

基于图像类推的场景编辑^①

王 坚, 陈昭炯

(福州大学 数学与计算机科学学院, 福州 350108)

摘 要: 在 Hertzmann 的图像类推算法的基础上, 使用块拼贴的纹理合成算法来实现其中的点匹配算法, 提出了一个有效的场景编辑算法. 为了使合成效果更加自然, 本文通过加入一个边界宽度来改进 efros 提出的动态规划算法以及借鉴纹理传输的思想, 使得目标图的边界缝合更加自然, 实验表明结果令人满意.

关键词: 图像类推; 块拼贴; 边界宽度; 纹理传输

Scenes Editing Based on Image Analogies

WANG Jian, CHEN Zhao-Jiong

(Department of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China)

Abstract: This paper presents an effective scenes editing algorithm based on the Hertzmann's Image analogies algorithm, using the patch-based texture synthesis to replace the only points matching. With joining a said of boundary width parameter to improve the patch-based texture synthesis and taking example by the texture transfer algorithm to make synthetic effect nature, the result of the experiment is satisfactory.

Key words: image analogies; patch-based; boundary width; texture transfer

1 引言

纹理合成技术, 一直在大规模的场景生成, 真实感以及非真实感的绘制中具有广泛的应用前景, 尤其是基于单样图的纹理合成, 合成技术已经相当成熟, 合成结果也令人满意, 然而对于场景编辑的应用, 由于需要考虑结果图中场景的整体布局以及不同景物之间的边界问题, 有时纹理合成不一定可以取得令人满意的结果. 这也是纹理合成技术在场景编辑应用中效果不佳的一个因素. 针对纹理合成技术在场景编辑应用中的不足, 本文基于 Hertzmann 提出的图像类推算法, 提出一种有效的场景编辑算法.

图像类推算法^[1]是由 Hertzmann 等人提出的, 其思想主要来自纹理合成技术, 采用像素点的合成方式, 在很多应用领域中, 如艺术滤镜^[2], 纹理合成^[3], 纹理分割^[4], 纹理数字化等, 取得了不错的效果. 但最佳像素匹配的算法效率低也一直是影响图像类推算法的合

成效果的因素. 场景编辑是由用户提供一张源图, 对于源图的不同景物进行不同形状的重新设计以及排列, 生成新的场景结果图, 使得结果图中各个新设计的景物可以很自然的缝合在一起.

由于 Hertzmann 并没有提及纹理数字化的具体实现算法, 国内外对于图像类推算法^[5-7]以及纹理数字化的相关应用研究^[8]并不多, 而且图像类推算法主要采用像素点的合成方法, 不仅降低合成速度, 也无法保证图像的局部信息完整性. 因而本文在图像类推算法的基础上, 采用块合成算法^[9]实现图像类推中的点匹配过程以提高合成效率, 但由于块合成效果很容易产生人工拼接痕迹, 本文对于块拼贴的缝合算法进行改进, 通过增加一个边界宽度以及采用图像融合的方法, 使得块缝合后的边界更加自然, 同时借鉴了纹理传输的思想, 将源图作为一个约束图来修改边界, 使用迭代的方式进行重新合成, 使得结果图的效果有了很大

^① 收稿时间:2012-04-25;收到修改稿时间:2012-05-25

的提高。

2 相关工作

2.1 Efros 的块缝合算法

算法的基本思想如下,从给定的样本纹理中随机抽取一个纹理块 $B1$ 放在待合成的纹理图中,然后再在给定的样本纹理中查找新的一个纹理块 $B2$,使得该纹理块放入待合成纹理图后与 $B1$ 有一定的重叠区域 E ,且重叠区域 E 的误差控制在一定的范围内;接着在重叠区域 E 中找出一条误差最小的路径作为 $B1, B2$ 的边缘,把边缘的右边部分用纹理块 $B2$ 贴入待合成图(如图 1 所示)。对于结果纹理图中未合成的部分重复上述过程,重新从给定的样本纹理图中抽取新的纹理块放入待合成区域进行缝合,直到获得整个结果纹理图。

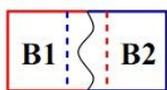


图 1 重叠区域的最优分割

误差最小的路径通以下方法进行计算:设纹理块 $B1, B2$ 具有垂直的重叠区域,重叠区域分别为 B^{ov1}, B^{ov2} , E 为重叠区域中各个像素点的误差。通过公式(1)计算重叠区域中各个像素点的误差。

$$d(p, q) = \sqrt{(R(p) - R(q))^2 + (G(p) - G(q))^2 + (B(p) - B(q))^2} \quad (1)$$

其中 p, q 为重叠区域中 $B1, B2$ 相对应位置的像素点,函数 $R(\text{pix}l), G(\text{pix}l), B(\text{pix}l)$ 分别表示该像素点红,绿,蓝三种颜色的像素值。

对于重叠区域中各个像素点的误差通过公式(2)来重新计算像素点的误差值,并记录下像素点 $\min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1})$ 的位置。

$$E_{i,j} = \begin{cases} e_{i,j} & i = 1 \\ e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}) & \text{else} \end{cases} \quad (2)$$

