

# 基于纹理渲染的分形图案设计

魏宝刚 庞向斌 朱文浩 潘云鹤

(浙江大学计算机科学与技术学院, CAD&CG 国家重点实验室, 杭州 310027)

**摘要** 传统分形图案的纹样在表现实际景物特征方面存在不足。针对这一问题, 提出了一种基于纹理渲染的分形图案设计概念和技术, 并改进了 Wei 的点匹配纹理合成算法, 即首先用 2 维自相关函数先分析纹理的构成和分布, 然后确定合成的 L 形邻域大小, 最后将分形图案作为结构约束, 并在 HSL 颜色空间上控制色彩渲染, 实现了结构约束的纹理合成, 并通过选择两个特殊的分形数集来进行实验, 生成了既有样图纹理特征, 又能控制色彩变化的分形图案。

**关键词** 分形几何 纹理合成 2 维自相关函数 图案设计 渲染绘图

中图法分类号: TP3 文献标识码: A 文章编号: 1006-8961(2006)05-0689-06

## Using Texture Render in Fractal Pattern Design

WEI Bao-gang, PANG Xiang-bin, ZHU Wen-hao, PAN Yun-he

(Computer Science and Technology College of Zhejiang University, State Key Lab of CAD&CG, Hangzhou 310027)

**Abstract** Traditional fractal pattern design has some disadvantages such as being lack of the ability for describing basic cell of pattern and unable to effectively reflect the characters of real scenery. In order to solve the problems, we present a novel pattern design technology of fractal geometry based on image texture rendering. First we improve the texture synthesis algorithm of Wei by using two-dimensional autocorrelation function to analyze the structure and distribution of the texture first, and then determine the size of L neighborhood. Second several special fractal sets are adopted and HSL color space is chosen, so that the fractal structure can be used to manipulate the texture synthesis. The experiments have shown that using the novel technology patterns with different style, color, texture characteristic can be generated quickly.

**Keywords** fractal geometry, texture synthesis, two-dimensional autocorrelation function, pattern design, image render

## 1 引言

近 20 多年来, 分形几何的研究受到非常广泛的重视, 其原因在于分形既有深刻的理论意义, 又有很大实用价值。目前, 分形技术已经在数学物理、地质生态、材料化学、纺织和艺术等众多领域得到了应用。而基于分形几何的图案设计技术则是分形的一个重要应用领域<sup>[1]</sup>。

图案设计主要是指对图案的纹样和色彩进行设计, 其中, 纹样题材广泛, 各种物体和景象都可成为纹样素材, 而色彩则主要依靠人为方式进行填充和

处理。

纹理是纹样的一种, 如今虽然对纹理的概念还没有统一的定义, 但纹理合成技术已经成为当前计算机图形学、计算机视觉和图像处理领域的研究热点之一, 特别是基于样图的纹理合成技术已得到了广泛的应用。通过纹理合成的方式, 可以生成任意大小, 并与样图纹理极具相似性的纹理块。这就使得自然的或人工设计的图案纹理和色彩引入到新图案中去成为可能。

本文提出一种基于纹理渲染的分形图案设计技术, 其目的是将上述设想变成现实和通过纹理合成和分形几何两者的结合, 为计算机图案设计提供新

基金项目: 浙江省自然科学基金项目 (M603228); 国家自然科学基金项目 (60373037)

收稿日期: 2004-11-12; 改回日期: 2006-02-13

第一作者简介: 魏宝刚 (1960 ~ ), 男, 副教授。1997 年获西北大学博士学位, 主要研究领域为人工智能、图像处理、模式识别。E-mail:

wbg@zju.edu.cn

的思路。

## 2 纹理合成相关技术

### 2.1 纹理和纹理合成技术

大家知道,纹理是能让观察者产生相同感觉的同源视觉图案,它可用纹理基元的数量和类型以及这些基元的空间组织或排列来描述。这种纹理的空间组织可能是随机的,也可能由具有成对关系的相邻基元组成,或者几个基元同时相互关联组成,而且这样的关联可以是结构的、概率的或是函数的<sup>[2]</sup>。

目前基于样图的纹理合成主要有点匹配算法<sup>[3,4]</sup>和块拼接算法<sup>[5-7]</sup>两种不同的样图合成采样算法。本文的合成算法主要依据 Wei 在 2000 年 SIGGRAPH 会议上提出的点匹配算法<sup>[3]</sup>,并在此基础上加入了纹理基元大小和空间组织的识别技术。

