Guided Image Weathering using Image-to-Image Translation

Li-Yu Chen
National Taiwan University
jcly.rikiu@gmail.com

I-Chao Shen
The University of Tokyo
ichao.shen@ui.is.s.u-tokyo.ac.jp

Bing-Yu Chen
National Taiwan University
robin@ntu.edu.tw

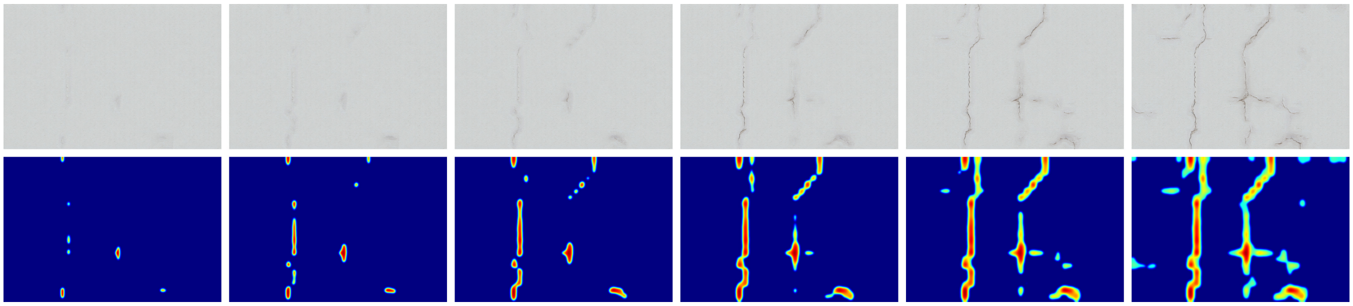


Figure 1: Weathering sequence guided by the age map (shown in the 2nd row). In an age map, the cold color (blue) indicates the less weathered region and the hot color (red) indicates the more weathered region. As the age value increases, the synthesized texture has more weathered effects on it.

ABSTRACT

In this paper, we present a guided image weathering method that allows the user to generate the weathering process. The core of our method is a three-step method to generate textures at different time steps of the weathering process. The input texture is analyzed first to obtain the weathering degree (age map) for each pixel, then we train a conditional adversarial network to generate texture patches with diverse weathering effects. Once the training is finished, new weathering results can be generated by manipulating the age map, such as automatic interpolation and manually modified by the user.

CCS CONCEPTS

• **Computing methodologies** → **Image manipulation**; **Graphics systems and interfaces**.

KEYWORDS

image weathering, image to image translation

ACM Reference Format:
Li-Yu Chen, I-Chao Shen, and Bing-Yu Chen. 2021. Guided Image Weathering using Image-to-Image Translation. In *SIGGRAPH Asia 2021 Technical Communications (SA '21 Technical Communications), December 14–17, 2021, Tokyo, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3478512.3488603

1 INTRODUCTION

The realistic texture is used by a wide range of applications: virtual reality, game development, digital visual effects, etc. The quality of textures can impact a lot, with more realistic ones often bring more immersion for the user experience. Thus, capturing realistic texture from the appearance of natural material becomes a major research area in computer graphics.

However, the appearance of natural material is not always the same, for example, in the area with temperature variation near freeze point, water freezes into ice and melts to water again, the expansion due to volume changes between two states gives enough pressure causing rock cracking. Other examples like paint peeling, metal rusting are also common phenomena around our daily life. The process of the natural environment influencing materials is known as weathering effect.

Weathering effect can be synthesized by simulating the physical process. The difficult part is that different materials usually have their type of weathering process, and constructing a simulation for each material may also not be an effective way. This makes weathering effect synthesis a challenging task. By observing the texture, we can find that weathered pixels often cover only a small part of the entire texture. That means calculating pixel distance inside texture gives a rough guess of weathering degree (age map) of the texture, then we can use the mapping as guidance to generate new textures.

