

19. 纹理合成再谈 - 一种非参数的方法



Wang H...
计算摄影从业者

106 人赞同了该文章

本文同步发表在我的微信公众号“计算摄影学”，欢迎扫码关注

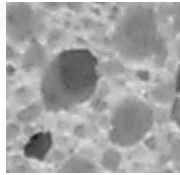


【转载请注明来源与作者】

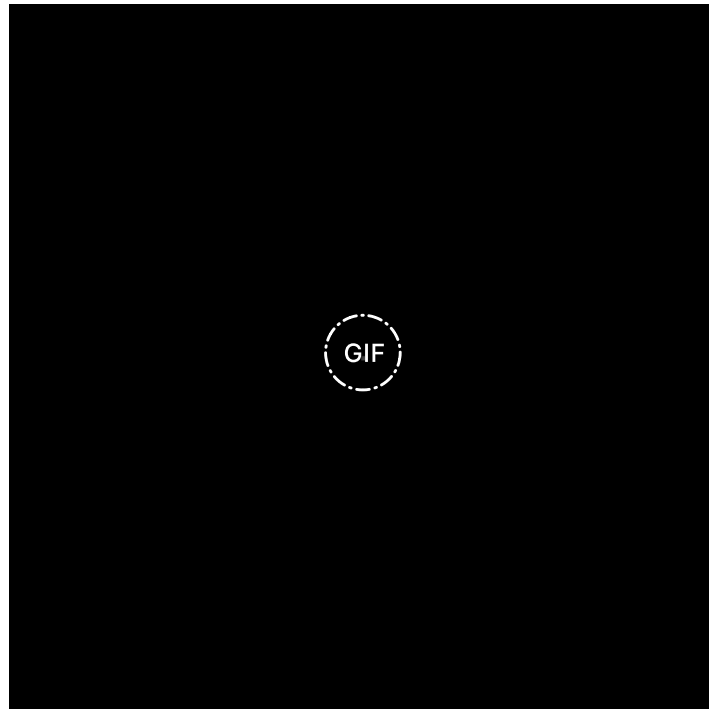
上次给大家讲了基于金字塔的纹理合成，引起了一些读者的兴趣，也引来了一些问题，看来还是有些朋友对纹理合成这个领域很感兴趣。

很多人可能都看过，现在深度学习的发展非常迅猛，其中利用GAN进行真实图像合成的结果让人非常震撼，甚至可以生成现实中完全不存在的人物的图像。而今天我所介绍的这篇发表于1999年的纹理合成的文章，在合成纹理图像这个方面，也有异曲同工之妙。它几乎可以无中生有的合成一张和参考图像高度相似的纹理图像。