### 2.2 纹理统计分析方法

为了提高纹理合成的质量和速度,纹理合成之前,应先对纹理进行分析。本文的纹理分析采用统计方法,首先把纹理理想化地看成是由形状相似的基元在空间均匀分布而产生的,然后采用自相关函数来分析纹理基元的大小和分布情况。这里 2 维自相关函数被定义为

$$R(\Delta x, \Delta y) = \frac{\sum_{i,j} I(i, j)I(i + \Delta x, j + \Delta y)}{\sum_{i,j} [I(i, j)]^2} \quad (1)$$

其中,  $i$  和  $j$  被限于特定的图像  $I$  之内,这相当于假设在此区域以外的图像为零。增量位移  $d = (\Delta x, \Delta y)$  可以是负的。对于给定的纹理图像来说,自相关函数在  $d = 0$  时的最大值为 1,并随正向和负向位移变化呈指数函数下降,其中央峰点的斜率可表示纹理的粗细度,如果纹理基元较大,则自相关函数的下降相对缓慢;与此相反,若基元较小,则下降迅速。当纹理是周期性的或规则的图案时,则  $R(\Delta x, \Delta y)$  就会周期性地出现最大值。这样根据自相关函数的峰点和谷点分布就可以分析纹理基元的大小和分布情况。

在具体实施分析时,为了减少  $\Delta x, \Delta y$  的搜索范围,以加快计算速度,可以先经验性地给定一个大概的基元大小范围值,然后让  $\Delta x, \Delta y$  在给定的范围的变化;如果  $i + \Delta x$  或  $j + \Delta y$  的值超出了纹理的范围,则采用拓扑学的方法通过减去纹理的宽度或高度来使得其值保证在合理的范围内。

以图 1(a) 作为实测纹理,其  $\Delta x, \Delta y$  的搜索范围为 0 ~ 20,实验得出的 2 维自相关值如表 1 所示(仅列出部分)。

表 1 2 维自相关值

Tab. 1 Two-dimensional autocorrelation value

$(\Delta x, \Delta y)$	自相关值	$(\Delta x, \Delta y)$	自相关值
(0,0)	1.000 00	(10,11)	0.785 97
(0,3)	0.806 82	(10,13)	0.764 55
(1,3)	0.784 18	(11,13)	0.746 78
(1,5)	0.690 43	(11,15)	0.739 21
(2,5)	0.671 93	(12,15)	0.729 29
(2,7)	0.642 54	(12,17)	0.742 22
(3,7)	0.637 19	(13,17)	0.742 95
(3,9)	0.653 51	(13,19)	0.726 72

自相关值的最小值出现在  $(\Delta x, \Delta y) = (3, 7)$  处,其自相关值为 0.637 19。在实测的 339 个自相关值数据中,统计后发现,自相关值大于 0.9 的有 3 例,0.8 ~ 0.9 之间的有 50 例,0.7 ~ 0.8 之间的有 228 例,0.6 ~ 0.7 之间的有 58 例,而小于 0.6 的值不存在。按下式对  $(\Delta x, \Delta y)$  做几何平均:

$$\bar{T} = \sqrt{\frac{(\Delta x)^2 + (\Delta y)^2}{2}} \quad (2)$$

则得到的  $\bar{T} = 5.38$ ,取整之后纹理基元大小为 5。

实验过程中发现,由于一些结构性不强的自然纹理,其自相关值一般都大于 0.9,因此在统计计算时,如果发现大于 0.9 的自相关值占了很大比例(如 80% 以上),那么就可以确定是自然纹理,这时就可以以一个较小的数值来反映纹理基元的大小,通常情况下设定该数值为 3,会得到比较好的合成效果,且速度上比取大的基元值也有提高。

### 2.3 改进的 Wei 纹理合成算法

一旦求得纹理基元的大小,就可以按纹理基元的大小来进行纹理邻域的比较。马尔可夫随机场模型认为,纹理是由局部相似的随机过程生成的,并可以把纹理图像中的任何一个像素都看成是由其周围空间存在的邻域点的集合特征化描述的结果。若将分析得到的纹理基元大小作为点匹配过程中 L 形邻域的大小,则可通过计算不同邻域之间的距离来进行邻域匹配。两个形状相同的邻域  $N_1, N_2$  的距离定义为

$$d(N_1, N_2) = \sum_{\substack{p \in N_1, q \in N_2 \\ p, q \text{ 相对应}}} \text{sqrt}((f(p) - f(q))^2) \quad (3)$$