In this paper, we develop a three-step method for generating textures at different time steps of the weathering process. Firstly, we analyze the input texture and predict the age value for each pixel [Bellini et al. 2016]. Secondly, we randomly crop pair patches of age map and texture, and deal with them as an image-to-image translation problem, which is achieved by training a conditional adversarial network [Isola et al. 2017]. Lastly, by manipulating (interpolating, extrapolating) the age map, we synthesize textures in different time steps of the weathering process.

De-weathering and weathering are two major applications of our method. We manipulate the age map step by step to create the weathering process and then generate textures by model inference. However, since we take the age map as the input of the network,

SA '21 Technical Communications, December 14–17, 2021, Tokyo, Japan

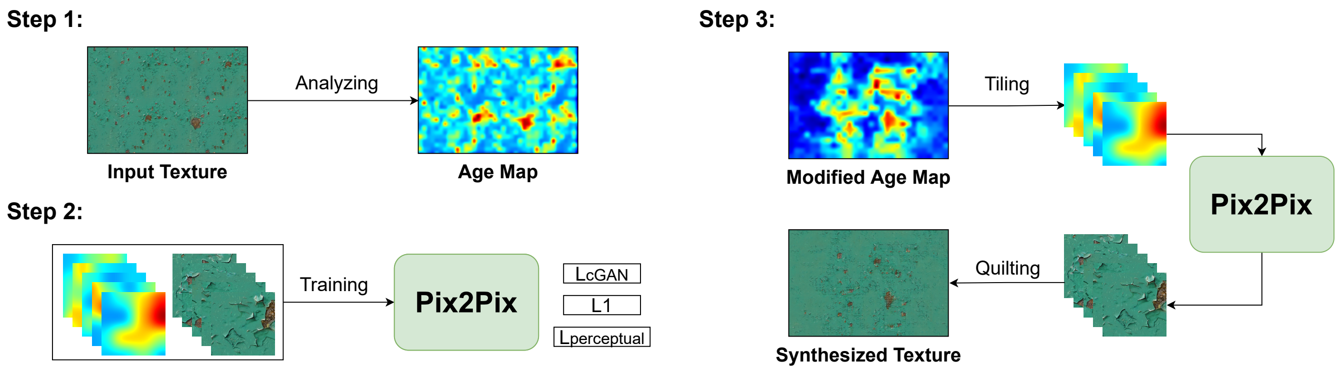


Figure 2: An overview of our 3-step method. Step 1 and 2 are the preprocessing stages and step 3 is for practical application.

the quality of the result is sensitive to the age map value. For some highly local changes, losing continuity could result in artifacts. We address this problem by fitting the age map to a B-spline surface, so we can manipulate the age map without introducing the discontinuity.

2 RELATED WORK

Time-varying weathering focuses on the weathering process of material over time. Performing physics simulations can obtain accurate results, while different materials need different simulation setups and are often time-consuming.

The more efficient way is by observing the input texture and synthesizing a similar texture with close distance in the image space, instead of dealing with the physical properties. Xue et al. [2011] consider smoothing, roughening, and silhouette erosion as primary effects of stone weathering, using image processing method to visually simulate the weathering progress of stone. Iizuka et al. [2016] use graph cut-based optimization to distinguish weathered regions in the texture, and perform patch-based synthesis with the weathering exemplar. Bellini et al. [2016] introduce an approach by analyzing the distance of patches in the input texture. This technique can produce an age map indicating the weathered-level (age) of each pixel in the input texture.

Image-to-image translation is the research about transferring images from a source domain to a target domain. To be more specific, the task is trying to synthesize a texture that matches the style in the target domain without changing the content of the image in the source domain.

Pix2Pix [Isola et al. 2017] uses U-Net structure [Ronneberger et al. 2015] as a generator and a patch GAN discriminator [Li and Wand 2016], to train a conditional adversarial network. Applications include semantic labeling, image coloring, day-night style transfer, etc.

Our method takes advantage of Pix2Pix [Isola et al. 2017], learning how to transfer the age map [Bellini et al. 2016] into texture, and with this mapping, we can create desired applications by manipulating the age map to guide texture synthesis.

