains with irregular spatial concentrations. The irregular concentrations appear as patterns of irregular strips (see Figure 6, middle row). To match these concentrations we can determine the distribution of stain material along a source using the degree map. Assuming that flow is mostly driven by gravity, just summing up the weathering degree along the vertical direction is enough in most cases. For more complex (typically nonvertical) flows, we need to collect the stain degree in a more general way. This is done by traversing the image from bottom-to-top and collecting the degree in an appropriate manner (see Figure 4, left).

This calculation starts by marking the pixels in the stain degree map that belong to the source. The stain degree map is then traversed from bottom to top row-by-row. The degree of every stained pixel is added to its neighboring pixels in the next higher

row, with a weight determined by their relative degree with respect to the other neighbors. Specifically, let the accumulated stain at pixel (i, j) be $D_{acc}(i, j)$. If three neighbors in the next row are considered, processing pixel (i, j) will increment values in row $j - 1$ according to the following.

$D_{sum} = D(i-1, j-1) + D(i, j-1) + D(i+1, j-1)$
For $m = i - 1$ **to** $i + 1$
  $D_{acc}(m, j-1) += D_{acc}(i, j) D(m, j-1)/D_{sum}$
end for

When a pixel belonging to the source is found, the accumulated degree is stored at the corresponding position on the source. The number of neighbors considered in the next higher row could be adjusted during this process depending on the spread of the stain in the stain degree map. However, we found that considering the three neighbors in the next higher row gives good results. The obtained degree distribution is then used during simulation to guide the emission of particles. This distribution is used during the fitting as it is very representative of the shape of the stains, but will also be used as a source 1D texture for generating new stains in the final scene, as described in Section 8.

7.2 Adjusting Flow Direction

During our simulation, we want to be able to redirect particles in order to match flow variations in the degree map. By locally analyzing the degree variations in the map, we can determine the directions that the flow may take at every point, and use this information to modify our particles. We found that using an existing vector field calculation, such as gradient vector fields [Xu and Prince 1997], produced unsatisfactory results to direct the flow; we thus developed an alternate approach based on local distributions of vectors.

Our calculation of vectors to direct the flow during the parameter fitting is performed by analyzing the local variations of degree in the input map as a preprocess (see Figure 4, middle). For every point, we randomly sample a neighborhood of a given distance in the directions with a positive component of the gravity vector. We discretize the directions from the point into a number of bins. We compute a stain degree distribution in each direction from the point. Given the number of directional bins n_b, the number of random samples n_s, and the neighborhood distance in pixel units d_p, the stain degree for each directional bin b is given by

$$D'(x, y, b) \;=\; \sum_{k=1}^{n_{s,b}} D(x + r_k cos\theta_k,\; y - r_k sin\theta_k), \qquad (3)$$

where $n_{s,b}$ is the number of random samples lying on the current bin, and (r_k, θ_k) is the sample's polar coordinates relative to the current position (x, y).

During the parameter estimation, the distribution map D' is used to vary the particle directions. As we want to obtain a particle dispersion consistent with the roughness being estimated, the distribution at every point is first weighted by a function of the current roughness, centered at the current particle direction. We tried both a Gaussian weighting and a uniform weighting, and the former tended to give better fitting results. This weighting is computed with

$$D'_w(x, y, b) \;=\; D'(x, y, b)\, e^{-\frac{(\theta_b - \theta_{v_i})^2}{2r^2}}, \qquad (4)$$

where θ_b is the half angle of the bin, θ_{v_i} is the angle of the particle velocity, and r is the roughness value. Once the distribution is weighted, we then sample it in order to obtain a new particle direction at the current point. This procedure is detailed in Figure 5.

1. Retrieve degree distribution $\rightarrow D'(x, y)$
2. Weight distribution using Equation (4) $\rightarrow D'_w(x, y)$
3. Get cumulative distribution $\rightarrow C'(x, y, b) = \sum_{b_i \le b} D'_w(x, y, b_i)$
4. Get random value $\rightarrow n = \xi\, C'(x, y, n_b)$
5. Find corresponding bin $\rightarrow b_i;\; C'(x, y, b_{i-1}) < n \le C'(x, y, b_i)$
6. Get relative position inside the bin $\rightarrow q = \frac{n - C'(x,y,b_{i-1})}{C'(x,y,b_i) - C'(x,y,b_{i-1})}$
7. Get new particle angle $\rightarrow \theta_i = \pi\,(b_i + q)/n_b$
8. Rotate velocity around target's normal $\rightarrow v'_i = R(N_t, \theta_i - \theta_{v_i})\, v_i$

Fig. 5. Pseudocode for sampling a distribution.