$f(p), f(q)$ 表示图像  $p, q$  点的特征值。

图1描述了纹理合成过程,图1(a)是纹理样图,图1(b)~图1(d)是图像放大处理后的合成过程。图中的L形区域为纹理的邻域,邻域大小已经通过纹理分析得到。图1(b)为开始合成的第1个点,而组成邻域的点则是通过拓扑关系得到的白噪声像素点,此时整幅图像也都是噪声;图1(c)是合成到中间部分的情况,此时组成邻域的点已经是前阶段合成得到的纹理像素点;图1(d)是最后的合成结果。

对于这种逐点进行扫描线合成的算法,由于输出纹理中,除了开始几行的匹配要使用到随机白噪声初始化的图像信息外,其他像素点的合成都是根据已合成点来进行,从而保证了合成纹理的准确性;同时又因为输入纹理只需要一小块,所以只需要较少的时间就可以合成出整个纹理。

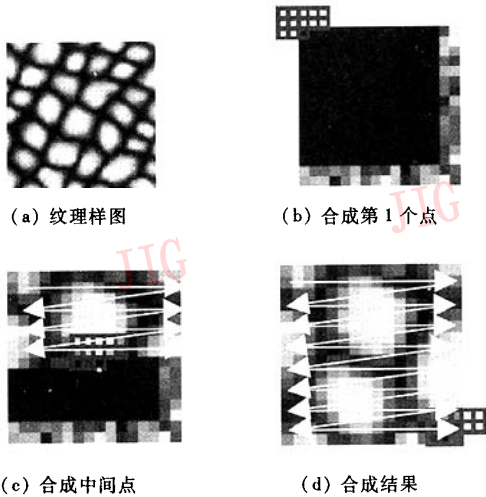


图1 合成过程示意图

Fig.1 Synthesis process sketch map

### 3 分形图案设计

分形图案是无标度意义下,具有无穷细节的自相似图形,它体现了自然界的无序和变幻无穷的特征。基于分形算法的图案设计和普通图案设计一样,包括纹样造型、色彩构成和构图等组成部分。纹样是组成图案的基础,它的基本单元称为单位纹,在分形艺术图案中,单位纹是一种具有分形特性的自由纹样,一般它们由固定的分形算法生成,而色彩则是由不同色相、明度和纯度,按照混合、对比和调和手法配合得到的效果。由于色彩的构成与颜色变量的取值和

运算有关,因此作为一组具有分形特性的数值集合的映象,其色彩同样具有自身的分形特性,这里构图主要是指分形图案画面的布局和组织形式。

#### 3.1 分形数集

在分形图案设计上使用得最多的是 Julia 数集和 Mandelbrot 数集,两个集合均来自非线性映射  $x \rightarrow x^2 + \mu$ ,可通过变换和反复迭代,并按照某一规则,对收敛的点设置不同的颜色,即可获得集合的分形计算机图像。

本文在分形数集的选择和实现上,除了基本的 Julia 和 Mandelbrot 数集之外,为了使得到的分形图案更具实用价值,本文采用了 Connett 和 Martin 分形数集。

##### 3.1.1 Connett 数集

Connett 数集是以圆为基础,却能产生多种以正方形为基础的图案。Connett 数集分形图案主要由以下4个参数进行控制,即被扫描的矩形区域的左下角坐标  $(x, y)$  和被扫描矩形的长  $L$  和宽  $W$ 。若矩形区域越小,则其平面就显得离观察者越近,图案的放大倍数就越大,并将出现类似于光学中的干涉或衍射现象的图案,但整体图案此时会显得比较单一;反之,矩形区域越大,就越能将分形图案的整体特性表现出来。由于不同的  $x$  和  $y$  值会导致产生不同的偏移量,从而产生不同的分形图案。

图2是3个由 Connett 数集产生的分形图案。表2列出了这3个图形的具体参数值。

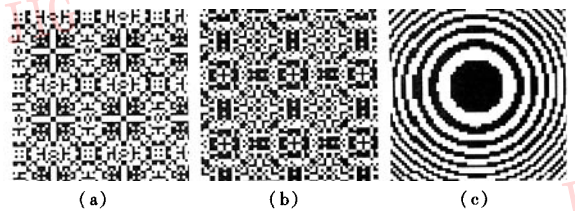


图2 Connett 数集的3个分形图案

Fig.2 3 Fractal images of connett set

表2 Connett 数集图例参数  
Tab.2 Parameters of connett set

图例	$x$	$y$	$L$	$W$
1	50	35	840	840
2	555	555	80	80
3	555	555	10	10

3 个分形图案中矩形区域依次变小,从图 2(a),图 2(b)分形图案可以看出,不同的  $x$  和  $y$  值生成的分形图案不同,比较图 2(b),图 2(c)可以看出,在相同的  $x$  和  $y$  值情况下,缩小  $L$  和  $W$  的值有明显的放大效果。