让我们先来看一张参考图像，然后看看该算法从一个3x3的种子逐步合成一张风格和参考图像类似



合成过程动图：

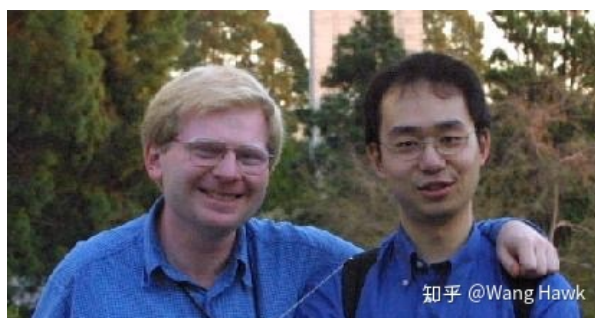


我相信以上视频也给你带来了惊讶，当你阅读完这篇文章理解了算法的原理之后，你会因为该算法的简单而更加的惊讶，甚至像我一样有自己动手复现它的冲动

今天要介绍的这个方法源自论文：

Efros and Leung, "Texture Synthesis by Non-parametric Sampling," ICCV1999

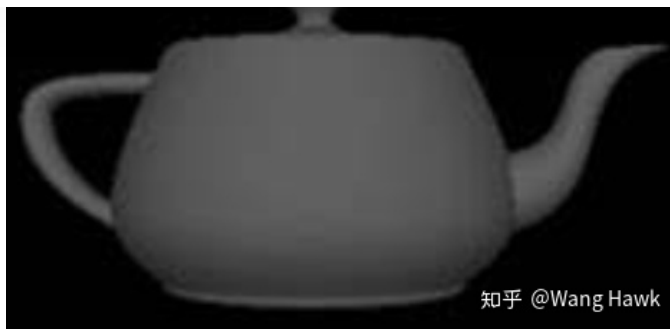
作者照片，看起来挺睿智~



今天我就带你详细看看这个特别的算法要解决的问题，它的原理，以及他有哪些应用方向。

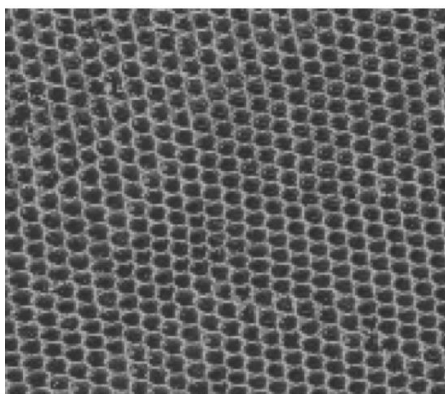
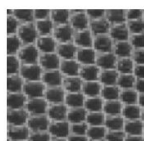
一、问题说明

我们在18. 纹理合成再探 - 一种基于金字塔的方法中讲过，纹理合成能够让单调呆板的物体显得很有生气，正如我所举的茶壶的例子：

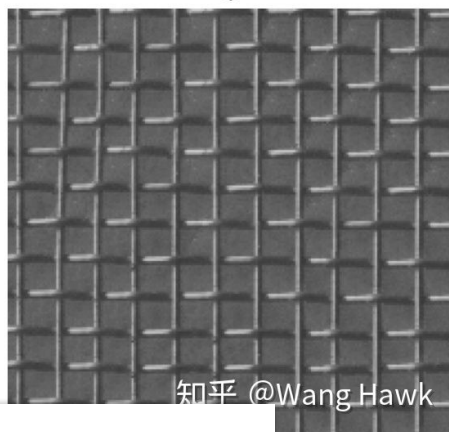
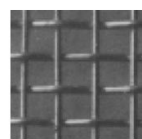


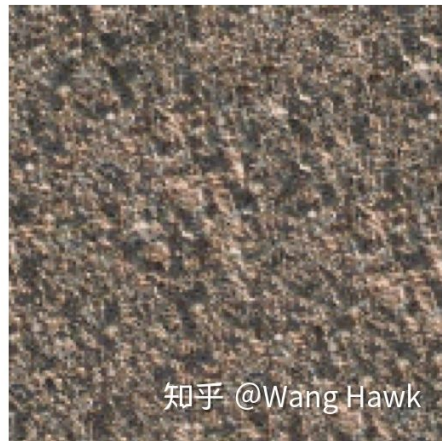
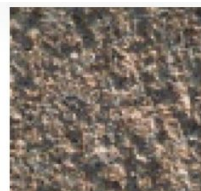
但纹理合成的作用远不止于此，另外一个很有用的用途则是从一个小的纹理片段合成一幅同等风格的大尺寸的纹理图，就像这样：

