# Time-varying Weathering in Texture Space

Rachele Bellini Yanir Kleiman Daniel Cohen-Or
Tel Aviv University

**Figure 1:** *Snapshots from three series of time-varying texture weathering computed from single input images presented in the middle column.*

## Abstract

We present a technique to synthesize time-varying weathered textures. Given a single texture image as input, the degree of weathering at different regions of the input texture is estimated by prevalence analysis of texture patches. This information then allows to gracefully increase or decrease the popularity of weathered patches, simulating the evolution of texture appearance both backward and forward in time. Our method can be applied to a wide variety of different textures since the reaction of the material to weathering effects is physically-oblivious and learned from the input texture itself. The weathering process evolves new structures as well as color variations, providing rich and natural results. In contrast with existing methods, our method does not require any user interaction or assistance. We demonstrate our technique on various textures, and their application to time-varying weathering of 3D scenes. We also extend our method to handle multi-layered textures, weathering transfer, and interactive weathering painting.

**Keywords:** weathering, textures, time-varying textures, regularity

**Concepts:** •**Computing methodologies** → **Image processing; Texturing;**

## 1 Introduction

Synthesis of realistic virtual objects and images is useful for a large range of applications, from engineering prototypes to virtual movie sets. Generating realistic models is therefore a central goal in the field of computer graphics. A substantial part of research in that area focuses on modeling realistic appearances of objects in a static point in time. However, time can have a significant effect over object appearance, leading to a common phenomenon known as *weathering*. Both color and geometry can change dramatically over time: a stone statue loses details, varnished wood peels revealing new colors, ceramic tiles crack, dust settles on surfaces, etc. In contemporary computer graphics productions, weathered object appearance is usually realized by mapping weathered textures over the surface of an object. Time-varying textures are needed to simulate the evolution of an object over time. In the absence of automatic tools, the generation of such time-varying textures is a meticulous endeavor performed by artists.

Weathering process synthesis is a challenging task since different materials evolve in different ways over time. Developing dedicated simulations for specific weathering effects is not effective, as within the weathering process materials may be subject to gradual non-uniform transformations that alters surface coloring (e.g., rust, dust, stains) as well as causes geometric deformations (e.g., cracks, holes, erosion). On the other hand, addressing many weathering effects with a single generic process is challenging.

In this paper, we introduce a generic technique to address the problem of time-varying weathered textures synthesis. Given a single weathered input texture, we analyze it and synthesize a sequence of images mimicking the texture evolution over time. The technique can generate backward and forward time series, that convey a de-weathering and weathering effect. A few examples are given in Figure 1, where the input texture is shown in the center, de-weathered textures on the left, and weathered textures on the right. Our method is based on the premise that textures are stationary and consist of repetitive or partially repetitive patterns, while natural weathering effects such as cracks, dust, stains, etc., are unique and typically not repetitive. Thus, the key observation is that recurring image patches are more likely to be part of the original texture pattern, while patches that do not recur are more likely to be a result of weathering effects. Based on this observation, the method first infers from the input texture the *weathering age* of each part of the


SIGGRAPH '16 Technical Paper, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07 $15.00
DOI: http://dx.doi.org/10.1145/2897824.2925891

**ACM Reference Format**
Bellini, R., Kleiman, Y., Cohen-Or, D. 2016. Time-varying Weathering in Texture Space.
ACM Trans. Graph. 35, 4, Article 141 (July 2016), 11 pages. DOI = 10.1145/2897824.2925891
http://doi.acm.org/10.1145/2897824.2925891.

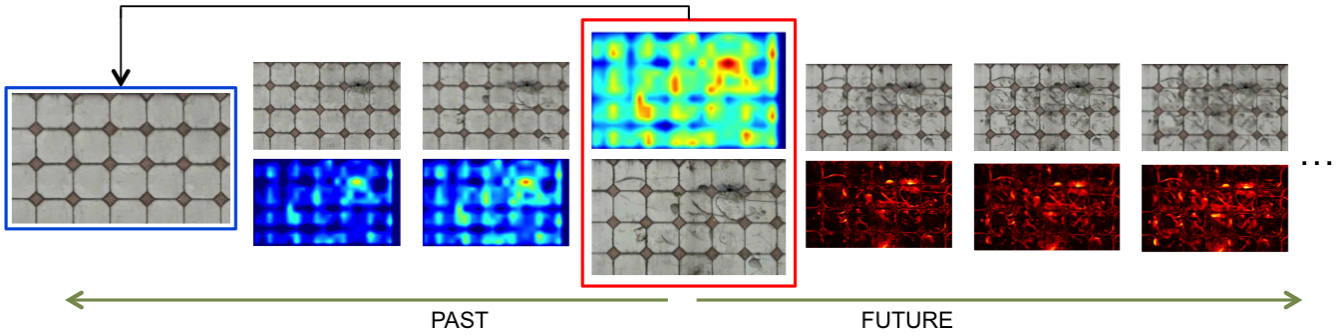



**Figure 2:** *Overview of the method. An intact texture (left) is generated using the age map of the input texture (center), and used to synthesize a series of textures emulating de-weathering (left) and weathering (right) of the input texture. The age maps of de-weathered textures and difference maps for weathered textures are displayed in the second row.*

image, and generates an *age map*. The definition of the age map is central as it guides the subsequent steps of the weathering process. Using the age map, we synthesize the *intact texture*, representing the appearance the input texture had when it was initially created, prior to any weathering effects. The intact texture is then used along with the age map for the de-weathering and weathering synthesis, simulating the manner in which the weathering progresses or retreats over time, generating a time-variant texture appearance.

The method is physically-oblivious and thus generic, and can deal with various types of textures, including both stochastic textures, and patterned textures which are regularly repeated. As we shall show, it yields believable and plausible weathering effects. In addition, our method can be used for controlling the regularity or entropy of a texture rather than its time-based weathering. For example, the wooden texture in the third row of Figure 1 is varied between a clean, intact version, and a high entropy version which is less regular and exhibits more unique artifacts and features. However, these changes do not reflect the passing of time but an inherent difference in the regularity of the texture.

## 2 Background and Related Work

**Physically-based weathering.** A common approach for material weathering is the simulation of a physical process on a 3D model [Dorsey et al. 2005; Chen et al. 2005; Xue et al. 2011]. These methods produce a high level of realism, based on the knowledge of the manner in which a specific material is weathered. For example, stone [Xue et al. 2011], paint [Paquette et al. 2002] and metal [Dorsey and Hanrahan 2006]. Other methods deal with particular forms of weathering, such as flows [Bosch et al. 2011; Dorsey et al. 2005], corrosion [Merillou et al. 2001], dust accumulation [Hsu and Wong 1995], cracks [Iben and O'Brien 2009; Aoki et al. 2004; Valette et al. 2008] and lichen growth [Desbenoit et al. 2004]. Requiring a specialized formulation of a particular phenomenon, they are not generic and cannot be easily extended. Methods such as [Chen et al. 2005; Wong et al. 1997], simulate interactions between the object geometry and weathering factors in the environment.

A different family of methods simulate weathering surface variations through texture synthesis. Mukai et al. [2003] propose to synthesize weathering effects using fractal patterns. Backet and Badler [1990] present a system that creates blemished or dusty surfaces through procedural texturing.

Our method is generic and oblivious to the physics, and can be applied to a wide range of different types of materials and effects. Even though our results are not physically modeled, they tend to have a realistic weathering appearance.