### 3.1.2 Martin 数集

Martin 数集是受到 Mandelbrot 数集启发而形成的,该数集虽然运用了 Mandelbrot 数集迭代计算的想法,但与 Mandelbrot 数集进行的复数迭代不同,它是以实数作为迭代的基础。此外, Martin 数集只选用一个初始值进行迭代,而 Mandelbrot 数集初始迭代值则遍布在一个复平面上的无数个点上。

以下是两对用于 Martin 数集迭代的公式:

$$\begin{aligned} X &= Y - \text{sgn}(X) \times \text{abs}(B \times X - C) / 2 \\ Y &= A - X \end{aligned} \tag{4-1}$$

$$\begin{aligned} X &= Y - \text{sgn}(X) \\ Y &= A - X \end{aligned} \tag{4-2}$$

式(4-1)中符号函数  $\text{sgn}(X)$  可根据参数  $X$  的正负值分别取 1 或 -1,函数  $\text{abs}(B \times X - C)$  表示取括号内数值的绝对值。若改变  $A, B, C$  3 个参数,则可以产生不同的分形图案,其中  $A$  用于控制图案的缩放大小,该数值越大,图像缩放的比例越小;  $B$  用于控制图案的外观;  $C$  用于控制图案的范围大小(扩散程度),该数值越大,图案可显示出更大范围内的形态。式(4-2)中只需一个参数  $A$  和简单的正弦计算就可以生成分形图案,通过实验可以验证,当  $A$  值与  $\pi$  相差在 0.07 以内时,用式(4-2)也能够产生出比较理想的分形图案。

式(4)的两个公式中,  $X, Y$  的初始值均可以设为 0,但需要指定的是迭代的次数,因为迭代次数表示分形图案由多少个像素点组成,次数越多,图案越精细。图 3 为 4 个 Martin 数集分形图案的例子。

图 3(a),图 3(b)是按式(4-1)生成的分形图案,图 3(a)是原图,图 3(b)是局部放大后的图案,其参数分别为  $A = 24, B = -185, C = -269$ ,迭代次数为 100 000;而图 3(c),图 3(d)则是按式(4-2)生成的分形图案,图 3(c)是原图,图 3(d)是局部放大后的图案,其参数  $A = 3.2$ ,迭代次数为 200 000。

### 3.2 合成分形与纹理渲染规则

在传统的分形图案生成的过程中,需要通过指定收敛点颜色的设置规则来进行渲染。虽然其已具备了分形的几何结构,但由于图案的色彩、纹样、构图特性并不能很好地满足设计要求,因此,本文引入

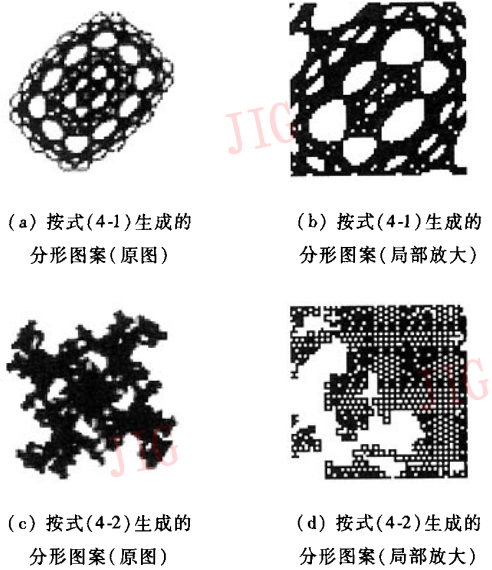


图 3 Martin 数集分形图案

Fig. 3 Fractal images of martin set

纹理合成的方法作为分形图案色彩、纹样的渲染规则。

合成分形的思想是通过采用纹理合成的渲染绘图规则来改善传统分形图案生成过程中色彩只依靠填色函数确定,纹样构图简单重复的缺陷,以便将现实中丰富多彩的纹理图案应用到分形图案上。在合成分形的过程中,是先通过分形技术控制图案的几何结构来生成具备分形结构的几何图案,再采用纹理合成的方法将样图纹理渲染到分形结构中去,使得图案同时满足分形的几何结构又具有纹理图案的特征。