3 METHOD

Our method can be split into 3 steps and is illustrated in Figure 2.

At first, we analyze the input texture and generate an age map that represents weathering age level of each pixel. Then the age map and input texture are randomly cropped into patches. We use these patches as training data to train a conditional GAN model. Finally, we can modify the age map and use the model to synthesize texture related to the age map.

The textures we use as dataset can be downloaded from the site. https://www.wildtextures.com/

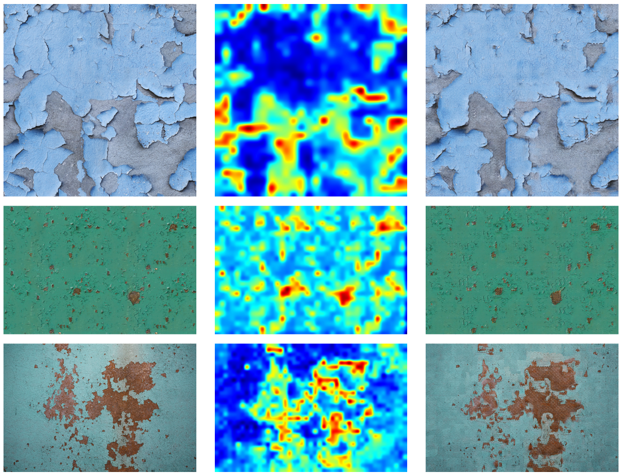
3.1 Age Map

Age map has an important role in our method, we follow a slightly modified algorithm from Bellini et al. [2016] for generating age map. The value of an age map is in the range [0, 1], and 0 means intact pixels and 1 means the most weathered pixels. In the step 1 of Figure 2, we visualize the age map by heat map, the area with normal blue paint has a low age value (blue), and the peeling area has a high age value (red).

To explain the algorithm, in short, the input texture is partitioned into $N \times N$ non-overlapping source patches ($N = 40$ in our experiments). Then we apply these patches to the template matching function in OpenCV, the output of this function can be seen as the distance between all possible target patches (overlapping) in the texture. For each source patch, we calculate the mean distance to K nearest neighbor target patches (K is 10% of patches in our experiments) and use the distance as the age value of the pixels in the source patch. Finally, we apply Gaussian blur on the age map to get a continuous one. More details can be found in Bellini et al. [2016]

3.2 Conditional Adversarial Networks

For each pair of age map and texture, we train a model separately. Generating texture from an age map can be seen as an image-to-image translation problem. Here, the age map is the condition and texture is the target output, we use the Pix2Pix model [Isola et al. 2017] with U-Net 256 [Ronneberger et al. 2015] as generator and patch GAN discriminator. Instead of using the entire age map directly, we randomly crop size $M \times M$ of patches ($M = 128$ in our experiments) from the age map, and upscale them to meet the input size of U-Net 256. Also, to increase the randomness, we add a



(a) Input texture (b) Age Map (c) Reconstructed

Figure 3: Texture reconstruction examples: (a) input texture, (b) age map and (c) reconstructed texture. The reconstructed textures are not identically the same as the input textures but still preserve the weathered effect in the same region.

Gaussian noise into another channel. The age map channel and the noise channel are concatenated together as the model input.

Pix2Pix [Isola et al. 2017] has two loss functions, GAN loss is for adversarial training and L1 loss is to accelerate the convergence of the training model. Here, we use the RGB color distance between real and fake images as an L1 loss. In addition to the original losses, we add a perceptual loss proposed by Johnson et al. [2016] into our objective function. The perceptual loss compares the L1 difference of gram matrix between feature maps through pre-trained VGG16. Textures generated by the model with perceptual loss have more random details, which is more realistic.

