

## 一种基于样本传播的图像风化方法

王玉铭<sup>1</sup>, 潘斌<sup>1</sup>, 郭小明<sup>1</sup>, 殷立<sup>1</sup>, 张曼琳<sup>1</sup>, 贾方利<sup>2</sup>

(1. 辽宁石油化工大学理学院, 辽宁抚顺 113001; 2. 辽宁石油化工大学计算机与通信工程学院, 辽宁抚顺 113001)

**摘要:** 对图像中的物体进行风化处理是一种提高模拟场景真实感的有效方法。利用一种图像处理方法, 实现了图像中物体表面的精细风化效果。首先, 利用径向基函数(RBF)方法计算风化程度。即使物体表面的颜色变化很大, 该算法也可以较精确地计算风化程度。然后, 使用图像分割算法来提取风化程度最深的区域作为“风化样本”。最后, 利用patchmatch图像修补算法对“风化样本”进行补全, 以此在图像上的待风化区域进行“风化样本”的无缝传播。结果表明, 此方法可以产生各种类型的风化效应。利用样本传播的方法可以更好地模拟风化图像。

**关键词:** 图像风化; 径向基函数; 图像分割; 图像修补; 样本传播

**中图分类号:** TP391.41

**文献标志码:** A

**doi:** 10.3969/j.issn.1672-6952.2021.01.014

## An Image Weathering Method Based on Exemplar Propagation

Wang Yuming<sup>1</sup>, Pan Bin<sup>1</sup>, Guo Xiaoming<sup>1</sup>, Yin Li<sup>1</sup>, Zhang Manlin<sup>1</sup>, Jia Fangli<sup>2</sup>

(1. College of Sciences, Liaoning Petrochemical University, Fushun Liaoning 113001, China;

2. School of Computer and Communication Engineering, Liaoning Petrochemical University, Fushun Liaoning 113001, China)

**Abstract:** In order to improve the realism of the simulated scene, weathering the objects of the image is an effective method. Using an image processing method to achieve the fine weathering effect of the objects surface in the image. First of all, use the radial basis function (RBF) method to calculate the weathering degree image. Even if the color of the surface of the object changed greatly, the algorithm could calculate the degree of weathering more accurately. Then, use the image segmentation algorithm to extract the most weathered region as the "weathering exemplar". Finally, the patchmatch image repair algorithm was used to complete the "weathering exemplar", so that the "weathering exemplar" could be propagated seamlessly in the weathering area of the image. The results show that the proposed method can generate various types of weathering effects. Using the method of exemplar propagation can better simulate the weathering image.

**Keywords:** Image weathering; RBF; Image segmentation; Image patching; Exemplar propagation

现实世界中大部分物体随着时间的推移, 其表面会根据材料属性和周围的环境而产生外观上的变化, 被称为风化作用。在计算机图形学中, 利用风化作用可以有效地提高场景的真实感。现有方法通常改变的是饱和度和反射度等材质属性, 实现落叶或者污渍等效果<sup>[1-2]</sup>。然而, 在现实世界中, 许多风化现象具有更加复杂详细的纹理上的变化。此外, 随着时间的推移, 风化的物体表面会具有更复杂的颜色分布。

目前, 有许多种基于物理模拟来产生风化效应的研究, 为特定的场景创造出逼真的风化效果, 例如流体模拟<sup>[3]</sup>、石像的风化侵蚀<sup>[4]</sup>、苔藓生长<sup>[5]</sup>等。但是, 这些方法都需要物体精确的几何形状和物理参数, 这些参数很难从单幅图像中估算出来。还有一些基于图像的物体表面风化算法。例如, S.Xue等<sup>[6]</sup>通过修改物体的几何外观, 对图像中的石头模拟出风化效果; 文献[7-8]研究了一种基于粒子模拟的方法, 在物体表面产生流体的侵蚀效果。这些方法