**Time-varying weathering.** Several works present methods that deal with the weathering of materials over time. A data-driven approach to time-varying weathering synthesis has been shown to be effective [Lu et al. 2007; Georghiades et al. 2005]. These methods require special equipment for capturing time-series data, material disposal, and are time-consuming and demanding during acquisition. In contrast, our method is highly accessible, requiring merely a single image of a weathered texture as input.

Other works, like ours, are based on data captured at a single time instant. Wang et al. [2006] require the capturing of spatially variant BRDF, while Xue et al. [2008] and Bandeira and Walter [2009] use casual images as input. In these works the user is required to manually pick the color of the least and the most weathered point in the image. Based on the selected colors, a non-linear weathering color space is defined where every color corresponds to a weathering degree. The weathering level of the image is increased or decreased by shifting pixel colors within the color space. These works provide realistic modeling of weathered appearances of materials. Like our method, they do not handle a specific material or weathering factor. However, these methods only account for smooth color variations, while texture structure is not affected or modified. The appearance evolution takes place in the color space only, therefore during the weathering process no new artifacts such as cracks or stains appear. Our method incorporates two notable advantages. First, in most cases it does not require user interaction or assistance, since the weathered and un-weathered regions of the input texture are automatically detected. Second, new structures within the texture can be created over time, in addition to smooth color variations.

**Texture analysis and editing.** User interaction is often applied to detect weathered parts in a texture and control various material properties. In AppWand [Pellacini and Lawrence 2007], a time-varying sequence of weathered images can be edited, such that weathered parts are edited separately from non-weathered parts. In AppGen [Dong et al. 2011], a single image is analyzed with some user assistance to model the diffuse map, shading map and material reflectance, which may be different for weathered and non-weathered regions. Lu et al. [2009] used the similarity between texture patches to detect dominant texture elements (texels) without user interaction. They use the distribution of texels on a diffusion distance manifold to determine whether a texel is a part of the dominant texture. In our work, we look at a close neighborhood of the patch to determine its weathering degree. Recently, Dekel et al. [2015] presented a method for correcting non-local variations between repeated structures in an image. While not strictly related to weathering, their work can be seen as complementary to ours, as it aligns the position of repeated elements in the texture while maintaining the original content of each element. On

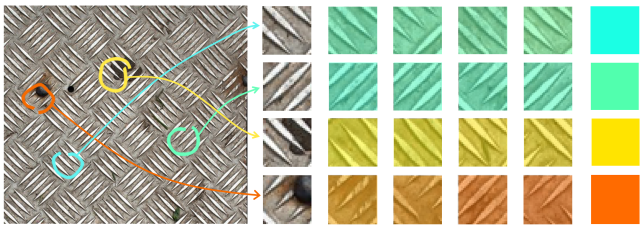



**Figure 3:** *Generation of the age map. Four patches (b) are taken as example from the input texture (a). Distances to four nearest neighbors (c) are computed and conveyed by the overlaying colors. For each patch the mean distance is computed (d).*

the other hand, our de-weathering process preserves the position of repeated elements, and adjusts the contents of each individual element as necessary.

## 3 Overview

An overview of our method is illustrated in Figure 2. Given an input image of a weathered texture (shown in the top row of the red frame), our method first computes an estimated age map of the texture based on the prevalence of similar patches in the texture. Using this age map, an intact texture is generated, and used to synthesize a series of textures emulating a weathering and de-weathering processes, yielding a time-varying texture. To produce de-weathered textures, the age map is manipulated to control an interpolation of the intact texture and the input texture. De-weathered textures are shown to the left of the figure above the manipulated age maps. Weathered textures are synthesized by extrapolating the differences between the input texture and the intact texture, and are shown to the right of the figure above the extrapolated difference maps.

Observing a weathered input texture, one can imagine the appearance of the texture in the past, before it was weathered, and in the future when it will become even more weathered. Intuitively, we recognize different weathering levels in various regions of the image, and commonly assume that in the past the entire texture consisted of the least weathered regions only, while in the future it will contain more and more regions with higher level of weathering. We propose an approach to time-varying texture synthesis that follows the same intuition. The degree of weathering is estimated at different regions of the input image, and used to increase or decrease the popularity of weathered patches. The series of weathering textures originating from the input texture appear plausible and realistic, as featured in Figure 1.

A critical step in our algorithm is the definition of a weathering *age map*, associating each pixel with a weathering degree. To estimate the weathering level, we apply a prevalence analysis of the texture patches and generate the age map associated with the input texture (see Figure 2). The weathering estimation is based on the observation that weathered regions usually present unique features, while non-weathered ones are generally more similar to each other. We elaborate on the prevalence analysis in Section 4.

Once the age map is computed, we synthesize the *intact texture*, a non-weathered texture which corresponds to the state of the texture at creation time (see Section 5). The intact texture is used as the target for the de-weathering process, which is an interpolation of the input texture and the intact texture. Conversely, the weathering algorithm is an open ended extrapolation process with an unknown target (see Section 6), in which the intact texture is used to obtain accurate pixel-wise knowledge of the weathering level of the input

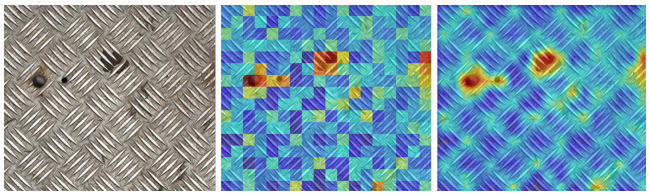


**Figure 4:** *Input texture (left), discrete age map (center) and continuous age map (right).*

texture. Note that this process mimics real-world behavior - while the history of an object is limited (up to the creation time), the future evolution is potentially infinite, at times to the point of such drastic changes that the object is left unrecognizable.

## 4 Age Map

A central component of our weathering algorithm is the generation of the age map. The age map associates every pixel of the image with an age value in the range $[0, 1]$, with 0 indicating intact pixels and 1 the most weathered pixels in the image. A prevalence analysis of image patches is performed to obtain the age map. Aiming to distinguish between weathered and non-weathered image patches, this analysis is based on the key observation that weathered patches are generally of a more unique appearance than their non-weathered counterparts. They tend to feature various colors and textures, whereas non-weathered patches appear rather commonly within the image.

Patch prevalence is computed in the following manner. The image is partitioned into $N \times N$ non-overlapping patches, which we call *source patches* ($N = 20$ in our experiments). We define the set of all possible patches of the same size, including overlapping patches in any position, as the *target patches*. For each source patch, we find its $K$ nearest neighbors within the target patches ($K = 10$ in our experiments). Then, we compute the mean distance from each patch to its nearest neighbors and use that value as the age of the patch. As illustrated in Figure 3, common patches are associated with lower values (cold colors) reflecting short distances to fellow patches, whereas rarer and more weathered patches are associated with higher values (hot colors). The final age map is smoothed with a Gaussian kernel to obtain continuous prevalence values, as shown in Figure 4.

**Distance between patches.** To compute the nearest neighbors of a patch, we need to define the distance between texture patches. We represent patches by the luminance and the direction of the most prominent gradient of each pixel. The color values can also be used, but we found that the luminance is discriminative enough in most cases, as significant changes in hue also derive changes in the luminance of a texture. The inclusion of gradient direction helps the detection of similar patterns (such as the patterns displayed in Figure 3), even where the overall color is slightly changed. For this, we use the following distance formulation:

$$D(P_i, P_j) = \frac{I(P_i, P_j)}{\max_{i',j'} I(P_{i'}, P_{j'})} + \frac{G(P_i, P_j)}{\max_{i',j'} G(P_{i'}, P_{j'})}, \quad (1)$$

where $I(P_i, P_j)$ is the difference in luminance between the patches, and $G(P_i, P_j)$ is the difference in the gradient between the patches.