在最后一行的像素点的误差值中选取误差最小的像素点,通过之前记录的各个像素点的位置反向跟踪来获得最优切割路径。根据待合成纹理块位于首行时,仅存在左边有重叠区域,即垂直的重叠区域,位于首列时,仅存在上边有重叠区域,即水平的重叠区域,该重叠区域可以采用类似的方法获得最小误差路径。

除此之外,则会包含包含水平和垂直两个方向的重叠区域,当水平与垂直方向都有重叠时,两条最小误差路径会相交,此时分割边界由两条路径共同决定,在相交点选择误差最小的路径最为边界。

2.2 Ashikhmin 算法

Ashikhmin 算法是基于像素点的纹理合成算法,基本思想如下,从给定的样本纹理图中随机抽取一个像素点作为初始点,对于待合成区域的每一像素点,重复下述过程,直至获得整个结果纹理图。待合成像素点只有左边和上边相邻的四个像素点是已合成的,称为 L 邻域,考虑待合成像素点 L 邻域中的每个像素点,找到在给定的样本纹理图中的位置,偏移相应的位置后作为该合成像素点的待选像素点,通过公式(1)计算待选像素点和待合成像素点 L 邻域中各个像素点误差,选择误差之和最小的点作为当前合成像素点。

2.3 图像类推算法

Hertzmann 图像类推算法基本思想如下,定义源图像对 A 和 A' 的任一像素点用 p 表示,目标图像对 B 和 B' 的任一像素点用 q 表示。 $A(p), A'(p), B(q), B'(q)$ 分别表示源图像对和目标图像对中像素点的特征向量,一般包括像素点 RGB 值,灰度值等。该算法中还要记录与目标图像 B' 的像素点 q 最佳匹配的像素点 p 在源图像 A' 中的位置,记为 $s(q)$ 。假设源图像对 A 和 A' 中同一位置的颜色值是相关的,目标图像对 B 和 B' 中同一位置的颜色值也是相关的。算法的基本的步骤如下:

- 1) 计算图像 A, A', B 中各个像素点的特征向量,一般采用像素点的 RGB 值;
- 2) 建立图像 A, A', B 的高斯金字塔,然后在图像 B 的基础上构建目标图像 B' 的高斯金字塔;
- 3) 对于高斯金字塔每一层 l ,按照线性扫描顺序合成 B_l' 中的像素点。对于每一个像素点 q ,通过遍历 A_l 中的各个像素点,找到和 B_l 最匹配的像素点,将 A_l' 中对应的像素点 p 的特征向量拷贝到 B_l' 中像素点 q 的位置。
- 4) 按照金字塔结构,从最顶层到最底层,逐层合成 B' 的高斯金字塔,然后通过重建高斯金字塔输出目标图像 B' 。

类推算法的核心是像素点的匹配函数。对于每一个像素点 q ,分别使用 WL 算法^[10]得到全局最优的像素点 p ,得到像素点之间的差值为 $d_{app} = \|F_i(p_{app}) - F_i(q)\|^2$,使用 Ashikhmin 算法^[11]找到近似最优的像素

点, 计算近似匹配点之间的误差 $d_{coh} = \|F_1(p_{coh}) - F_1(q)\|^2$, 通过设置一致性参数 K , 若 d_{coh} 小于 $d_{app}(1+2^L * K)$, L 表示金字塔的总层数, 则将像素点 p_{coh} 的特征向量赋予像素点 q , 否则选取 p_{app} 像素点的特征向量.

3 本文的算法思想

由于 Hertzmann 只是将类推的算法简单的应用于纹理数字化中, 并没有提及具体的算法, 本文在 Hertzmann 算法的基础上, 针对纹理数字化的应用, 提出了一个有效的场景编辑算法, 使得用户可以根据需要进行不同的设计, 并根据设计图得到场景编辑的结果图.

算法的基本思想: ①将用户提供的源图 A 分割为若干区域, 不同区域用不同的颜色标记, 得到表达示意图 A' ; ②用户根据表达示意图, 对于不同的区域重新设计, 并使用相应的颜色进行标记, 得到用户的设计示意图 B ; ③对用户设计的示意图 B 中的每一个子块, 应用块缝合算法合成, 得到初始化的合成结果图, 同时为了合成效果更好, 对块缝合算法进行改进; ④对于初始化的结果图借鉴纹理传输的思想, 从源图中寻找更为匹配的边界, 改善合成结果;

3.1 改进的块缝合算法

为了改进结果图的合成效果, 本文提出一种改进的块拼贴算法, 使得边界缝合更加自然. 在 efros 提出的块缝合算法的基础上, 通过增加一个重叠区域的边界宽度 n , 在重叠区域 E 计算出一条更适合的缝合线, 然后借鉴图像融合中的思想, 以该缝合线为中心, 对左右两边的重叠区域进行选择加权平均. 算法基本思想如下:

首先计算重叠区域的边界宽度 n 的选取. 设定一个阈值 T , 作为边界宽度的阈值. 对于阈值范围内的不同宽度, 计算出重叠区域的相应的方差, 然后选取方差最大的宽度作为最后合成的宽度值 n .