**收稿日期:** 2019-10-23 **修回日期:** 2019-12-20

**基金项目:** 国家自然科学基金项目(61602228、61572290); 辽宁省“兴辽英才计划”青年拔尖人才项目(XLYC1807266); 辽宁省自然科学基金项目(2015020041)。

**作者简介:** 王玉铭(1995-), 男, 硕士研究生, 从事计算机视觉、图像处理方面的研究; E-mail: wangym\_6317@foxmail.com。

**通信联系人:** 潘斌(1981-), 男, 博士, 副教授, 从事计算机视觉、机器学习、图像处理方面的研究; E-mail: panbin@lnpu.edu.cn。

通过限制目标物体的材料和风化作用来获得逼真的效果。

为了更逼真地模拟风化效果,J.Wang等<sup>[1]</sup>研究了一种可视化的模拟方法,从单一材料表面上不同点的BRDF(双向反射分布函数)中获得时变外观,构造出一种外观流形,这种外观流形近似于一种利用风化度生成的外观子空间,以此来计算风化程度的分布,并模拟材料表面的时变效果。D.Bandeira等<sup>[9]</sup>基于一种外观图像,将外观流形简化为二维特征空间。Y.Endo等<sup>[10]</sup>通过添加用于表示阴影变化的高频分量来细化外观图的显示结果。G.Hwang等<sup>[11]</sup>基于时间的外观流形模拟风化效应,并且用于风化视频的处理。

A.A.Efros等<sup>[12]</sup>提出了一种基于纹理补丁拼接的纹理合成技术,该算法对基于图像分割的拼接技术和能量优化等方面进行了改进<sup>[13]</sup>。关于图像补全技术,D.Simakov等<sup>[14]</sup>提出了一种基于全局优化的技术,可以达到更加细致的填充效果。文献[15—16]研究了一种名为PatchMatch的快速随机补丁搜索算法来加速算法运算。白宏阳等<sup>[17]</sup>提出一种利用加权矩阵补全图像的方法。R.Bellini等<sup>[18]</sup>提出了一种估算部分风化纹理的“年龄图”方法,该方法能够模拟合成风化效应在时变演化过程的序列。Y.Zhou等<sup>[19]</sup>提出一种自动分析和控制合成高度不均匀纹理的新方法。结果表明,当待补全区域相较采样区域较大时,结果很难令人满意,得到的效果并不理想,视觉感不真实。综合来看,风化效果的模拟可以大致看成一种图像补全问题,其中未风化区域是待修补区域,风化区域则是修补的补丁,也就是样本。

利用径向基函数(RBF)方法可以合理地计算风化区域的分布程度,模拟出具有丰富颜色变化的风化效果。从风化样本中取样,并根据风化程度和外观一致性将其传播到目标区域,以此来模拟外观特征。该算法可以更加快速地生成风化模拟,效果更加逼真。

## 1 图像风化

图像风化模拟流程如图1所示。

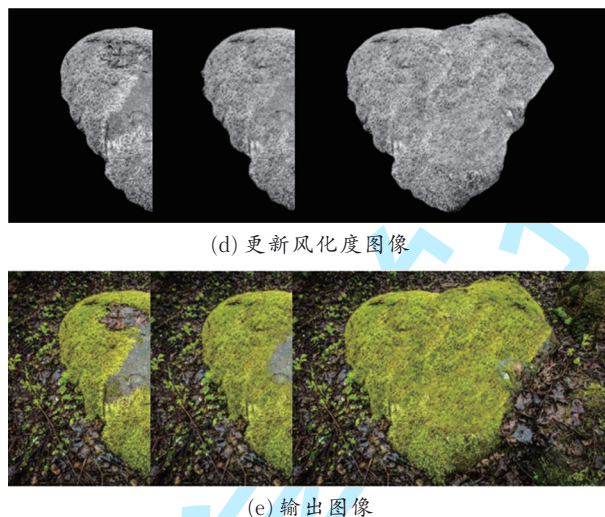
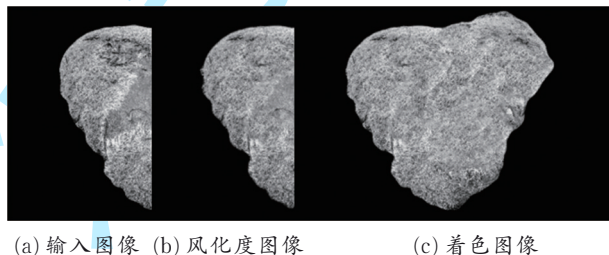