During the simulations, the parameters n_b and n_s affect the quality of the distributions obtained. The value of d_p specifies whether the flow should adapt to local or global degree variations. A low value of d_p means that the flow will try to match local variations of degree, resulting in a more detailed pattern but less sensitive to the global stain flow. In our examples, the following values are used: $n_b = 15$, $n_s = 100$, $d_p = 10$.

Figure 6 shows some fitting results for a point source (top), a line source (middle), and a point source over a complex target (bottom). For the line source, both the particle distribution and the flow variation were obtained from the input degree map, as described before. In Figure 7, our deflection model is compared to a model based on gradient vector fields [Xu and Prince 1997], where a single vector is used to model the flow variation at each point. As can be seen, the complex flow distributions that are found on stains cannot be well represented with simple vector fields. The image also shows a comparison without using any deflection model, where the obtained simulation does not match the input stain as well as when using our deflection model.

It is important to note that the vector fields computed to direct the flow during fitting are not used in generating new stains. They are only used to obtain values of the parameter set β. The direction of flow in new stains is determined by the new target geometry, as described in Section 8.

7.3 Detail Maps

The simulations that result from the parameter fitting phase fit the overall shape of the stain, but do not have the high-frequency spatial variations that are typically found in the real world. To add these details, we synthesize a detail map in the spirit of Jagnow et al. [2004]. An initial image of details alone is computed by taking the difference of our optimized simulation and the stain distribution map found from the original stain image. This difference image can not be used directly in creating new stains, since it is limited in extent to the shape of the original stain. To create a large map for all possible stains created from the original example, we use texture synthesis (specifically the method described by Lefebvre and Hoppe [2005]), to generate a large image of high-frequency details. The method of Lefebvre and Hoppe [2005] was adapted to take a guide map into account in the spirit of image analogies [Hertzmann et al. 2001], using the stain shape as guide. In order to avoid undesirable artifacts, this guide is further adjusted so that regions with high difference are not considered. Then, after the synthesis, we apply the obtained detail map onto the target surfaces and use it to adjust the adhesion of material during the simulation in the output scene.

8. SCENE SPECIFIC STAINS

The parameters and maps obtained in the previous sections can be applied to generate stains on new scenes.

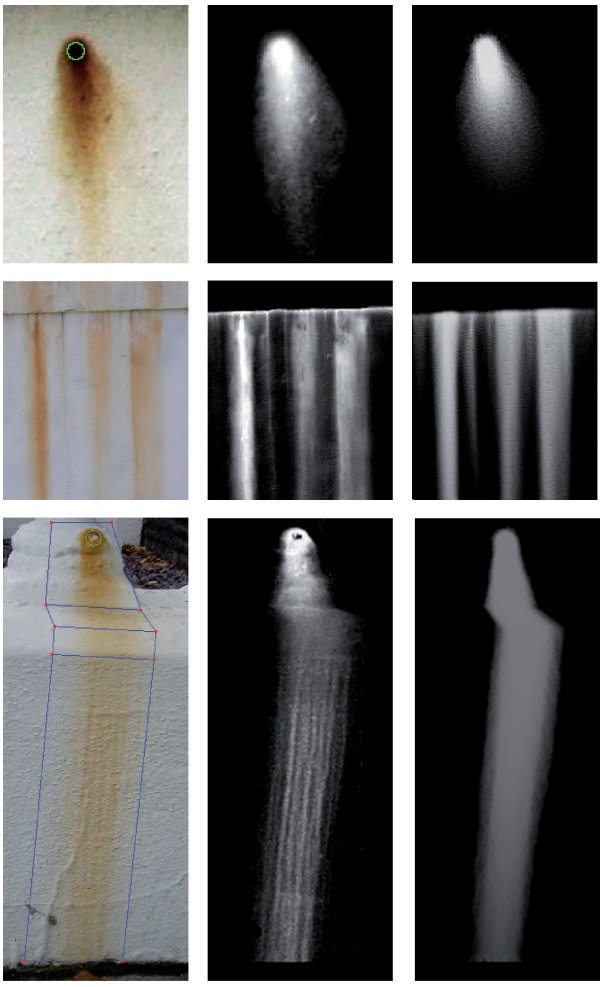

Fig. 6. Examples of (a) input stains, (b) degree maps, and (c) simulation results using estimated parameters for a point source (top row), line source (middle row), and a nonvertical target (bottom row). In the input images, source geometry is shown in green and target geometry in blue. We can see that the simulation resulting from the optimization process matches the form of the stain well.