**Nearest neighbors computation.** For each source patch $P_i$, we compute the distance $D(P_i, P)$ to every target patch $P$. This can be done quickly using convolutions of the image with the patch for

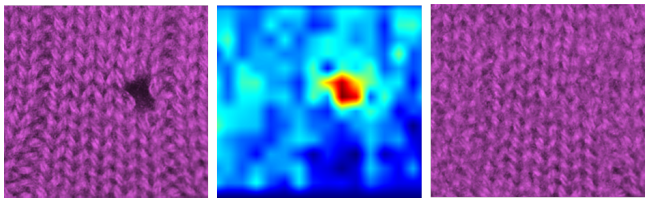


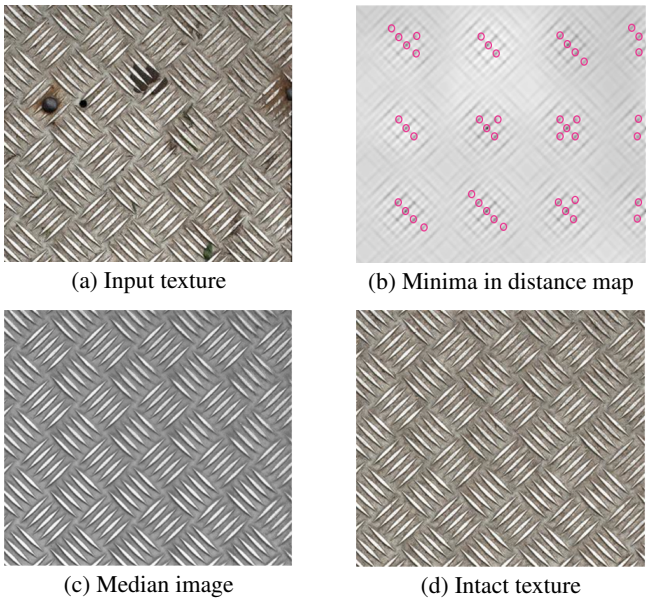
**Figure 5:** *Synthesis process of a stochastic intact texture. (left) input texture, (center) age map, (right) resulting intact texture.*

both the luminance and gradient channels. The result is a continuous distance map for every pixel in the image. Often, nearby pixels have similar distance values, which may result in the nearest neighbors of a patch originating in the same region of the image. Since we aim at measuring the prevalence of similar patches in the image, we select nearest neighbors that come from different regions of the image. This is done by discretizing the continuous distance map, or selecting a representative value for each of the $N \times N$ patches. Every pixel in the patch represents the distance from the source patch to a patch centered at that pixel. We select the minimum distance to any pixel-centered patches as the representative age of the entire patch. Then, the patches with the $K$ smallest age values are chosen as the nearest neighbors of the patch.

## 5 Intact Texture Generation

The generation of an intact texture is a necessary step for both weathering and de-weathering. It consists of two phases. First, we generate a template texture. This template is flexible and generated loosely, and does not have to maintain realism or continuity. Then, we use a texture transfer method, which takes the template texture and attempts to reconstruct it using a subset of patches from the original texture. This ensures that the final result is plausible and realistic. Our method supports two kinds of textures: stochastic textures, and structured textures which contain regularly repeated patterns. If a regularly repeated pattern is not detected, we treat the texture as stochastic. Note that we do not deal with semi-structured textures which contain patterns that repeat non-regularly.

**Tile detection.** To detect the tiling pattern, we find potential tile centers and check whether they are regularly repeated within the texture. First, we compute the distance between an arbitrary central patch and every other patch in the image. This can be done quickly using spatial convolution. Note that the size of the patch is not crucial as long as it covers a significant fraction of the image (we use about a third of the image). As can be seen in Figure 6b, the distance map tends to be continuous. The potential tile centers are local minima in the distance map. We detect them by marking $0.5\%$ of the pixels which have the lowest values, and finding the center of each detected region (marked in pink in Figure 6b). To check whether a tile is regularly repeated, we compute a convolution with a patch centered around it, and detect peak regions again. The peak regions of all tile centers are aggregated and only regions where many of the potential centers agree on are kept. In a structured texture with repetitive elements, many patches yield similar peak regions, while in a stochastic texture only a small number of potential tile centers are repeated. If many patches are repeated, we look for a regular grid of repeated tiles. To this end, we compute the offset vectors between all pairs of potential tile centers, and find the vectors which repeat the most. Assuming the distribution of tiles follows a regular grid structure, the vectors which are repeated most represent the direction of the grid, even when some of the tile centers are outliers. We find the two most common vectors, and define the grid according to these offsets.



(a) Input texture (b) Minima in distance map

(c) Median image (d) Intact texture

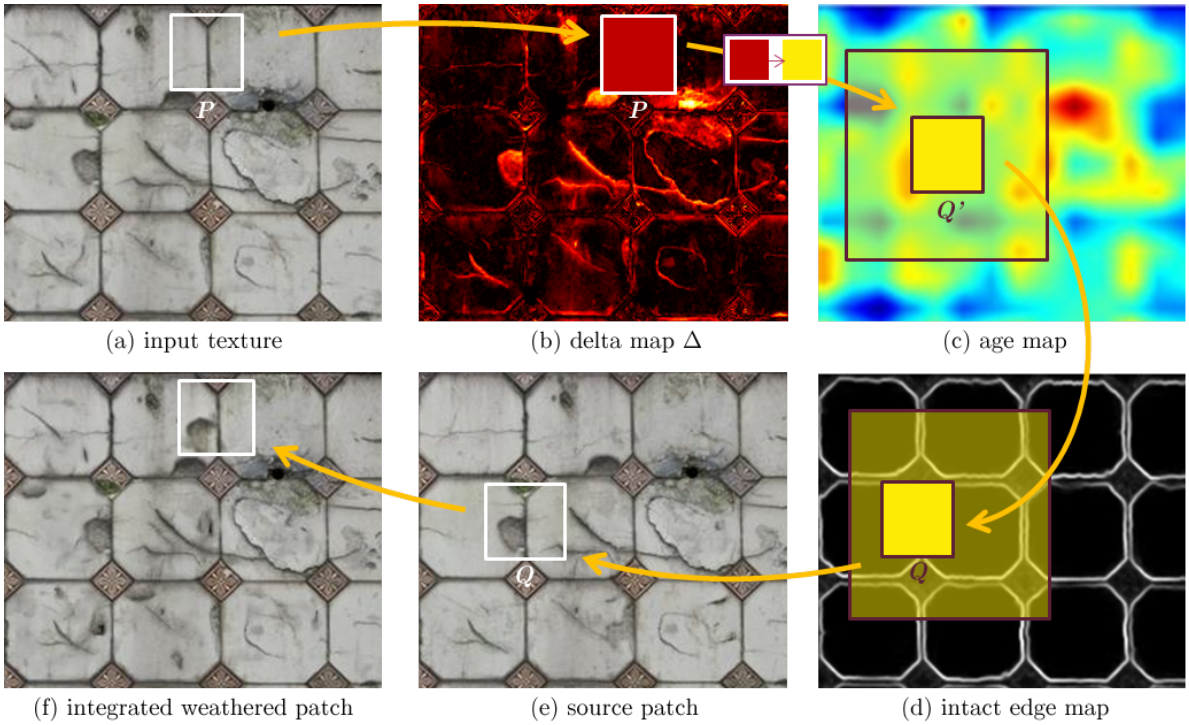
**Figure 6:** *Synthesis process of a structured intact texture. (a) input texture, (b) distance map between the input and a central patch with minima values marked in red, (c) output of the median computation, (d) intact texture.*