图1 图像风化模拟流程

图像风化基本流程如图2所示。

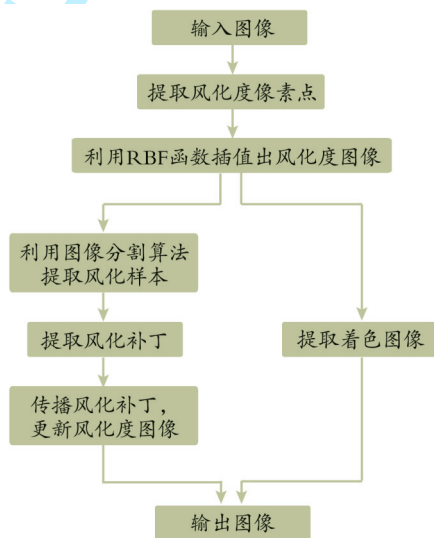


图2 图像风化基本流程

首先,使用GrabCut算法<sup>[20]</sup>或者图像编辑工具提取目标对象的像素点,并且指定提取点分别为风化程度最高和最低的点。具体方法,利用算法或者提取工具,人为选择提取风化程度最高与最低的一个点。将这些提取的像素点作为风化程度的一种基准,然后利用径向基函数的方法对这些像素进行函数插值,构建一个在高维特征空间中的风化度平滑函数。利用构造的非线性函数,可以计算出风化程度图像。最后,利用图像分割算法从高度风化的像素中提取出风化程度最高的区域。与此同时,基于风化程度图像提取着色图像,着色图像大致包含图像中物体的表面细节。当进行风化度图像更新时,从风化样本中提取风化补丁在风化度图像上传播。为保持外观效果的一致性,风化补丁由全局优化算法生成。最终,将风化效果与着色图像结合,得到输出图像。

### 1.1 基于 RBF 的风化度计算

利用 RBF 算法进行函数插值计算的方法已经广泛地应用于计算机图形学中,例如基于点数据的三维重建<sup>[21]</sup>、目标检测<sup>[22]</sup>、图像去噪<sup>[23]</sup>等。利用一种光滑函数表示高维特征空间中的风化度分布。这种算法可使图像的风化程度平稳变化,合理地计算具有广泛颜色分布变化的图像风化程度。因为指定像素为中心的 RBF 数值,通常只需要几十个或几百个像素值,所以这种算法的速度特别快。

首先要确定每个 RBF 的系数,计算每个像素的风化程度,以此作为 RBF 的线性组合。设  $\Omega$  为用户指定区域的一组像素。将径向基函数  $\phi(r) = e^{-r^2}$  赋给每个像素  $i \in \Omega$ ,并通过求解以下最小二乘问题来计算其系数:

$$E(\alpha) = \sum_{i \in \Omega} (d_i - \sum_{j \in \Omega} \alpha_j \phi(\|f_i - f_j\|))^2 \quad (1)$$

式中,  $f_i = (L_i/\sigma_a, a_i/\sigma_a, b_i/\sigma_a, x_i/\sigma_s, y_i/\sigma_s)$  为视觉特征和空间特征组成的特征向量,其中  $L_i$ 、 $a_i$ 、 $b_i$  为 Lab 颜色空间中的亮度和色度值,而  $x_i, y_i$  为归一化后的像素坐标,  $\sigma_a$  和  $\sigma_s$  为控制外观效应和空间局部性的参数;根据测试,令  $\sigma_a = 0.2, \sigma_s = 50, \alpha_j$  为未知系数;  $d_i$  为每个像素的风化度,其中风化程度最低的像素的  $d_i = 0.01$ ,最高的像素的  $d_i = 1.00$ ;  $f_j$  为像素  $j$  的特征向量。

利用系数  $\alpha$ , 计算每个像素  $p$  的风化度:  $D_p \in [0, 1]$ 。

$$D_p = \sum_{i \in \Omega} \alpha_i \phi(\|f_p - f_i\|) \quad (2)$$

式中,  $f_p$  为像素  $p$  的特征向量。

### 1.2 全局着色

利用补丁合成的方法模拟风化物体表面的外观特征效果。然而,将补丁直接合成到目标位置上,有时会影响下方物体的原有形状。因此,预先计算出物体表面的着色图像,然后再进行补丁合成。

根据物体表面的风化程度来计算整体着色。利用风化样本描述的二值分割结果来识别风化程度较少的区域,将风化程度较低的区域均匀划分为 20 个离散区间,并计算出每个区间  $k$  的平均亮度值  $\mu_k$ 。然后,将区间  $k$  中的每个像素  $p$  的亮度  $I_{p,k}^{\text{lum}}$  除以对应的  $\mu_k$  来计算全局着色  $S_{p,k}$ , 即  $S_{p,k} = I_{p,k}^{\text{lum}} / \mu_k$ 。其余的高度风化区域保持原有亮度,因为在模拟风化的过程中,这些区域的外观不发生变化。