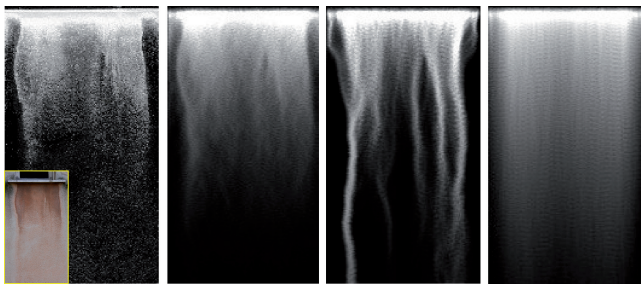

Fig. 7. From left to right: Input degree map, fitting with our flow deflection model, fitting with a gradient-based vector field, and fitting without deflection. The input color image is shown in the bottom left inset.

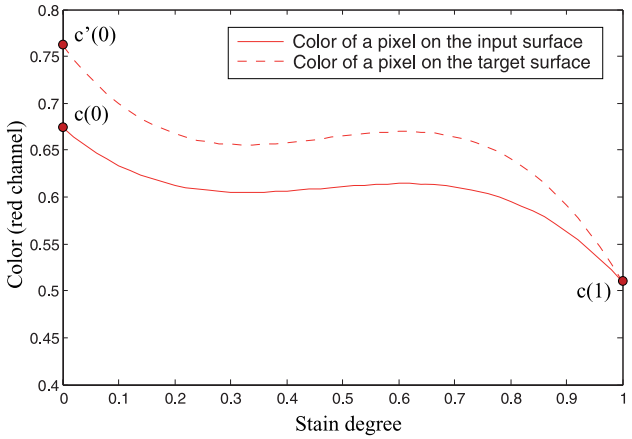

Fig. 8. The nonlinear variation of each color channel c as a function of stain degree is estimated from the appearance manifold. The values of the color channel can be adjusted for a new target color $c'(0)$ by warping the function.

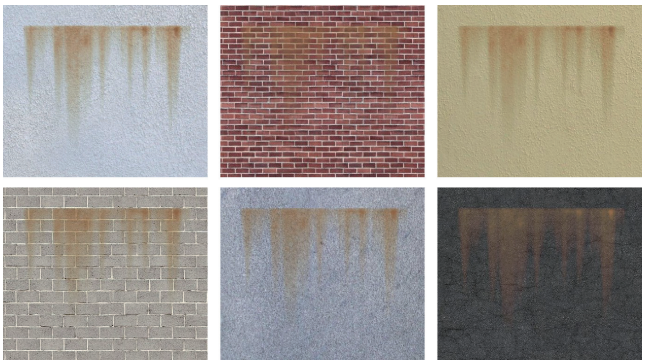

Fig. 9. A simulated stain applied to clean target surfaces with different background material and colors, using our warping approach. The input exemplar is shown in Figure 6, middle row.

The parameters represent material properties associated with stain sources and target background materials. In modeling a new synthetic scene, a model feature's propensity to act as a stain source is associated with the material assigned to that feature. In our implementation, we associate source material names with stain data (parameters, detail map, etc.).

The same data producing a stain model may be applied to many similar features in a scene, such as sets of sills on windows, or multiple pipes on a building. Natural variations are introduced in stains generated from the same parameters by introducing small perturbations in the values. If several input stains are available for a material model, the parameters are then obtained by randomly interpolating among the set of parameters. As noted in the previous section, for the 1D density distributions of line sources, we use texture synthesis in one dimension to create new density distributions that are similar to the original. The length of each distribution is given by the relative length of the new source with respect to the input exemplar. Similarly, the number of particles that are generated at each time-step (N) is proportional to this length. In this way, the shape of the stain is correctly maintained on every source.

In running the simulation, the detail map is used to vary the particles' stain adhesion rate, k_S as a function of its position on the

Fig. 10. Input exemplars for the stains simulated in Figures 11–15.