**Structured template generation.** Using the detected grid of tiles, we generate a template for the structured texture. Our assumption is that the majority of repeated pixels were not severely weathered, and weathered pixels are outliers. Therefore, weathered pixels can be discarded by taking the median of pixels with the same offset within the tile. We use this median image as the template texture. Corresponding pixels from different tiles are compared and consolidated to form the intact tile, by computing their median. See the result of the cross-tile median in Figure 6c. The result template texture incorporates the repetitive elements without the weathering effects, but may lose some of the stochastic details of the texture, thus reducing the realism and likeness to the original texture. Finally, we synthesize a realistic intact texture by reconstructing the template using patches from the original texture (see Figure 6d). For this, we use a variation of the quilting algorithm [Efros and Freeman 2001], where the distance between patches is computed according to the luminance and the gradient direction, similarly to (1). Since the template texture is clean, it is likely to be reconstructed using only non-weathered patches.

**Stochastic intact texture generation.** If regularly repeated patterns are not found (see second row of Figure 1), we generate a stochastic intact texture with a single texture transfer process. Based on the age map, we remove the most weathered regions ($20\%$ of the patches in our implementation), and apply texture transfer to re-synthesize the full image using only the remaining patches as source (see Figure 5). The result maintains the same geometric and textural characteristics of the original image, with weathered regions replaced by non-weathered regions with similar appearance. The texture transfer process maintains the realistic appearance of the texture.

## 6 Weathering and De-weathering

Weathered and de-weathered textures are produced in a similar way to the intact texture, by generating a rough template texture and reconstructing it using patches from the input texture. The templates




(a) input texture (b) delta map Δ (c) age map

(f) integrated weathered patch (e) source patch (d) intact edge map

**Figure 7:** *Weathered elements retrieval in the weathering process for a patch: (a) random patch $P$ is selected, (b) the $\Delta$ map where $P$ is colored according to its maximum value $m$, (c) a patch $Q'$ in the age map which matches the increased age, (d) selection of $Q$ from the neighborhood of $Q'$ according to the intact edge map, (e) $Q$ in the input texture, (f) $Q$ overlaid over $P$ in the synthesized texture.*

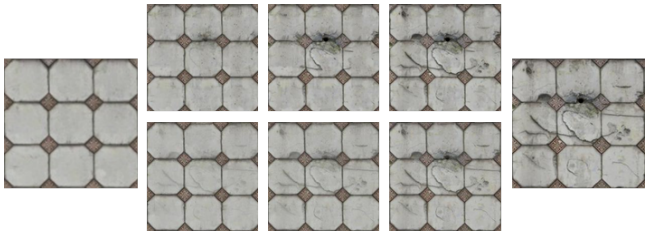


**Figure 8:** *Comparison between our de-weathering process (top row) and blending (bottom row) between the intact texture (left) and the input one (right).*

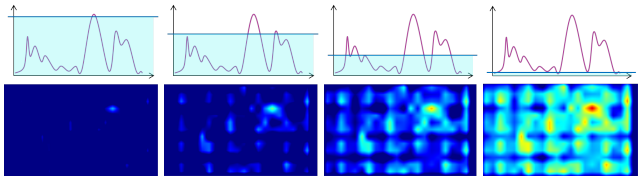



**Figure 9:** *Generation of the different age maps (bottom row). For every time value, the values below the threshold are discarded while those above are reduced (top row).*

are generated in a manner which preserves continuity from one time frame to the next, and yields a progression of textures, from the intact texture to the input texture, and further on to more weathered textures in an open ended manner, degrading further at each phase.

**De-weathering.** To generate a de-weathered template texture, we blend the given input texture and its corresponding intact texture, using the age map as a guide. This results in a series of inbetween template textures where weathering artifacts grow naturally, rather than fade in, as demonstrated in Figure 8. In the age map guided de-weathering process (top row), weathered artifacts are introduced gradually: severe artifacts develop over time while smaller artifacts, such as small scratches, only appear in later time steps. In comparison, blending linearly between the intact texture and the input texture (bottom row) results in a fade-in effect, producing an unnatural weathering effect.

Let $t = 0$ indicate the time of the intact texture, and $t = 1$ the time of the input texture. The synthesis of an age map for $0 < t < 1$ is based on the premise that highly weathered regions in the input image were typically created before less weathered parts, and had longer evolution through time. Therefore, their weathering stage at time $t$ is more progressed than less weathered regions, which might not be weathered at all at time $t$. In order to reflect this in the age map of time $t$, we use the following scheme to synthesize the template texture. Patches which are lightly weathered, for which the age is less than $1 - t$, presumably were not weathered at all at time $t$, hence their age is 0 and the patch is taken from the intact texture. Patches which are heavily weathered, for which the age is more than $1 - t$, are considered to have an age of $a - (1 - t)$ at time $t$, where $a$ is the patch age in the input texture. Such patches are blended accordingly between the intact texture and the input texture. This is illustrated in Figure 9.

Note that while the synthesized template textures monotonically evolve with the increment of the input time value, texture transferring results may not. This may cause unnatural flickering in the progressive view of different frames, for example when generating videos. In order to provide an incremental appearance of the grow-

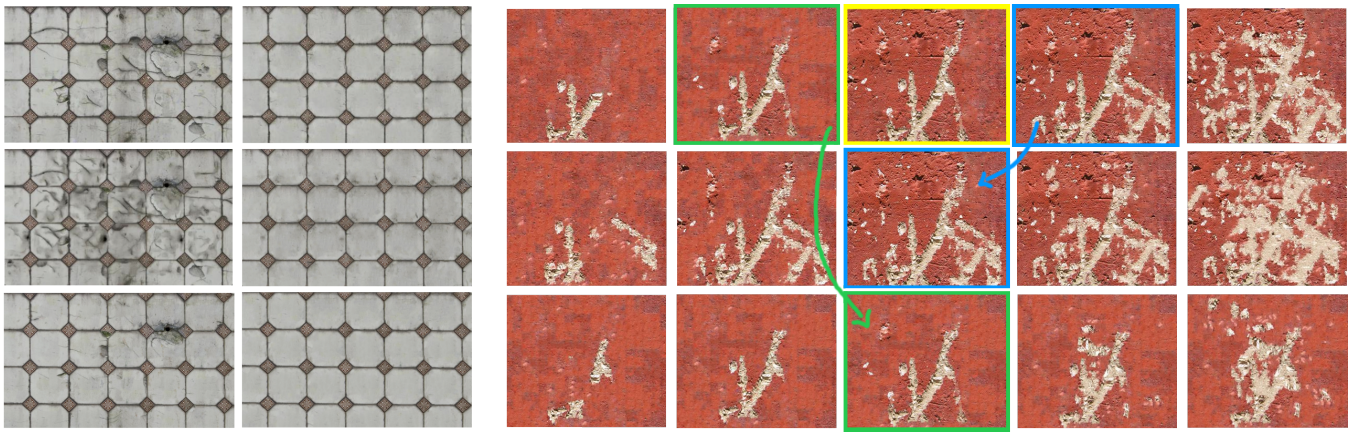


**Figure 10:** *Stability of our method. Left: Textures at different weathering stages (first column) and their corresponding intact textures (second column). Right: Weathering and de-weathering results of the input texture (top row), a synthesized weathered texture (middle row), and a synthesized de-weathered texture (bottom row).*

ing weathering effects, another element is introduced into the texture transfer algorithm: once a matching patch is found, it is compared with the same patch in the latest synthesized image (if any), their respective differences with the intact version of the patch are computed and for every pixel the most weathered value is chosen.