### 1.3 风化效应模拟

利用风化补丁在待风化区域进行传播,以此来

模拟风化效果。这种方法与以前仅仅改变色度或反射率的方法不同。

首先,使用图像分割算法在高度风化的区域中识别并分割出风化程度最高的图像块,并标记为风化样本。然后,利用风化程度引导的全局能量优化算法来找到最佳的风化补丁。最后,将补丁和着色区域进行合成,以此来模拟风化效应。

1.3.1 风化样本 风化样本的定义包含最明显风化特征的矩形区域。给定风化图像,首先进行二值分割,以此来标记风化区域和未风化区域。最显著的风化像素通常会在图像中颜色密集的地方出现,利用风化度  $D$  定义代价函数  $E(L)$ :

$$E(L) = \sum_p -\lg U_p(D_p, L_p) + \lambda \sum_{p,q \in N} [L_p \neq L_q] V_p(D_p, D_q) \quad (3)$$

其中,  $U_p(D_p, L_p) = \begin{cases} D_p & L_p \in \text{“风化”} \\ 1 - D_p & L_p \in \text{“未风化”} \end{cases}$ 。

式中,  $L_p \in \{\text{“风化”, “未风化”}\}$ ;  $N$  为相邻像素组的集合;  $V_p$  为平滑项;  $\lambda$  为决定平滑项相对重要性的平滑系数;  $[ ]$  如果参数为正,则指示函数取值为 1, 否则为 0。

根据文献[24],使用  $V_p(D_p, D_q) = \exp(-\beta(D_p - D_q)^2 / \text{dist}(p, q))$ , 其中  $\text{dist}(p, q)$  是相邻像素的欧氏距离,  $\beta = (2\langle (D_p - D_q)^2 \rangle)^{-1}$  是一个常数,  $\langle \rangle$  表示平均值。对于所有的结果,  $\lambda$  取值都为 20。利用图像分割方法可以有效地解决能量最小化问题。然后,提取一个矩形,其中包含标记为“风化”的大多数像素。在系统中使用文献[25]算法进行实时计算矩形的尺寸为  $150 \times 150$  像素。提取的区域通常存在孔洞,会引起风化效应噪声。即使明显地提高风化度,孔洞内像素还是保持不变。为了避免这些问题,使用 PatchMatch<sup>[15]</sup>来填充孔洞。

1.3.2 基于全局优化的风化补丁搜索 在生成风化样本之后,将采集风化补丁分布到整个目标区域,也就是待风化区域。在这个过程中,标记为“风化”的像素将保持不变,也就是处在风化区域。为保证风化后效果具有较好的视觉纹理性,需要在输入图像  $I$  中处理从风化样本  $T$  中采样的风化补丁,具体步骤为:首先,从风化样本中随机抽取补丁,并与相邻补丁重叠,两块补丁重叠的面积为补丁面积的  $1/4$ ,这种方法类似于图像拼接技术<sup>[14]</sup>;然后,基于全局测量距离  $d(I, T)$ ,对修补程序进行迭代更新。风化样本传播如图 3 所示。为寻找输入图像的每个补丁中匹配度最高的风化像素块,将重叠像素



和相邻像素以及标记为“风化”的像素颜色度的平方差之和(SSD)最小化, $d(I, T)$ 定义为:

$$d(I, T) = \sum_{p \in I} \min_{q \in T} (K_p G_{\text{dat}}(p, q)) + (1 - K_p) G_{\text{nei}}(p, q) \quad (4)$$

式中, $G_{\text{dat}}$ 为风化像素颜色度的 SSD; $G_{\text{nei}}$ 为相邻像素与重叠像素之间颜色度的 SSD; $K_p$ 为校正风化度,“未风化”像素的风化度设置为 0,而“风化”像素的风化度不变。