Our final objective function is

$$\arg\min_{G}\max_{D} \mathcal{L}_{cGAN}(G, D) + \mathcal{L}_{L1}(G) + \mathcal{L}_{Perceptual}(G) \quad (1)$$

3.3 Texture Synthesis

Since we train our conditional GAN model by patches, to synthesize a single texture, we have to crop the age map into tiling patches and merge the inference output of them into a texture. Simply tiling the output patches together produces a texture with the salient seam between patch borders. To eliminate this artificial effect, we crop the age map into overlapping tiling patches, and then apply the quilting algorithm [Efros and Freeman 2001] to find the min-cut edge to merge two patches seamlessly.

Reconstructing the texture from its age map is not perfect, since the age map certainly loses some information. However, we can see examples of reconstructed textures in Figure 3. The powerful conditional GAN model still captures the appearance of the input texture without losing lots of details.

4 APPLICATIONS

Once the training is finished, we can exploit the model to synthesize textures. For example, intact texture (the least weathered texture) can be obtained by filling the entire age map with 0 and synthesizing the texture from it. The intact texture can also be useful (as the least weathered reference) for later processing.

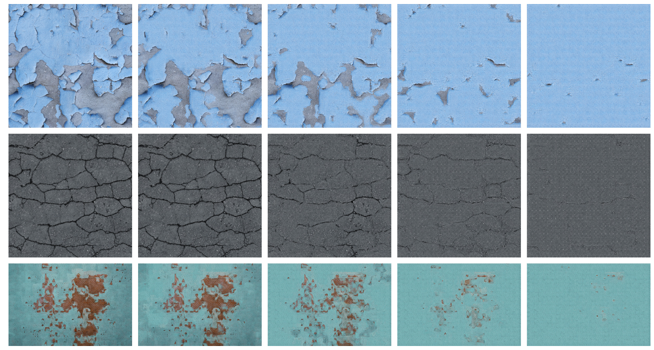


Figure 4: De-weathering results. Start from leftmost: input texture, and to the rightmost is closer to intact texture. Our method handles well on many weathering effects such as paint peeling or asphalt cracking.

Also, users can modify the age map directly and using it to synthesize texture. We introduce a simple interface that enables users to paint on the age map, with basic functions like increasing or decreasing the age value and changing the size of the stroke.

In addition to direct user modification on the age map, we can generate the weathering process automatically by interpolating between two age maps. More details are in the following sections.

4.1 De-weathering process

The age map of the de-weathering process can be generated by interpolating the age map of input texture and the intact texture. A simple way used in Bellini et al. [2016] is decreasing a certain value of the entire age map at each time step. If we want to interpolate from input texture to intact texture in N steps, since the age value is in the range [0, 1], this can be done by using the threshold $\frac{i}{N}$ at step i. The value below the threshold is set to 0, and above is reduced by $\frac{i}{N}$.

The meaning of this method is that we suppose the less weathered pixels appear at a later step, while the more weathered pixels appear in the early time step. However, in our experiments we find that in some cases, there is no medium state, so medium age value transfers into a mixture of a non-weathered and weathered pixel, a direct solution is only set the age value below the threshold to 0 and not to reduce the age value above the threshold.

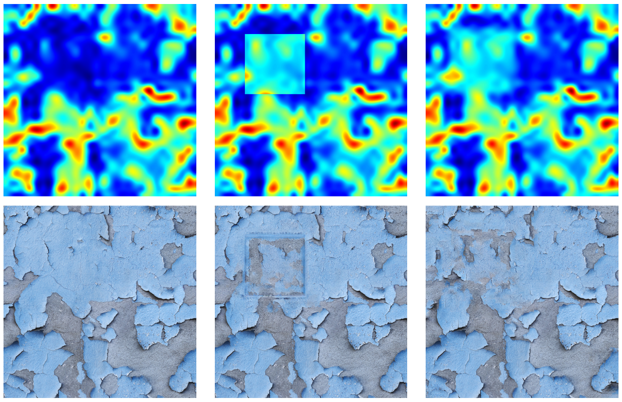
For each pixel p in an age map:

$$\text{age map}(p) = 0, \text{ if age map}(p) < \frac{i}{N}$$

To increase the reconstruction quality, at the tiling stage, if a patch has the exact same age map as the input, we use the patch from the input texture directly, instead of the synthesized patch of the model.