**Weathering process** The weathered versions of the texture are generated progressively, such that each texture $t_{i+1}$ is created by incrementally weathering the previous texture $t_i$. Each weathering step consists of selecting a random patch in the texture, estimating the age of the patch, picking a source patch with increased age in the input texture, and overlaying the source patch over the weathered texture in a way that preserves the structure of the texture.

The weathering process is illustrated in Figure 7. A patch $P$ is randomly picked in texture $t_i$ (a). The $\Delta$ map illustrated in (b) associates each pixel in $t_i$ with its distance from its corresponding pixel in the intact texture, i.e., $\Delta = \| t_i - intact \|$. The degree of weathering of the patch $\Delta(P)$ is assessed by its maximum value $m$. Note that accuracy is not crucial as some randomness is necessary to create a realistic weathering effect. Given age $m$, we look for a patch $Q'$ of age $m + \delta$ in the age map of the input texture (c). To enrich the variation, we assume that the weathering is anisotropic, and the patch is selected from four possible rotations of the image. The selected patch does not necessarily align correctly with its target position with respect to the structure, represented by a smoothed edge map of the intact texture (d). Thus, we search in the neighborhood of $Q'$ for a nearby patch $Q$ whose structure agrees with the structure of patch $P$. Once $Q$ is found, we overlay it over patch $P$ in texture $t_i$ to increase the weathering level of the patch. To do so, we define $\Delta_Q$ as the pixel-wise distance between the intact texture and the input texture at patch $Q$. The value of $\Delta_Q$ is normalized such that its range over the entire texture is between 0 and 1. It is used to weight the blend between patch $P$ and patch $Q$:

$$t_{i+1}(P) = t_i(P) \cdot (1 - \Delta_Q) + input(Q) \cdot \Delta_Q \qquad (2)$$

This way, only weathered pixels in patch $Q$ are synthesized into the new texture, while pixels that were close to the intact texture are taken from patch $P$, allowing the manifestation of previous weathering effects. Finally, we reconstruct the texture with patches from the input texture, in order to smoothly assimilate the new weathered parts into the texture.

## 7 Evaluation and applications

Results of our time-varying weathering textures are presented in Figure 11. The central column shows the input texture. Three steps of the de-weathering process are shown on the left, from the intact (left most column) to the input, and three incremental weathering steps are displayed on the right. All examples in this figure are fully-automatic, with no parameter tuning. The textures shown in (a), (b), and (c) present structured repetitive patterns, and the rest are stochastic textures. Each material reacts differently to weathering effects; our method learns the behavior of the material from the input image alone and mimics its weathering or increase in complexity, providing plausible results. Note the large variety of effects our method can mimic: change of color (b, d), ripped wallpaper (c), cracks and holes (d, k), torn wool (i), and more. Our method is also successful for textures that are affected by several combined effects (d, g, h). It can also be used for textures which do not contain signs of weathering but rather of entropy, or complexity of the texture; for example, sea shells (e), cracked glass (f) or knots in the wood (j). For these textures, the progression is not in time (wood knots do not appear over time) but have a plausible appearance which is useful for increasing or decreasing the entropy of the image.

Recall that for stochastic textures, intact textures are created by reconstructing the texture after discarding 20% of the patches. For some of the more weathered textures, the output texture is not completely intact since some weathered patches are still used (see (f), (h), and (k)). However, this does not hurt the realistic appearance of the de-weathered or weathered textures. In addition, this can be solved by adjusting the fraction of discarded patches. Another option is to run the process again with the intact texture as input, a process which requires a stable algorithm as discussed below.

**Stability.** The question arises whether our algorithm is *stable*, in the sense that it produces similar results when given similar textures at different time steps or weathering stages as input. To evaluate the stability of our method, we run two experiments, as shown on Figure 10. On the left, we show generated weathered and de-weathered versions of a structured input texture (left column), and compute an intact texture from each texture (second column). A stable algorithm is expected to yield similar intact textures. Indeed, our algorithm yields results with the same structure and texture for the three inputs. On the right, in the top row we show two generated de-weathering steps and two generated weathering steps of an input texture (marked in yellow). Then, we generate similar sequences

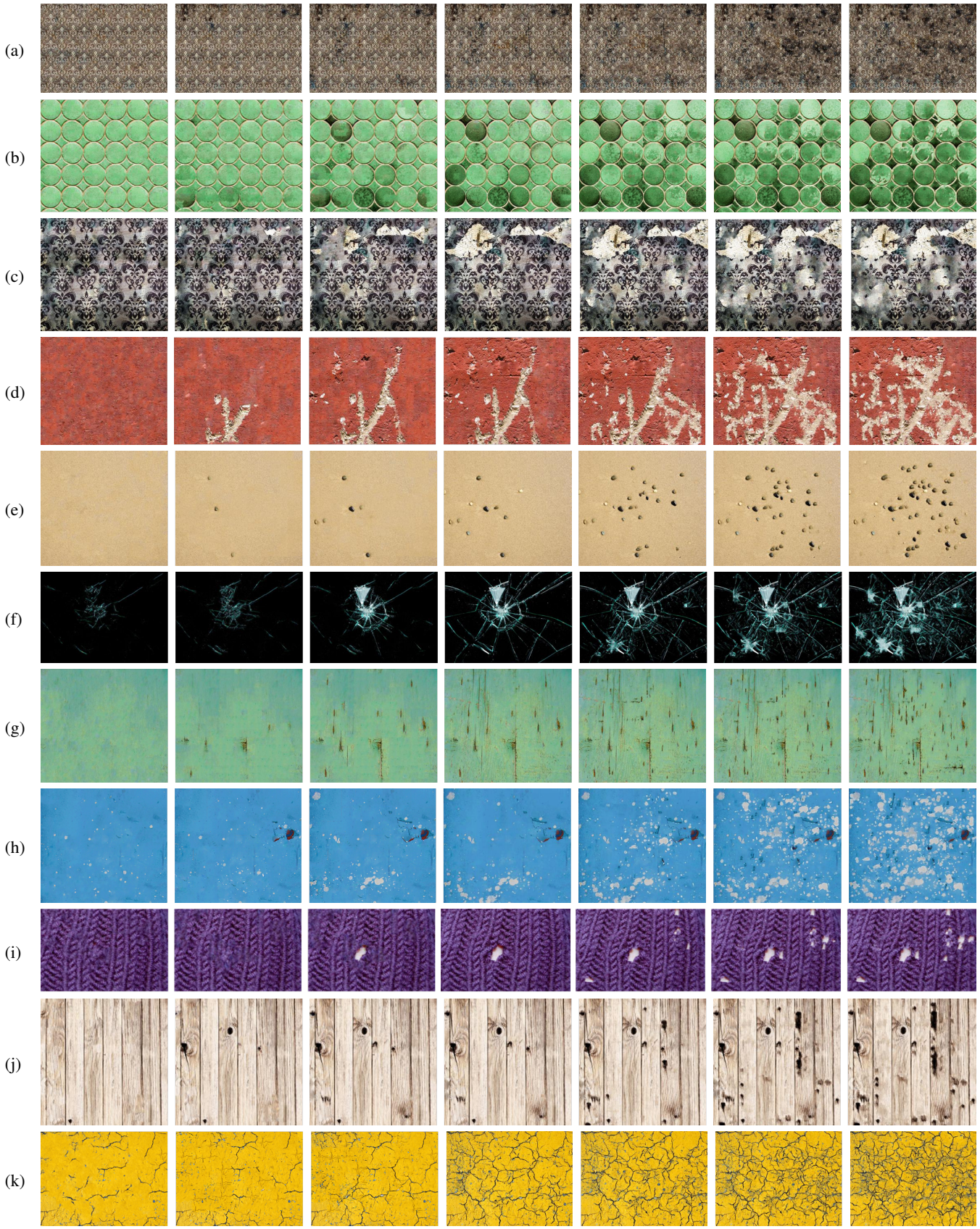



**Figure 11:** *Synthesized time-varying weathering. Input textures are in the central column, de-weathered textures to the left, and weathered textures to the right. The leftmost images are the intact textures.*