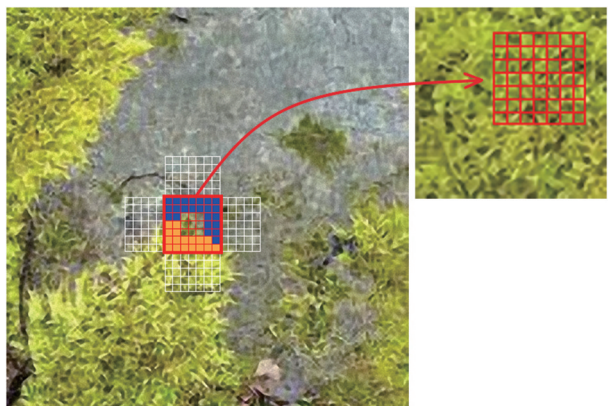


图 3 风化样本传播

测量距离 $d(I, T)$ 优先考虑高度风化区域的外观相似性,否则保持相邻补丁之间的视觉一致性。这种优化可以生成一种自然的风化纹理,既可以和风化区域无缝过渡,又能保持风化样本的外观特征。在实现过程中,会有两步由粗到细的细化。与此同时,会改变风化补丁的大小。首先,使用 $60 \times 60$ 像素搜索补丁以保留风化样本的整体结构。然后使用 $20 \times 20$ 像素更新补丁以达到更详细的特征效果。使用 PatchMatch<sup>[15]</sup>可以加速补丁搜索过程,修改偏移量以查找相邻补丁。在合成过程中,使用最小边界误差分割算法<sup>[16]</sup>来确定相邻重叠块之间的分割。

#### 1.4 风化效果渲染

利用全局着色和风化补丁,生成了最终图像 $I'$ ,然后将 Lab 颜色空间中的亮度值与着色值相乘。因为设定的风化效应出现在风化程度超过一定阈值 $\tau$ 的区域,所以在风化度 $D$ 大于阈值的像素中合成风

化纹理。计算最后的图像:

$$I'_{\text{lum}} = (D' Z_{\text{lum}} + (1 - D') I_{\text{lum}}) * S \quad (5)$$

$$I'_{\text{hue}} = D' Z_{\text{hue}} + (1 - D') I_{\text{hue}} \quad (6)$$

其中, $D' = \begin{cases} D & D > \tau \\ 0 & D \leq \tau \end{cases}$ 。

式中, $I$ 为输入图像; $Z$ 为对未风化区域处理后的风化纹理; $S$ 为着色图像; $*$ 为卷积乘积;下标 lum 表示亮度通道,下标 hue 表示 Lab 空间中的色度通道。与以往的方法不同,这种方法在视觉上保证了风化纹理效果的连续性。

#### 1.5 风化模拟

通过自动更新风化度图像的办法来模拟风化效果。采用添加平滑度图像技术来平滑地扩展风化区域<sup>[9]</sup>。风化度图像整体更新基于粒子模拟的方法<sup>[7]</sup>。利用粒子模拟方法可以轻易地实现模拟风化效果,具体的方法是将算法中流体染色中的粒子替换成风化度。当粒子穿过像素时,每个像素的风化程度将会增加。

## 2 结果分析

利用 C++ 实现了原型系统编写,并在配置为 i5-8400 型号 CPU、8 GB 内存和 GTX1060 显卡的电脑上运行。计算风化程度的平均时间约 0.1 s,风化样本的生成时间约 0.5 s,风化补丁的搜索时间约 3.0 s。

基于样本传播,在图像中的物体表面模拟出精细的风化效果,选取大量的风化图像进行模拟处理,并且与以往的算法进行了比较。模拟风化效果如图 4 所示。

不同算法的模拟风化效果图 5 所示。从图 5 可以看出,文献[9—10]算法生成的图像效果非常不自然,颜色十分趋近并且单一。本文算法与以往的算法不同,为了可以处理颜色差异较大的风化图像,利用 RBF 插值算法计算风化度图像,利用样本传播的方法模拟风化效果,合理地模拟了风化效应,并保留了风化区域的详细特征,效果更为理想。