reptile skin



aluminum wire





知乎 @Wang Hawk

white bread

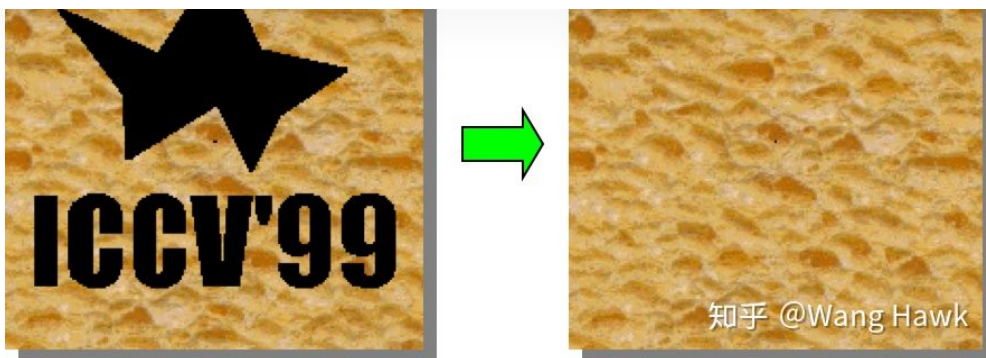


brick wall

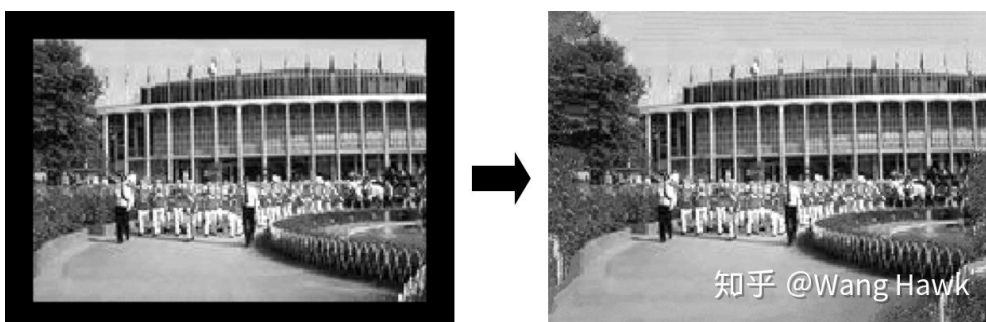
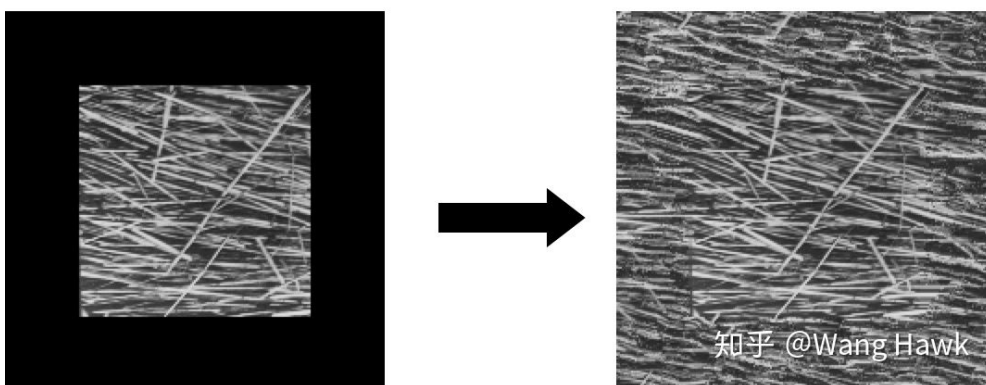
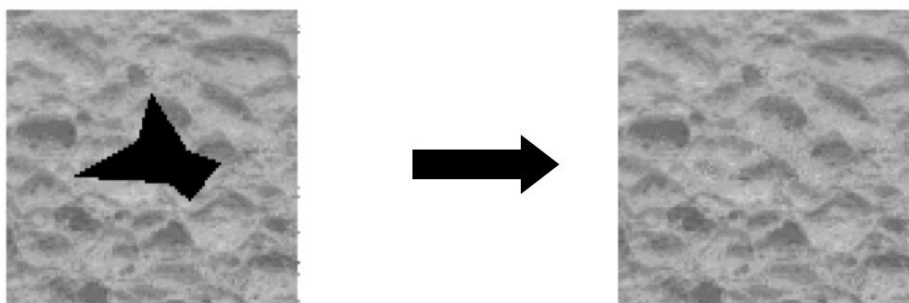