**Figure 12:** *Comparison between our method and Wang et al. For each row: (left) input image, (center) our result, (right) their result.*

from a de-weathered version (marked in green) and a weathered version (marked in blue). The sequence generated from a weathered texture is very similar to the original sequence, since no data is lost. In the sequence generated from a de-weathered texture, some elements are lost during the first de-weathering process, therefore the weathered versions are less rich with details than those in the original sequence. However, the structure and main characteristics of the texture remain similar in all three sequences.

**Comparison.** In Figure 8 we demonstrate the difference between our de-weathering process and a simple blending between the intact and the input image. As can be observed in the figure, the simple blending process does not have a realistic temporal appearance, since weathered regions do not grow, but gradually fade in during the process, resulting in an unnatural effect. In our de-weathering process, weathered elements grow monotonically, but non-linearly, and unlike the blending effect, each synthesized texture appears natural (see the time sequences in the supplementary video).

Many existing texture weathering methods define an *appearance manifold* [Wang et al. 2006; Xue et al. 2008] or *appearance map* [Bandeira and Walter 2009], a model which associate weathering degrees with gradually changing colors. As such, these methods require a smooth temporal variation in appearance. In contrast, our method extracts the weathering pattern from the input texture, and can introduce cracks, peeling paint and various other effects which are subject to discrete weathering events, rather than a gradual variation in color. As a result, many weathering effects which are supported by our method and presented in Figure 11 cannot be reproduced using the methods above.

For completeness, we compare our results with the results of Wang et al. [2006] for textures that contain smooth temporal variations. Figure 12 shows the input images on the left, our weathering outputs in the center, and the result of Wang et al. [2006] on the right. In both cases the image is manually segmented and only the areas of the manhole and the door are manipulated. The generation of the age map in our method and appearance manifold in [Wang et al. 2006] are assisted by user input. Our method clearly introduces new structures which preserve the look and feel of the material, such as new rusted regions in the manhole and new peeled regions in the door image. On the other hand, the results of [Wang et al. 2006] produce overall color changes which are less consistent with the material in the input texture.

To further emphasize the importance of structure synthesis, an extreme example is shown in Figure 13, where a binary texture is

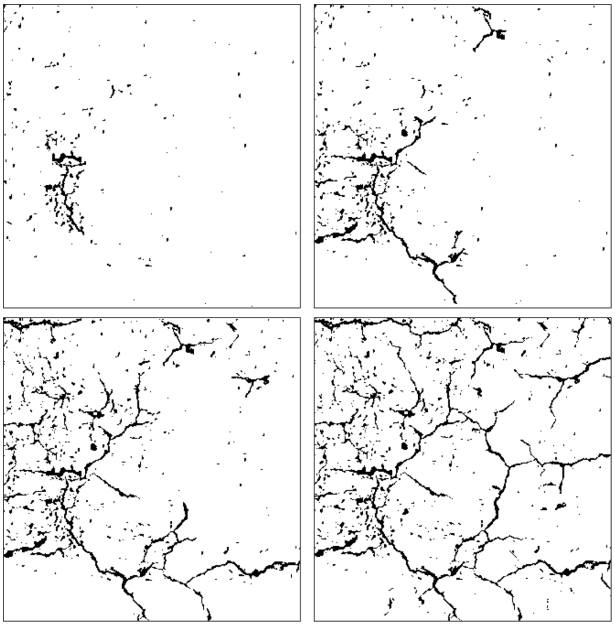


**Figure 13:** *Example of de-weathering a binary texture. The input texture is shown on the bottom right.*

taken as input. Our method can handle this kind of textures since it does not rely on a color model, but uses an image-based approach that introduces new structural parts.

**Applications.** In the following paragraphs, we discuss a few extensions and application scenarios that could benefit from our method. First, we apply the time-varying textures to a 3D scene (see Figure 14 and the accompanying video). The synthesized textures are used both for the color channel and displacement maps, affecting both colors and geometry. See time-varying sequences in the supplementary video.

**Layering.** Our basic weathering method presented above can be extended to deal with layered textures. Real world textures often contain regions made of the same material with different colors. Directly applying our technique on such non-stationary textures is likely to yield erroneous age maps, since the change of colors is interpreted as part of the weathering effect. For example, images with text typically consist of at least two color layers, one for the background and one for the foreground, while the geometry of the letters cannot be statistically separated in a reliable manner.

We develop a user assisted extension to our method to deal with textures that contain several layers. The texture is segmented into layers according to a basic color model, and the user can interactively improve the segmentation. We use this segmentation to compute the age map and generate an intact texture separately for each layer. Then, the age maps are combined into a unified age map, as well as the intact textures, and the weathering and de-weathering processes are computed without considering the layered structure. Figure 15 shows an image weathered with this layering method. From left to right it shows the input texture, age map, intact texture, and a weathered version of the input texture.

**Weathering transfer.** Another possible extension of our method is transferring the weathering behavior of one material to another. This method has two main applications: First, changing the weathering behavior of a material, mimicking that of another one. Second, weathering an almost intact texture, which contains little information about the weathering behavior, by learning the weathering

**Figure 14:** *Applying synthesized textures to 3D models. Textures are applied both on the color and the displacement channels. The models are rendered with the input textures (left), intact textures (center), and weathered textures (right).*

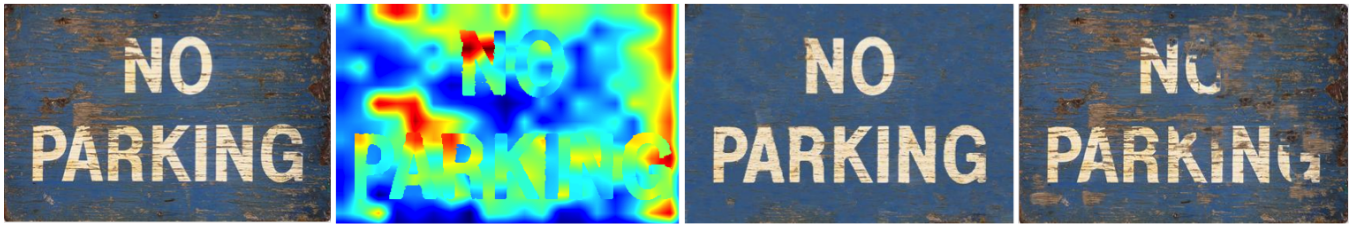



**Figure 15:** *A layered texture, where the computation of the age map and the intact texture are computed separately for each layer. From left to right: The input texture, the age map combined for the two layers, the synthesized intact texture, and a weathered texture.*

behavior of a second texture. We call the texture to be weathered *target texture* and the texture from which the weathering is learned *source texture*. Figure 17 shows two examples of weathering transfer in three different time steps. The top row shows the source texture and its computed $\Delta$ map for each time step. For each row below, the input target texture is on the left and three synthesized time steps are shown on the right. For each time step the weathered elements of the source are transferred to the target texture.