渲染的实现过程是根据迭代过程中产生的点集的信息,对于点集中的每个点,则是先设置一个 Mask 位,然后采用改进后 Wei 的纹理合成算法来进行渲染绘图。假设样图纹理为  $T_{in}$ ,合成纹理为  $T_{out}$ ,则具体有以下两种不同的纹理渲染规则:

**规则 1:** 根据 Mask 位进行纹理渲染,即只对 Mask 位置的点进行渲染,而其他点则略过。这一规则相当于对分形图案的几何外形进行渲染,其虽速度较快,但由于合成过程中,在进行邻域相似性匹配时,用到的邻域信息是不完整的(包含未合成的点),因此渲染效果受到一定影响。规则 1 的形式化描述如下:

```
For each Pixel(X, Y) in Texture
  If Mask(X, Y) = TRUE Then
```

```
Synthesis(X, Y);
ShowPixel(X, Y);
```

规则2:根据分形图案的外接的矩形来合成整个矩形内部的纹理,但对置Mask位的点(即点集中的点)显示纹理,而对其他点则只需临时存放渲染后纹理信息。使用这一规则相当于对整个外接矩形进行了纹理渲染,其渲染速度会慢一些,但渲染效果更好。规则2的形式化描述如下:

```
For each Pixel(X, Y) in Texture
  Synthesis(X, Y);
  If Mask(X, Y) = TRUE Then
    ShowPixel(X, Y);
  Else
    SavePixel(X, Y);
```

为了产生更加丰富的纹理渲染效果,可考虑将各个像素点的颜色从RGB颜色模型转换到HSL颜色模型。由于HSL颜色模型着重于光线强弱和颜色色调的变化关系,因此通过适当的改变色调的取值,可以获得色调不同的分形图案。

综合以上的论述,假设输入样图纹理为 $T_i$ (下角 $i$ 代表in),合成(输出)纹理为 $T_o$ (下角 $o$ 代表out),两个纹理渲染规则分别为Rule1和Rule2,则整个合成分形过程可以描述为以下的伪代码:

```
Function SynthesisFractal( $T_i, T_o, \text{Offset}, \text{Rule}$ )
{
   $T_o \leftarrow \text{CreateFractalStructure}()$ 
   $S \leftarrow \text{CalculateParticelSize}(T_i)$ 
  Switch(Rule)
  Case Rule1:
    MaskSynthesis( $T_i, T_o, S$ )
  Case Rule2:
    NormalSynthesis( $T_i, T_o, S$ )
  ChangeHUE( $T_o, \text{Offset}$ )
}
Function MaskSynthesis( $T_i, T_o, S$ )
{
  For Each Pixel(X, Y) In  $T_o$ 
  If Mask(X, Y) = TRUE Then
     $P \leftarrow \text{CreateNeighborHood}(T_o, X, Y, S)$ 
    Pixel(X, Y)  $\leftarrow \text{FindBestMatch}(P, T_i, S)$ 
}
Function FindBestMatch(P,  $T_i, S$ )
```

```
MatchValue  $\leftarrow ax$ MatchValue
TMathValue  $\leftarrow \text{MaxMatchValue}$ 
BestMatch  $\leftarrow \text{NULL}$ 
For Each Pixel(X, Y) In  $T_i$ 
   $P' \leftarrow \text{CreateNeighborHood}(T_i, X, Y, S)$ 
  TMatchValue  $\leftarrow \text{CalculateL2Distance}(P, P')$ 
  If MatchValue > TMatchValue
    MatchValue  $\leftarrow \text{MatchValue}$ 
    BestMatch  $\leftarrow \text{Pixel}(X, Y)$ 
Return BestMatch;
}
```

函数CreateFractalStructure()用于生成具有分形结构的原始图案;CalculateParticelSize()用于计算样图纹理中纹理基元的大小;ChangeHUE()用于改变合成之后纹理的色调关系,丰富渲染效果;MaskSynthesis()和NormalSynthesis()则分别按不同的纹理渲染规则进行纹理渲染;函数FindBestMatch()主要用于邻域搜索和邻域匹配的实现;CalculateL2Distance()用于计算两个邻域之间的距离。