Fig. 11. Rust stain simulated over different scenes showing a variety of geometries and backgrounds. Figure 10, top left is used as input image.

target. The variations on the stain flow are then given by the target geometry. Displacement maps can alternatively be used to modify the flow along the target surface [Dorsey et al. 1996], or directly use the color texture of the target to compute these variations [Endo et al. 2010]. We use this latter approach but based on a simplified expression, where roughness is not used to amplify the influence of the map. In our case, we also want roughness to affect the flow even if no variations are found, thus the following expressions are used:

$$v_{i,x} = v_{i,x} + ((2\xi - 1)r_t(|v_i| + M_y(x_i, y_i)) - |v_{i,x}|M_x(x_i, y_i))\Delta\tau, \quad (5)$$

$$v_{i,y} = v_{i,y} + (g - v_{i,y}M_y(x_i, y_i))\Delta\tau, \quad (6)$$

where $M_x(x_i, y_i)$ and $M_y(x_i, y_i)$ are the luminance variations at a given point in the target.

Running particle simulations on complex scenes may require a lot of computational effort, mainly due to the particle-surface collisions. For efficiency, we use an intersection cache based on uniform grids that are aligned with the main building facades. This basic solution reduces computation time considerably.

After running the simulations, the resulting detailed stains are finally deposited as stain images. These images must then be combined with the underlying "clean" target texture.

8.1 Color Variations

Unlike Wang et al. [2006], we will apply the stain appearance to background materials that are not precisely the same color as the original source. We note that in each partially stained point, there is a mixture of the color of the background and the color of the stain. This mixture is not linear with the degree of stain obtained. We separately compute the mixture of colors as a function of degree. To do this, we form histograms of R, G, and B values for 50 bins of weathering degree values using the appearance manifold data, and we fit polynomials to these histograms, $R(D)$, $G(D)$, and $B(D)$. The triplet $(R(0), G(0), B(0))$ corresponds to the underlying clean surface color, while $(R(1), G(1), B(1))$ corresponds to the full stain color. As the color of the new target is generally not the same as in the original stain sample, we use these polynomial functions to replace the underlying target color. With the following equation, we "warp" the stain colors to a new clean target color $(R'(0), G'(0), B'(0))$, for c equal to R, G or B.

$$c'(D) = s\, c(D) + (1 - s)c(1); \quad s = \frac{c'(0) - c(1)}{c(0) - c(1)} \quad (7)$$

Fig. 12. Urban scene with stains simulated on it. "Clean" versions are shown below each row, while input stains are included on the right column.

By modeling the color channels separately, we capture the effect of changing stain hue as a function of degree, as well as separating out the background color. The variation of a color channel before and after warping to a new clean target is shown in Figure 8. Note that the functions $c(D)$ and $c'(D)$ are both nonlinear. Only the "warping" between the two functions is linear. Eq. (7) may result in negative or extreme scalings when the original and target backgrounds are considerably different. This effect, however, is only noticeable in some extreme cases.

Figure 9 shows a simulated stain blended with different target textures, demonstrating how the warping approach correctly combines the stain with the different backgrounds. The bottom-right image shows an example in which the warping might give unrealistic results, as input and target backgrounds are extremely different (input is white and target is black). In this case, a more appropriate input exemplar should be used instead.

9. RESULTS

In this section, we show results of our approach for simulating stains on new synthetic scenes. The input images used in the examples are included in Figure 10. These images show a variety of stain materials and background surfaces that are typically found in real environments, including rust, efflorescence, mold, or green biological growth. The parameters, distributions, detail maps, and color information extracted from the images and other exemplars are included in the supplementary material.

Fig. 13. More buildings from the urban scene with stains simulated on them.

Fig. 14. Close-up view of two stains from Figure 12 bottom-left. Left-Middle: Stain maps after simulation. Right: Final result after color warping and lighting.

Figure 11 shows a single stain applied in different scenarios, where the stain starts from two different sources and flows over several target surfaces. The parameters, detail maps, and color information for the stain and the target objects were obtained from Figure 10 top-left image. As can be seen, the shape and appearance of the obtained stains closely match the input exemplar, while offering new interesting flows and patterns by means of the simulation. These effects could not be obtained with synthesis methods, for instance. During simulation, the background texture of each target is used to change flow direction based on the luminance-based deflection model described in Section 8, while the final appearance is obtained through our warping approach of Section 8.1. In these examples, the distribution of particles and the particle rate are adapted for each source according to its length, thus obtaining a similar stain shape along the different sources.