知乎 @Wang Hawk

或是填充一幅图像中的空洞：

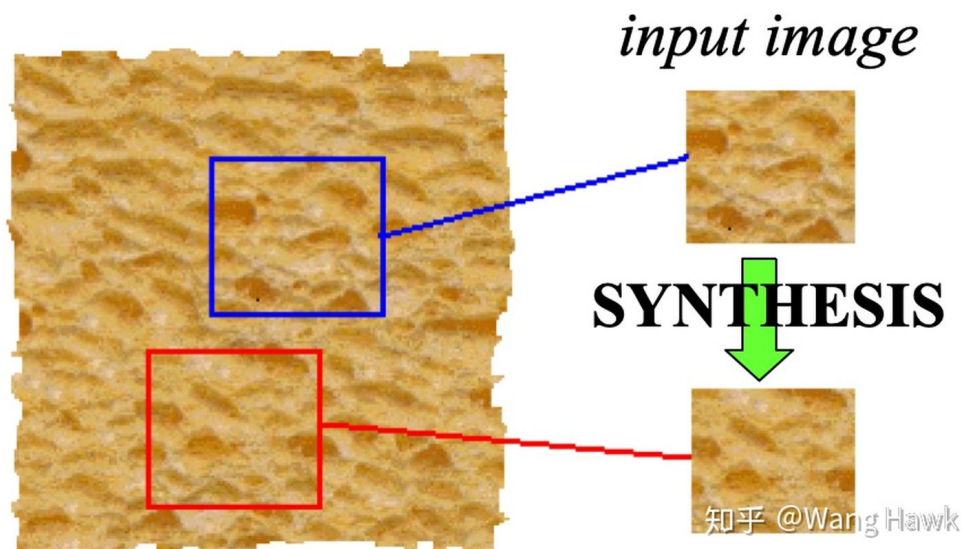


或是扩张图像，填补周围原本不存在的细节：



甚至还能填充原本不存在的文字！





这个问题是如此有用，使得很多人就此问题做了大量的研究。但这个问题也确实有很大的挑战性：纹理图像是多种多样的，正如我们之前看到的，纹理的类型有随机纹理、规则纹理，大量实际的纹理图像则是介于两者之间的。那么如何捕捉到纹理的本质呢？之前的一些研究希望能够用某些模型来捕捉纹理的本质，例如18. 纹理合成再探 - 一种基于金字塔的方法所讲，它就是用直方图来捕捉纹理的特征，这本质上是一种带参数的做法，它对随机性纹理能起到一定的效果，但对其他类型的纹理效果就很不好。所以，需要一种更好的方法来解决这个问题，能够使得输出图像能够有输入图像的局部结构，并能够适配多种类型的纹理。

让我们来看看作者的思路。

二、解决思路

1948年，信息论之父香农发表著名的论文：



Claude Elwood Shannon (1916–2001)

知乎 @Wang Hawk

其中他提到了一种有趣的使用n-gram生成看起来像英语的书面文本的方法。其思想是将语言建模为一个广义马尔可夫链：一组n个连续的字母(或单词)组成一个n-gram，并完全确定下一个字母(或单词)的概率分布。使用大量的语言样本(例如，一本书)，可以为每个n-gram构建概率表。然后，我们可以从这个马尔可夫链中反复取样，生成听起来像英语的文本。这是早期计算机程序“MARKVSHANEY”的基础，这个程序可以让计算机从一个种子字母，自动的生成一些有趣的英语句子，例如：

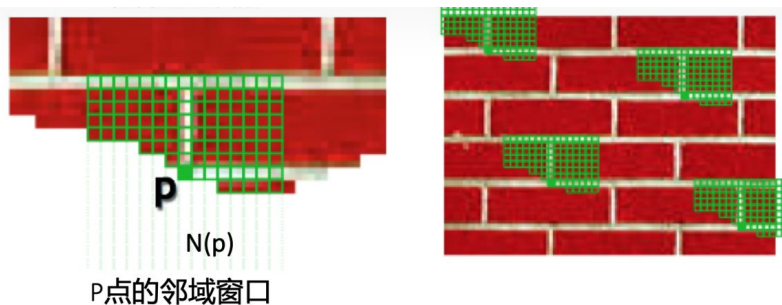
I spent an interesting evening recently with a grain of salt

As I've commented before, really relating to someone involves standing next to impossible.

One morning I shot an elephant in my arms and kissed him.