The weathered texture is produced using the same framework of generating a template texture and reconstructing it using patches from the input texture. First, an intact version of the source texture in synthesized. Then, we compute the $\Delta$ map of the source texture (shown in the top-right of each column of Figure 17) to isolate the weathered parts. The template texture $I$ is a combination of the target texture $T$ with the weathered elements from the source texture $S$, using the $\Delta$ map as a weighted mask:

$$I = T \cdot (1 - \Delta) + S \cdot \Delta, \tag{3}$$

This creates a template texture that contains elements from both textures and is not necessarily cohesive. Finally, we reconstruct the template texture using patches from the original target texture, so the weathered elements are consistent with the rest of the target texture. This process can also be applied to time-variant sequences: different levels of weathering of the source textures are synthesized and each of them is transferred onto the target texture.

**User-provided age map and intact texture.** The user can also provide the age map and intact texture for examples that are difficult to handle automatically. For example, for very weathered images, our algorithm cannot correctly distinguish the intact and weathered parts. Figure 18 shows two examples where the age map and intact texture are provided by the user, while the de-weathered and weathered versions are produced automatically. The example shown in the second row is taken from [Kaspar et al. 2015].

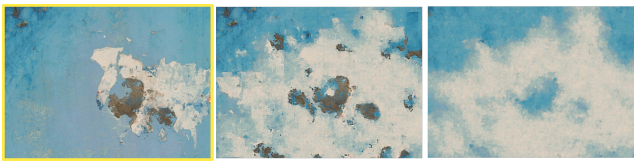


**Figure 16:** *Limitation : (left) original input image, (center) weathered version, (right) intact image computed from the central image as input. Some weathered regions are interpreted as intact.*

## 8 Limitations and Future Work

As our results show, our weathering algorithm is useful for a very large variety of patterns and textures. However, it is somewhat limited in the required weathering level of the input texture. For the algorithm to perform well without user interaction, the majority of pixels in the input texture should be intact. For a heavily weathered texture, the generated intact texture might contain a mixture of weathered regions and intact regions. An example of this is depicted in Figure 16. The input texture is displayed on the left, an automatically generated weathered version in the center, and on the right, an intact texture that was automatically generated taking the weathered version as input. Since most of the pixels in the weathered version are weathered, the generated intact texture is not entirely composed of truly intact pixels (blue pixels in the example) as expected.

On the other hand, the input texture must be somewhat weathered to produce pleasing results. Since we do not focus on a specific material or a specific weathering effect, and the generated textures are based on a single image, we cannot deduce how heavily weathered parts should look like for textures which are only lightly weathered. Existing weathered elements can be reproduced and enhanced, but new elements are not introduced. Such textures can still be weathered using our weathering transfer technique, by providing a weathered reference texture.

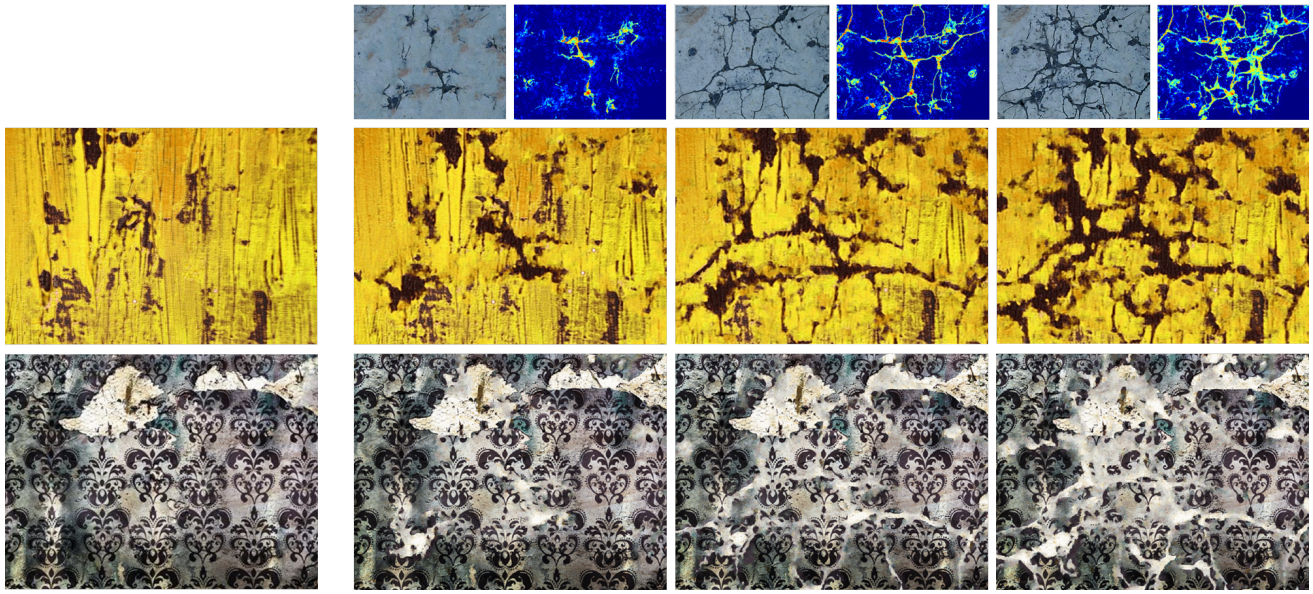


**Figure 17:** *Weathering transfer. For each row, the input texture (left) is weathered according to the weathering signature of the source texture in three different time values. The source textures are displayed on the top row along with their $\Delta$ maps.*

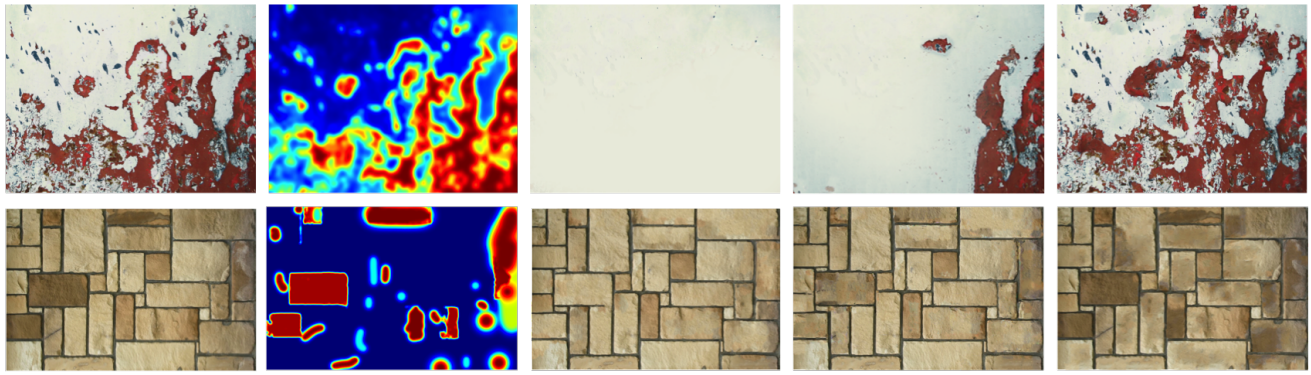


**Figure 18:** *User provided age map and intact texture. For each row, from left to right: user provided input texture, age map, and intact texture, followed by generated de-weathered and weathered versions. The input texture in the second row is taken from Kaspar et al.*

Integrating user assistance into the process may greatly benefit the issues discussed above. Our technique was designed to be automatic: while previous methods typically require the user to select weathered and non-weathered regions, our method requires no user input in most cases. Indeed, most of the results presented in the paper are fully automatic with the exception of Figures 12, 15, and 18. As we show in these figures, with some additional user assistance, a larger scope of weathered textures can be handled.