Figures 12, 13, and 15 show more complex scenes with urban environments, where the different stains of Figure 10 have been used. For every type of source (sills, pipes, etc.), we mostly use a single stain image as input. The variety in the stains is then obtained by means of the 1D synthesis of their distributions, the detail map associated to the target, and the geometric variations of the target itself. For the efflorescence stains of the brick wall (Figure 12, bottom rows), we used three input stains, corresponding to the top rightmost images of Figure 10. The combination of different input stains offers yet more variety on the obtained stains, as their different distributions and parameters are combined during the simulation.

In Figure 14, a close-up view of two stains from one of the buildings of Figure 12 is shown. The stain maps obtained by the simulation are shown on the left and middle images. Note how the stain maps are affected by the detail map of the input exemplars and the geometric features of the target. These maps are then combined during the color warping and rendered for the final result (right image).

The urban scene in Figures 12 and 13 contains a total of 91 line sources and 50 point sources, while the Mediterranean model of Figure 15 has 7 line sources and 2 point sources. The stain simulations required 2 to 5 minutes per stain on a 2.33Ghz Xeon machine, depending on stain extension and texture resolution. Performing the color warp required 5 to 8 additional seconds per stain, depending on texture resolution too. All renderings are done with MentalRay

Fig. 15. Mediterranean model with stains simulated on it. “Clean” versions are shown on the left, while input stains are included on the right.

for Maya. In terms of preprocessing, the appearance manifold computation takes 1 to 3 minutes, while fitting for 500 iterations takes 30 to 60 minutes, depending on image resolution. A 1024^2 detail map synthesis takes 1 to 2 minutes, depending on the stain. All timings refer to unoptimized implementations in C++, except for the warping and appearance manifold construction, which are performed in Matlab.

Limitations. Our models from images are clearly limited to the effects for which we can find photographs from a reasonable frontal view. The whole process also relies on good quality extractions of

the stains, so stains over complex backgrounds should be avoided. A small amount of manual intervention is required to indicate the stained areas in the photographs, which is also derived from the appearance manifold method [Wang et al. 2006]. The parameters we extract from the photographs are not true physical properties. Rather, they are simple parameters that form similar patterns when used in the physically inspired particle system. The synthesized detail map may contain features related to the underlying geometry of the photograph. For consistency, these should be applied to objects with similar characteristics.

Our system still takes considerable time for high-resolution images. However, preview images at lower resolution to show the stain distributions across a scene can be computed much faster. Furthermore, all simulations can be computed in parallel.

Our model computes stains that are similar to the results in Dorsey et al. [1996]. However, our approach requires no trial and error to estimate parameters; it also adds details derived from images (detail map), and is able to model variations in color in stains (color warp) that did not appear in the earlier work.

10. CONCLUSION

We have presented a new method for using photographs as input for generating realistic imagery. Rather than use photographs directly for texture synthesis, we instead extract simulation parameters and detail maps that are used to generate scene-specific stains with natural variations. By making use of photographs to generate realistic patterns, our approach eliminates the difficult and sometimes impossible task of finding appropriate parameters for simulations. Future work includes applying a similar approach to extract data from images of other types of aging effects as the basis for a system capable of generating a broad palette of weathering phenomena. Given the parameters for individual weathering effects, cascading effects (weathering effects that produce patterns that in turn contribute to additional effects) could be applied without the need for locating images for each possible combination of effects.

ACKNOWLEDGMENTS

We would like to thank Li Ying and Su Xue for their help in the extraction of stains and the texture synthesis part. We also thank Samantha Close and Fernanda Andrade-Cabral for their help with the scene models.

REFERENCES

BOURQUE, E. AND DUDEK, G. 2004. Procedural texture matching and transformation. *Comput. Graph. Forum 23,* 3, 461–468.

CHEN, Y., XIA, L., WONG, T.-T., TONG, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2005. Visual simulation of weathering by γ-ton tracing. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. ACM, New York, 1127–1133.

CLÉMENT, O., BENOIT, J., AND PAQUETTE, E. 2007. Efficient editing of aged object textures. In *Proceedings of the 5th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa (AFRIGRAPH'07)*. ACM, New York, 151–158.

DEMERS, O. 2001. *Digital Texturing and Painting*. New Riders Publishing, Thousand Oaks, CA.

DORSEY, J., PEDERSEN, H. K., AND HANRAHAN, P. 1996. Flow and changes in appearance. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96)*. ACM, New York, 411–420.