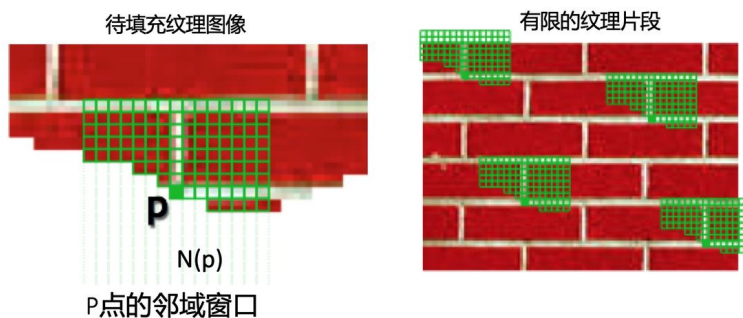
这个技术也是现在很多自然语言处理技术的基础，包括我们的各种中文输入法也使用了这种思想：当你敲下一个字母时，输入法会自动的联想出多种可能的词汇供你选择，并按照概率从高到低排列。

本文的作者也受到了这篇文章的启发，只不过他们将原本用于文字的想法扩展到了图像领域，其基本思想是，



- 所有相似的邻域构成了一个无参数的概率分布函数 $P(p|N(p))$
- 当随机挑选时，实际上是从此函数中进行采样

用模板匹配的方式，搜索邻域与待填充像素的邻域最接近的N个匹配点，并从中随机挑选一个来填充目标

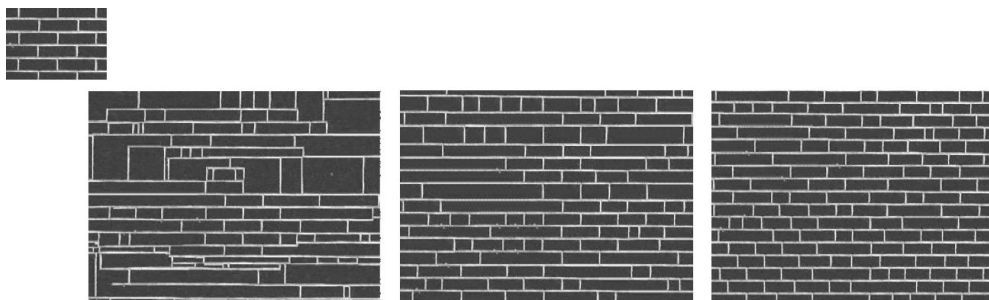


由于已知的纹理片段是有限的，所以与待填充点邻域完全一致的源可能并不存在，所以作者采用SSD来衡量待填充点邻域与源之间的相似度，用于挑选最相似的一些源

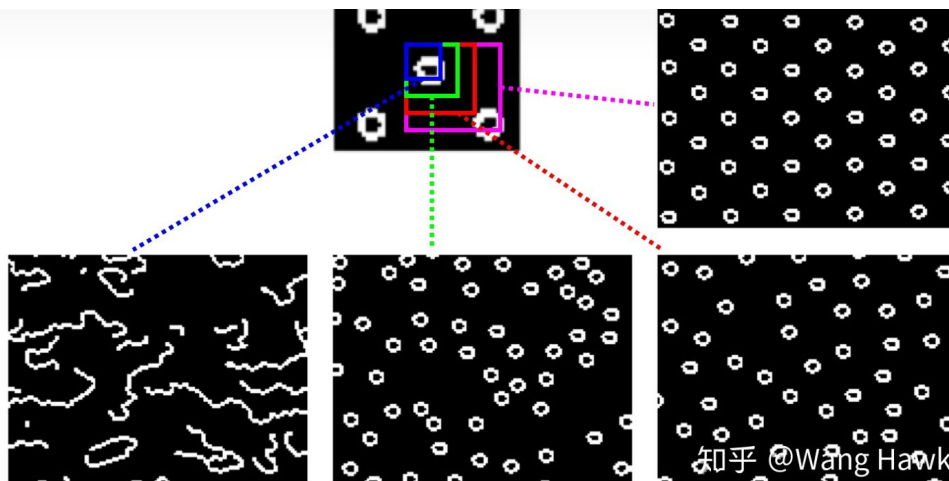
这个方法看起来确实是惊人的简单，不过里面还是有几个细节需要注意：

1. 邻域的大小：

待匹配的点的邻域大小会影响效果的好坏，这里有一个例子：



Increasing window size →



2. 填充点的顺序

这里作者特别指出，我们不能按照通常的行扫描线的顺序（从左到右，从上到下）去填充，而是应该按照洋葱皮的方式，由外层到内层去填充，因为这样才能得到好的效果：

Toy inpainting example:



image with hole



raster-scan order



知乎 @Wang Hawk

我这里给大家看一个动图来说明洋葱皮填充顺序：



三、算法实现

作者非常贴心的在论文网页上附上了算法实现说明和伪代码：[graphics.cs.cmu.edu/peo...](http://graphics.cs.cmu.edu/people/peo...)