4.2 Weathering process

Weathering process has more freedom since we do not need to reach a specific target texture. We can modify the age map arbitrarily to



(a) (b) (c)

Figure 5: Weathering results. (a) The origin texture and its age map. (b) Increasing the age map value directly leads to a synthesized texture with a significant discontinuous visual appearance. Instead, (c) increasing the control points of the B-spline surface preserves the continuity of the age map and the synthesized texture.

guide the texture synthesis. However, locally changing the age map value too much could generate bad texture, it turns out that the GAN model is sensitive to high gradients, discontinuous age map leads to significant artifacts.

A B-spline surface can preserve continuity. For an age map with resolution (1024×1024), we fit a B-spline surface with 32×32 control points, and both UV degrees are set to 3. Then, we modify the control points instead of the age map value directly, Figure 5 shows that the GAN model can get a better result if the age map remains continuous.

4.3 Delta Map Blending

Although we can modify the age map monotonically decreasing or increasing, the model inference output may not be monotonic, resulting in a flickering effect. To deal with this problem, we use the same approach as Bellini et al. [2016], by calculating the L1 distance of synthesized textures to the intact texture ($\Delta = ||texture - intact||$), which is called delta map.

The delta map provides a measurement of the weathered level on each pixel. For the de-weathering process, a larger time step t means closer to the intact texture, ideally, the delta map should be monotonically decreasing, therefore, we can blend textures with the delta map to make sure that the de-weathering process is monotonic. In detail, if the pixel p on delta map $t+1$ is greater than delta map t, then replace the pixel p of texture $t+1$ with texture t, and vice versa (weathering process).

For each pixel p in the texture:

$$\text{texture}_{t+1}(p) = \text{texture}_t(p), \text{ if } \Delta_{t+1}(p) > \Delta_t(p)$$

Figure 6 shows the de-weathering process. From left to right, patches should be closer to intact texture, however, some new cracks appear at the right-most patch, this problem can be avoided by blending with the delta map.

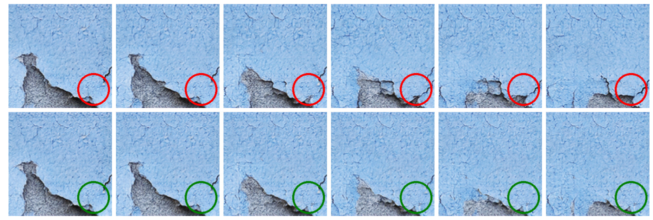


Figure 6: The top row is a sequence of synthesized textures in the de-weathering process, however, new cracks appear in the red circles. The bottom row is textures after applying delta map blending. We can see cracks are removed in the green circles.

5 CONCLUSION

We present a method to generate textures at different time steps of the weathering process. In specific, we combine two techniques, extending the usage of age map introduced by Bellini et al. [2016] and taking advantage of image-to-image translation model [Isola et al. 2017]. In our experiments, we show that the conditional GAN is able to synthesize texture with the guide of the age map, and we also make use of the B-spline surface to perform manipulation on the age map without introducing discontinuities.

ACKNOWLEDGMENTS

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST110-2634-F-002-026, and National Taiwan University. We thank to National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources. I-Chao Shen was supported by JSPS KAKENHI under Grant Number JP21F20075.

REFERENCES

Rachele Bellini, Yanir Kleiman, and Daniel Cohen-Or. 2016. Time-varying weathering in texture space. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

Alexei A Efros and William T Freeman. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 341–346.

Satoshi Iizuka, Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. 2016. Single image weathering via exemplar propagation. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 501–509.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5967–5976.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*. Springer, 694–711.

Chuan Li and Michael Wand. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*. Springer, 702–716.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 234–241.

Su Xue, Julie Dorsey, and Holly Rushmeier. 2011. Stone Weathering in a Photograph. *Computer Graphics Forum* 30, 4 (2011), 1189–1196.