DORSEY, J., RUSHMEIER, H., AND SILLION, F. 2007. *Digital Modeling of Material Appearance*. Morgan Kaufmann Publishers, San Francisco, CA.

ENDO, Y., KANAMORI, Y., MITANI, J., AND FUKUI, Y. 2010. An interactive design system for water flow stains on outdoor images. In *Smart Graphics*. 160–171.

GU, J., TU, C.-I., RAMAMOORTHI, R., BELHUMEUR, P., MATUSIK, W., AND NAYAR, S. 2006. Time-Varying surface appearance: Acquisition, modeling and rendering. In *ACM SIGGRAPH 2006 Papers (SIGGRAPH'06)*. ACM, New York, 762–771.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of ACM SIGGRAPH International Conference on Computer Graphics and Interactive Techniques*. 327–340.

HSU, S.-C. AND WONG, T.-T. 1995. Simulating dust accumulation. *IEEE Comput. Graph. Appl. 15*, 18–22.

JAGNOW, R., DORSEY, J., AND RUSHMEIER, H. 2004. Stereological techniques for solid textures. *ACM Trans. Graph. 23,* 3, 329–335.

KOUDELKA, M. L. 2004. Capture, analysis and synthesis of textured surfaces with variation in illumination, viewpoint, and time. Ph.D. thesis, Yale University, New Haven, CT.

KOUTSOURAKIS, P., SIMON, L., TEBOUL, O., TZIRITAS, G., AND PARAGIOS, N. 2009. Single view reconstruction using shape grammars for urban environments. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'09)*.

LEFEBVRE, L. AND POULIN, P. 2000. Analysis and synthesis of structural textures. In *Proceedings of the Graphics Interface*. 77–86.

LEFEBVRE, S. AND HOPPE, H. 2005. Parallel controllable texture synthesis. *ACM Trans. Graph. 24,* 3, 777–786.

LOURAKIS, M. 2004. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. `http://www.ics.forth.gr/~lourakis/levmar`. [Accessed].

LU, J., GEORGHIADES, A. S., GLASER, A., WU, H., WEI, L.-Y., GUO, B., DORSEY, J., AND RUSHMEIER, H. 2007. Context-Aware textures. *ACM Trans. Graph. 26,* 1, 3.

MÉRILLOU, N., MÉRILLOU, S., GHAZANFARPOUR, D., DISCHLER, J.-M., AND GALIN, E. 2010. Simulating atmospheric pollution weathering on buildings. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'10)*. 65–72.

MÉRILLOU, S. AND GHAZANFARPOUR, D. 2008. Technical section: A survey of aging and weathering phenomena in computer graphics. *Comput. Graph. 32,* 2, 159–174.

MERTENS, T., KAUTZ, J., CHEN, J., BEKAERT, P., AND DURAND, F. 2006. Texture transfer using geometry correlation. In *Proceedings of the Eurographics Symposium on Rendering*.

MILLER, G. 1994. Efficient algorithms for local and global accessibility shading. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 319–326.

SUN, B., SUNKAVALLI, K., RAMAMOORTHI, R., BELHUMEUR, P. N., AND NAYAR, S. K. 2007. Time-varying BRDFs. *IEEE Trans. Visualiz. and Comput. Graph. 13,* 3, 595–609.

WANG, H., MUCHA, P. J., AND TURK, G. 2005. Water drops on surfaces. *ACM Trans. Graph. 24,* 3, 921–929.

WANG, J., TONG, X., LIN, S., PAN, M., WANG, C., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Appearance manifolds for modeling time-variant appearance of materials. *ACM Trans. Graph. 25,* 3, 754–761.

WATSON, B., MÜLLER, P., VERYOVKA, O., FULLER, A., WONKA, P., AND SEXTON, C. 2008. Procedural urban modeling in practice. *IEEE Comput. Graph. Appl. 28,* 3, 18–26.

WEYRICH, T., LAWRENCE, J., LENSCH, H. P. A., RUSINKIEWICZ, S., AND ZICKLER, T. 2009. Principles of appearance acquisition and representation. *Found. Trends. Comput. Graph. Vis. 4,* 2, 75–191.

XU, C. AND PRINCE, J. L. 1997. Gradient vector flow: A new external force for snakes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'97)*. 66–71.

XUE, S., WANG, J., TONG, X., DAI, Q., AND GUO, B. 2008. Image-Based material weathering. *Comput. Graph. Forum 27,* 2, 617–626.

Received November 2010; accepted February 2011