你可以自行仔细阅读。

我也自己实现了此算法（偷懒只实现了单通道）。该算法的一大缺点是速度特别慢，因为它是一个像素一个像素的填充，所以真的是龟速。一开始我打算用Python像往常一样给大家展示 Notebook，奈何它实在太慢耗尽了我有限的耐心，所以我干脆用C++实现了一把，为了让它速度能快上一些，我还用上了OpenMP尽量加速，但还是太慢~

所以这次就大家自行根据上面的伪代码和我的C++实现来揣摩吧，此次的代码位于：

github.com/yourwanghao/...

四、总结

纹理合成是一个非常有用的领域，今天我们介绍了一个发表于1999年但却真的能做到天衣无缝的合成纹理的方法，并且还给出了C++实现。该方法基于1948年香农提出的信息论的思想，认为一个像素的值只与其邻域相关，并将问题转换为了一个在邻域概率分布函数中采样的问题。它的算法惊人的简单（也惊人的慢），效果也惊人的好。

希望这篇文章成功的吸引了你对纹理合成，特别是纹理合成用于空洞填充的兴趣，因为我接下来还会有两篇文章都会延伸讲解这个领域。

别忘了给我点赞哦^_^

本文的大部分信息来自于作者对论文的官方介绍：

1. 论文: [Texture Synthesis by Non-parametric Sampling](#), Alexei A. Efros and Thomas K. Leung, ICCV'99
2. 代码说明: graphics.cs.cmu.edu/peo...

少量信息来自于：

[CMU 2017 Fall Computational Photography Course 15-463, Lecture 9](#)

编辑于 2019-11-22 19:02

Computational photography

计算机视觉

计算机图形学

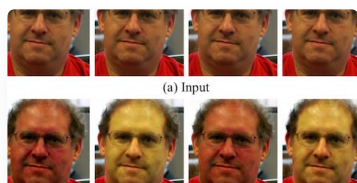
文章被以下专栏收录



计算摄影学

深入的、系统性的介绍计算摄影学

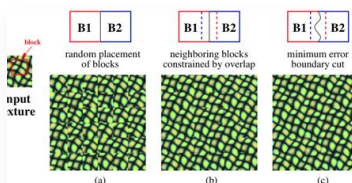
推荐阅读



9. 图像处理的应用 - 欧式视频放大的实现

Wang ...

发表于计算摄影学



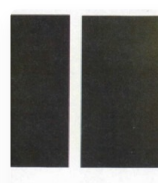
20. 用于纹理合成和转移的图像缝合

Wang Hawk



18. 纹理合成再探 - 一种基于金字塔的方法

Wang Hawk



11. 图像合成

Wang Hawk

9 条评论

⇌ 切换为时间排序

写下你的评论...



高木森

2019-06-09

pre DL 的工作还是很有趣的

👍 1



Bruce Yan

2020-05-30

能不能这样理解，其实思想有点像当今流行的分布式嵌入表示：一个像素应该是什么样的，取决于它与邻居们（相邻像素）的关系

👍 赞



大眼睛

2019-10-04

▲ 赞同 106 ▼

● 9 条评论

➦ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...

知乎用户

2019-06-19

为什么这里又说上一篇是带参数的方法？上一期提到“它是一种非常简单的非参数的方法”

👍 赞



Wang Hawk (作者) 回复 知乎用户

2019-06-19

好问题，我审视了一下我上一篇文章，发现确实不正确。上一篇其实确实是把图像的信息抽取为了直方图，严格来说算是有参数的，我随后会修改之，感谢你的指正

👍 1



知乎用户 回复 Wang Hawk (作者)

2019-06-20



👍 赞



知乎用户

2019-06-10

这个算法和最近很火的Wave Function Collapse异曲同工的感觉

👍 赞



知乎用户

2019-06-09

解决游戏里地形纹理Tiling的神器，手动点赞！

👍 赞



oreman 回复 知乎用户

2020-02-02

地形的话直接上纹理合成几乎不可能实时吧😂

👍 赞