原图

半风化

全风化

(a) 爬山虎生长

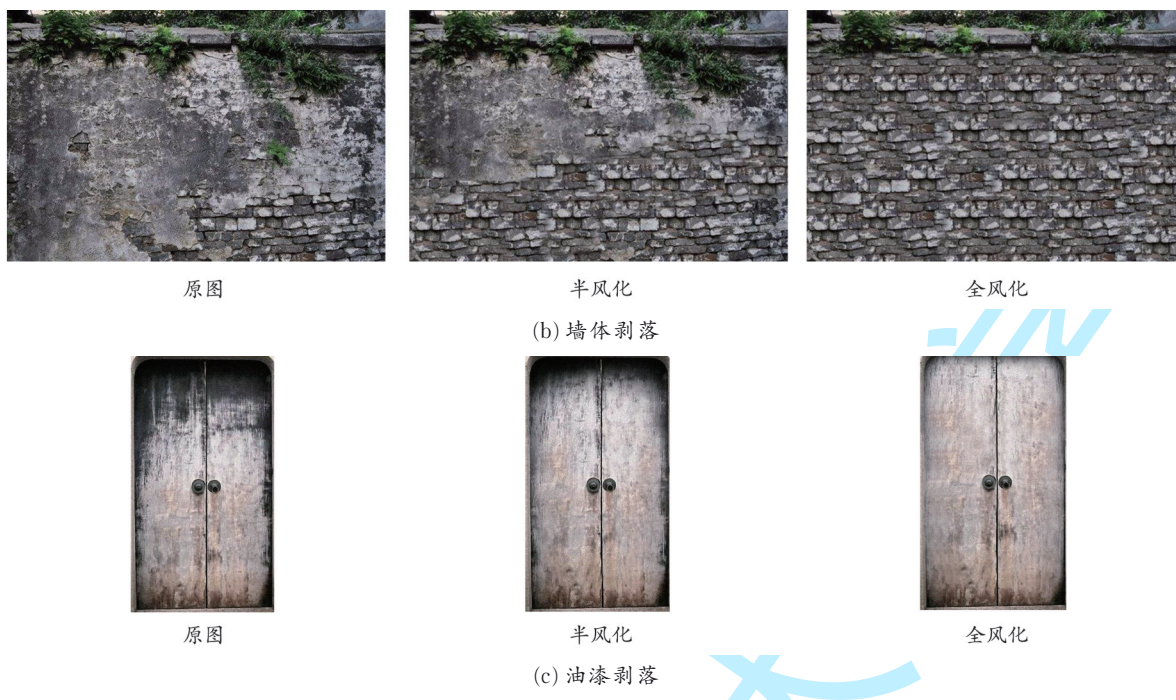


图 4 模拟风化效果

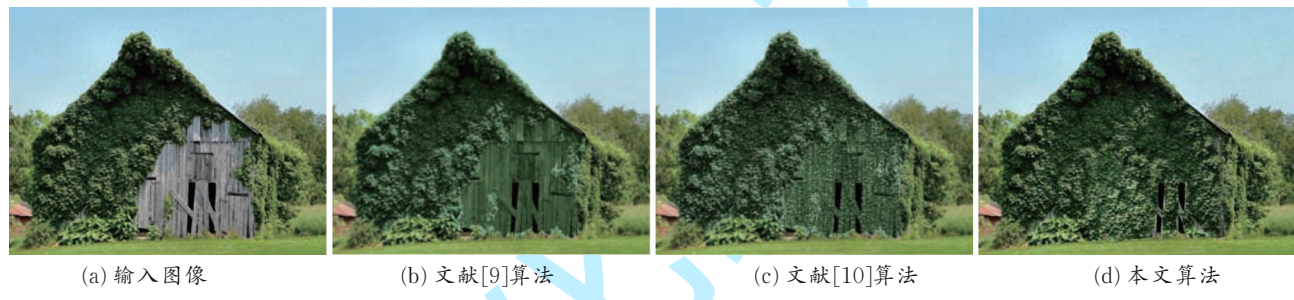


图 5 不同算法的模拟风化效果

模拟风化图像的生成时间如表 1 所示。

表 1 模拟风化图像的生成时间		s
算法	运行时间	
文献[9]	8.3	
文献[10]	7.2	
本文算法	3.6	

### 3 结 论

研究了一种基于图像的风化效应模拟方法,可

以生成纹理复杂且精细的风化效果。利用 RBF 插值的算法可以很好地处理颜色差异大的风化图像。利用图像分割算法和全局优化引导的风化度图像方法来传播详细的风化特征,可以生成效果逼真的风化图像,并且保证风化后图像中的纹理仍然连贯、一致,与原图的差异很小。在接下来的研究中,还可以利用圆形的补丁来进行风化传播。此外,也希望可以实现一种系统,能够自动地提取风化样本,这些样本中包含着不同等级的风化程度,以此构造出不同程度的风化效果。

### 参 考 文 献

[1] Wang J, Tong X, Lin S, et al. Appearance manifolds for modeling time - variant appearance of materials [J]. ACM Transactions on Graphics, 2006, 25(3): 754-761.

[2] 刘世光,陈国军,杨鹏,等.布料上污渍效果的真实感模拟[J].计算机辅助设计与图形学学报,2008,20(9):1110-1116.