Our method deals well with stochastic, structureless textures, as well as structured textures with repeated patterns. However, some textures are not stochastic, yet their patterns are not repeated or they are repeated with some transformation: different scale, different rotation, etc. A mild example is the wool texture in Figure 11i: some distortion of the pattern appears in the generated intact texture, due to the stochastic nature of the texture generation. The effect may be more severe for textures with greater variations in the repeated pattern. Such textures cannot be generated with no concern of the structure, yet they require a different approach than the one we used for structured textures. This is indeed a challenging task, and an interesting direction for future work.

In our implementation, the variation of weathered patches is done using symmetry and $90^\circ$ rotations. Further exploration of the range of transformations and deformations that is applied on the weathered patches may benefit the variation in weathered textures. Geometric warping and photometric manipulations may reduce the amount of self-similarity of weathered textures and increase the realism of textures.

## 9 Conclusion

We present a novel technique to generate time-varying weathered textures. Unlike previous methods, our method does not require user interaction, and takes a single image as input with no prior knowledge of the weathering behavior of the material. The technique creates weathered textures using non-parametric texture synthesis techniques and patch manipulations, completely oblivious to the physical weathering phenomena. We analyze a single weathered exemplar and automatically induce enough valuable information to synthesize the intact texture and an age map, which together drive the synthesis of the weathered and de-weathered textures. This universal approach allows our method to produce pleasing and

realistic results for a large variety of materials and weathering effects, from simple color changes to elaborate cracks and peeling walls. In addition, it can be used to increase or decrease the entropy of textures which contain non-temporal artifacts such as clutter or wood knots. We also provide an extension for multi-layer textures, and develop a method for weathering transfer from one weathered texture to another.

## Acknowledgments

The authors thank Noa Fish and Su Xue for providing comments of early versions of the paper, and the anonymous reviewers for their suggestions. This work was supported by the Israel Science Foundation (ISF).

## References

AOKI, K., DONG, N. H., KANEKO, T., AND KURIYAMA, S. 2004. Physically based simulation of cracks on drying 3d solids. In *Computer Graphics International, 2004. Proceedings*, IEEE, 357–364.

BANDEIRA, D., AND WALTER, M. 2009. Synthesis and transfer of time-variant material appearance on images. In *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on*, IEEE, 32–39.

BECKET, W., AND BADLER, N. I. 1990. Imperfection for realistic image synthesis. *The Journal of Visualization and Computer Animation 1*, 1, 26–32.

BOSCH, C., LAFFONT, P.-Y., RUSHMEIER, H., DORSEY, J., AND DRETTAKIS, G. 2011. Image-guided weathering: A new approach applied to flow phenomena. *ACM Transactions on Graphics 30*, 3.

CHEN, Y., XIA, L., WONG, T.-T., TONG, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2005. Visual simulation of weathering by $\gamma$-ton tracing. In *ACM Transactions on Graphics (TOG)*, vol. 24, ACM, 1127–1133.

DEKEL, T., MICHAELI, T., IRANI, M., AND FREEMAN, W. T. 2015. Revealing and modifying non-local variations in a single image. *ACM Transactions on Graphics (TOG) 34*, 6, 227.

DESBENOIT, B., GALIN, E., AND AKKOUCHE, S. 2004. Simulating and modeling lichen growth. In *Computer Graphics Forum*, vol. 23, Wiley Online Library, 341–350.

DOLLÁR, P. Piotr's Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

DONG, Y., TONG, X., PELLACINI, F., AND GUO, B. 2011. Appgen: interactive material modeling from a single image. In *ACM Transactions on Graphics (TOG)*, vol. 30, ACM, 146.

DORSEY, J., AND HANRAHAN, P. 2006. Modeling and rendering of metallic patinas. In *ACM SIGGRAPH 2006 Courses*, ACM, 2.

DORSEY, J., PEDERSEN, H. K., AND HANRAHAN, P. 2005. Flow and changes in appearance. In *ACM SIGGRAPH 2005 Courses*, ACM, 3.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 341–346.

GEORGHIADES, A. S., LU, J., XU, C., DORSEY, J., AND RUSHMEIER, H. 2005. Observing and transferring material histories. *Technical Report 1329*.

HSU, S.-C., AND WONG, T.-T. 1995. Simulating dust accumulation. *Computer Graphics and Applications, IEEE 15*, 1, 18–22.

IBEN, H. N., AND O'BRIEN, J. F. 2009. Generating surface crack patterns. *Graphical Models 71*, 6, 198–208.

KASPAR, A., NEUBERT, B., LISCHINSKI, D., PAULY, M., AND KOPF, J. 2015. Self tuning texture optimization. In *Computer Graphics Forum*, vol. 34, Wiley Online Library, 349–359.

LU, J., GEORGHIADES, A. S., GLASER, A., WU, H., WEI, L.-Y., GUO, B., DORSEY, J., AND RUSHMEIER, H. 2007. Context-aware textures. *ACM Transactions on Graphics (TOG) 26*, 1, 3.

LU, J., DORSEY, J., AND RUSHMEIER, H. 2009. Dominant texture and diffusion distance manifolds. In *Computer Graphics Forum*, vol. 28, Wiley Online Library, 667–676.

MERILLOU, S., DISCHLER, J.-M., AND GHAZANFARPOUR, D. 2001. Corrosion: simulating and rendering. In *Graphics interface*, vol. 2001, 167–174.

MUKAI, N., SAKAGUCHI, Y., SHIGEOKA, H., AND KOSUGI, M. 2003. A method for generating texture images used on landscape simulation. MODSIM.

PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 2002. The simulation of paint cracking and peeling. In *Proceedings of Graphics Interface*, Canadian Human-Computer Communications Society, 10.

PELLACINI, F., AND LAWRENCE, J. 2007. Appwand: editing measured materials using appearance-driven optimization. In *ACM Transactions on Graphics (TOG)*, vol. 26, ACM, 54.

VALETTE, G., PRÉVOST, S., LUCAS, L., AND LÉONARD, J. 2008. A dynamic model of cracks development based on a 3d discrete shrinkage volume propagation. In *Computer Graphics Forum*, vol. 27, Wiley Online Library, 47–62.

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. In *ACM Transactions on Graphics (TOG)*, vol. 25, ACM, 754–761.

WONG, T.-T., NG, W.-Y., AND HENG, P.-A. 1997. A geometry dependent texture generation framework for simulating surface imperfections. In *Rendering Techniques' 97*. Springer, 139–150.

XUE, S., WANG, J., TONG, X., DAI, Q., AND GUO, B. 2008. Image-based material weathering. In *Computer Graphics Forum*, vol. 27, Wiley Online Library, 617–626.

XUE, S., DORSEY, J., AND RUSHMEIER, H. 2011. Stone weathering in a photograph. In *Computer Graphics Forum*, vol. 30, Wiley Online Library, 1189–1196.