## 4 效果示例

示例1:用式(1)的Martin数集,参数分别为: $A = -398, B = 5, C = -452$ ,迭代次数为5000,色调(H)偏移量为0,邻域大小为3。图4(b)采用渲染规则2,用时35s;图4(c)采用渲染规则1,用时27s。样图(图4(a))为 $32 \times 32$ 个像素,分形图案大小为 $100 \times 100$ 个像素。比较图4(b)和图4(c)可看出,采用渲染规则1得到的图案整体效果比采用规则2得到的图案要差一些。

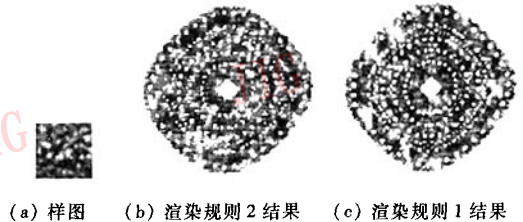


图4 示例1

Fig.4 Sample 1st

示例2:采用Connett数集,参数分别选为: $x = 555, y = 555, L = 80, W = 80$ ,邻域大小为5,采用渲染规则2。图5(b)为未使用色调偏移量的结果;

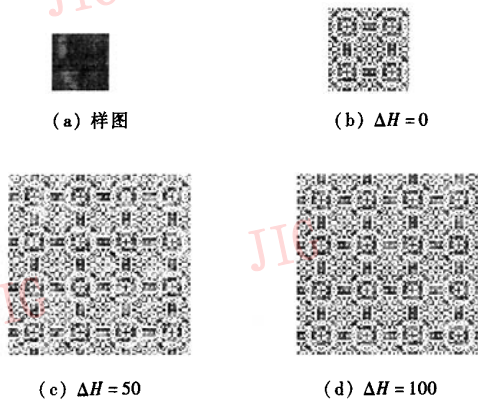


图 5 示例 2

Fig. 5 Sample 2th

图 5(c) 色调偏移量为 50; 图 5(d) 色调偏移量为 100。图 5(a) 为  $32 \times 32$  像素的样图。分形图案大小为  $100 \times 100$  像素, 用时均为 41s。

## 5 结 论

本文通过使用 2 维自相关函数改进了 Wei 的点匹配纹理合成算法, 使得算法具备了自动识别纹理基元和确定 L 形邻域大小的特点。在此基础上, 提出了基于纹理渲染的分形图案设计方法, 综合运用纹理合成和分形图案设计技术, 通过纹理渲染快速得到具备任意样图特征的分形图案。这一技术为图案设计提供了新的思路, 也为纹理合成研究开辟了新的应用领域, 可以应用于工艺美术、纺织、印染等

涉及图案设计的行业。

接下来的研究重点将放在研制更好的纹理合成渲染绘图算法和能自动生成具有特殊风格的分形图案技术, 以及这一方法的实际应用。

## 参考文献 (References)

- 1 Wang Xiao-min. Modeling algorithm of computer aided fractal artistic pattern design[J]. South China Normal University Journal(Scientific Edition), 2003, 1: 46 ~ 51. [王小铭. 计算机辅助分形艺术图案设计的算法造型技术[J]. 华南师范大学学报(自然科学版), 2003, 1: 46 ~ 51.]
- 2 Xu Guang-you. Computer Vision [M]. Teaching material of Computer Science of TsingHua University, Revised Edition, 2002. [徐光祐. 计算机视觉[M]. 北京清华大学计算机系教材, 2002 年修订版.]
- 3 Efros A, Leung T. Texture synthesis by non-parametric sampling [A]. In: International Conference on Computer Vision[C], Corful, Greece, 1999; 1033 ~ 1038.
- 4 Wei Li-yi, Levoy Marc. Fast texture synthesis using tree-structured vector quantization[A]. In: Proceedings of ACM SIGGRAPH[C], New Orlean, Louisiana, USA. 2000; 479 ~ 488.
- 5 Efros A, Freeman W T. Image quilting for texture synthesis and transfer [A]. In: Proceedings of ACM SIGGRAPH [C], Los Angelas, CA, USA, 2001; 341 ~ 346.
- 6 Cohen M F, Shade Jonathan, Hiller Stefan, Deussen Oliver. Wang tiles for image and texture generation [A]. In: Proceedings of SIGGRAPH[C], San Diego, CA, USA, 2003; 287 ~ 294.
- 7 Vivek Watra, Arno Schödl, Irfan Essa, Greg Turk, Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts[A]. In: Proceedings of SIGGRAPH[C], San Diego, CA, USA, 2003; 277 ~ 286.