[3] 邵绪强,杨晓丹,李继荣,等.大规模流体场景的真实感与实时模拟[J].中国图象图形学报,2016,21(7): 913-922.

[4] 郭迪,周明全,王学松,等.曲率分析方法模拟被雨水冲刷三维模型外观[J].计算机辅助设计与图形学学报,2015,27(2):344-350.

[5] Desbenoit B, Galin E, Akkouche S. Simulating and modeling lichen growth[J]. Computer Graphics Forum, 2004, 23(3):



- 341-350.
- [6] Xue S, Dorsey J, Rushmeier H. Stone weathering in a photograph[J]. Computer Graphics Forum 2011, 30(4):1189-1196.
- [7] Endo Y, Kanamori Y, Mitani J, et al. An interactive design system for water flow stain on outdoor images [C]// International Symposium on Smart Graphics. Berlin:Springer, 2010.
- [8] Bosch C, Laffont P Y, Rushmeier H, et al. Image-guided weathering: A new approach applied to flow phenomena[J]. ACM Transactions on Graphics, 2011, 30(3):1-13.
- [9] Bandeira D, Walter M. Synthesis and transfer of time - variant material appearance on images [C]//2009 Brazilian Symposium on Computer Graphics and Image Processing, Rio de Janeiro: IEEE, 2009:32-39.
- [10] Endo Y, Kanamori Y, Mitani J, et al. Image editing for weathering effects with geometric details [C]//In Proceedings of Computer Graphics International. Ottawa:[s.n.], 2011.
- [11] Hwang G, Yoon S H, Park S. Video-based weathering gallery[J]. Multimedia Tools and Applications, 2016, 75(6): 3391-3407.
- [12] Efros A A, Freeman W. Image quilting for texture synthesis and transfer [C]// Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. New York:ACM, 2001:341-346.
- [13] Kwatra V, Schoedl A, Essa I, et al. Graphcut textures: Image and video synthesis using graph cuts[J]. ACM Transactions on Graphics, 2003, 22(3):277-286.
- [14] Simakov D, Caspi Y, Shechtman E, et al. Summarizing visual data using bidirectional similarity [C]// 2008 IEEE Conference on Computer Vision and Pattern Recognition. Anchorage:IEEE, 2008:1-8.
- [15] Barnes C, Shechtman E, Finkelstein A. Patchmatch: A randomized correspondence algorithm for structural image editing [J]. ACM Transactions on Graphics, 2009, 28(3):1-11.
- [16] Barnes C, Shechtman E, Goldman D B, et al. The generalized patchmatch correspondence algorithm [C]// European Conference on Computer Vision. Berlin Heidelberg:Springer-Verlag, 2010.
- [17] 白宏阳, 马军勇, 熊凯, 等. 图像修复中的加权矩阵补全模型设计[J]. 系统工程与电子技术, 2016, 38(7):1703-1708.
- [18] Bellini R, Kleiman Y, Cohenor D. Time-varying weathering in texture space[J]. ACM Transactions on Graphics, 2016, 35(4):1-11.
- [19] Zhou Y, Shi H, Lischinski D, et al. Analysis and controlled synthesis of inhomogeneous textures[J]. Computer Graphics Forum, 2017, 36(2):199-212.
- [20] Tang M, Ayed I B, Marin D, et al. Secrets of GrabCut and Kernel K-means [C]// International Conference on Computer Vision (ICCV 2015). Santiago: IEEE, 2015, 1555-1563.
- [21] Garrido P, Zollhöfer M, Wu C, et al. Corrective 3D reconstruction of lips from monocular video[J]. ACM Transactions on Graphics, 2016, 35(6):1-11.
- [22] Montazer G A, Giveki D. An improved radial basis function neural network for object image retrieval[J]. Neurocomputing, 2015, 168:221-233.
- [23] Taherkhani F, Jamzad M. Restoring highly corrupted images by impulse noise using radial basis functions interpolation[J]. IET Image Processing, 2018, 12(1):20-30.
- [24] Rother C, Kolmogorov V, Blake A. GrabCut: Interactive foreground extraction using iterated graph cuts [J]. ACM Transactions on Graphics, 2004, 23(3), 309-314.
- [25] Crow F C. Summed-area tables for texture mapping [C]// Conference on Computer Graphics & Interactive Techniques. New York:ACM, 1984:207-212.

(编辑 陈 雷)