使用块缝合算法时, 通过选取的最适合的 n 值并使用公式(3)来代替公式(2), 重新计算重叠区域中像素点的误差值

$$E_{i,j} = \begin{cases} e_{i,j} & i = 1 \\ e_{i,j} + \min(E_{i-1,j-n}, \dots, E_{i-1,j}, \dots, E_{i-1,j+n}) & \text{else} \end{cases} \quad (3)$$

根据改进的公式得到重叠区域的误差值后, 先在最后一行的所有误差值中选取误差最小的一个像素点

后, 通过反向跟踪来获得误差最小的路径(改进的最优路径示意图如图 1 所示). 根据得到的最优分割路径采用之前的块缝合算法继续合成.

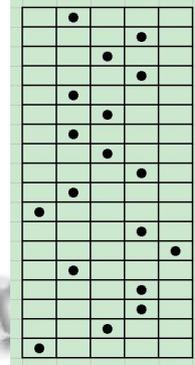


图 1 改进的最优路径示意图

为了使缝合以后边界更加平滑和自然, 可以在找到边界缝合线, 以该边界缝合线为中心, 对重叠的纹理块进行选择加权平均, 改善纹理合成中的边界不匹配问题.

3.2 纹理传输算法

由于结果图中各个子块是分别采用块拼接技术进行合成, 边界很容易产生不匹配. 为了使边界更加自然, 本文借鉴纹理传输的思想^[12], 将源图 A 作为约束图, 使得合成结果图可以参照源图的边界效果, 使得各个块之间的合成更加自然. 首先设置块的大小为 $w * w$, 将初始化的合成结果图分割为 $w * w$ 块的大小, 相邻块之间的重叠宽度为 $w/3$, 对于每一个子块, 只需在源图 A 中寻找比较接近的子块, 而不需要最相似子块. 这样不仅可以减少搜索范围, 提高运行效率, 同时还可以增加纹理传输的多样性, 使得最终结果更丰富. 然后将搜索到的子块代替合成结果图中对应子块的区域, 重叠区域仍然采用块拼贴算法.

为了是边界更加匹配, 我们对块拼贴纹理合成算法中的匹配约束进行修改, 使其为一个加权和, 即 λ 倍块重叠区域匹配误差加上 $1 - \lambda$ 倍源纹理块和当前目标图像位置处纹理块对应图像素之间的 F 范数计算公式如式(4):

$$\text{error} = \lambda \|F^{ov}(p) - F^{ov}(q)\| + (1 - \lambda) \|F(p) - F(q)\| \quad (4)$$

其中 $F^{ov}(p)$, $F^{ov}(q)$ 分别表示源图和结果图的纹理块的重叠区域, $F(p)$ 表示源图中抽取的纹理块, $F(q)$ 表示结

果图中待合成的纹理块, $\|\cdot\|$ 表示 F 范数. 通过选取不同的权值 λ , 可以得到不同结果, 它是合成结果图与近似源图的效果一个平衡. 如果一次的传输不能取得满意的结果, 我们采取迭代的方法进行继续优化, 每次将块的大小增大为原来的 $4/3$, 同时增大相应的权值, 继续按照之前的算法进行合成.

4 本文的算法步骤

场景编辑的基本过程: 由用户提供源图 A , 通过纹理分割将源图分割为若干个区域, 每个区域用一种颜色进行表示, 形成源图的表达示意图 A' ; 用户根据表达示意图进行设计, 对于不同的区域使用不同的颜色进行设计, 得到设计示意图 B , 依照用户的设计示意图, 通过纹理合成的相关算法, 生成结果图 B' . 具体的算法步骤为:

Step1: 对于用户提供的源图按照需要进行相应的分割, 不同的分割区域用不同的颜色进行表示, 得到源图的表达示意图;

Step2: 用户根据自己的需要进行设计, 按照表达示意图使用不同的颜色设计不同的形状, 并对这些区域进行颜色标记, 得到设计图;

Step3: 对于设计图的不同颜色区域, 使用块缝合算法进行合成: 先根据表达示意图找到源图中对应的区域, 以这些区域的图像作为样本, 利用改进的块缝合算法进行合成, 得到初始化的结果图;

Step4: 借鉴纹理传输的思想, 将初始化的结果图分割为若干子块, 对于每一子块在源图中寻找比较相似的子块, 将该子块拷贝到初始化结果图中, 使用块拼接算法合成;

Step5: 如果合成结果自然, 则输出结果图, 否则转入 **Step6**;

Step6: 如果合成效果边界仍然明显, 则修改块的大小以及相应的权值, 转入 **Step4**.

5 实验结果

利用本文的算法, 我们进行了相关的场景编辑实验, 在 P1.18GHz/1.0GB 的 PC 机上, 使用 Matlab 7.0 完成了改进的动态规划的块缝合算法以及相关的图像类推算算法. 算法中块的大小初始化为 70×70 , 相邻块之间的重叠宽度为 23, 动态规划使用的边界宽度为 5.

实验结果如下:

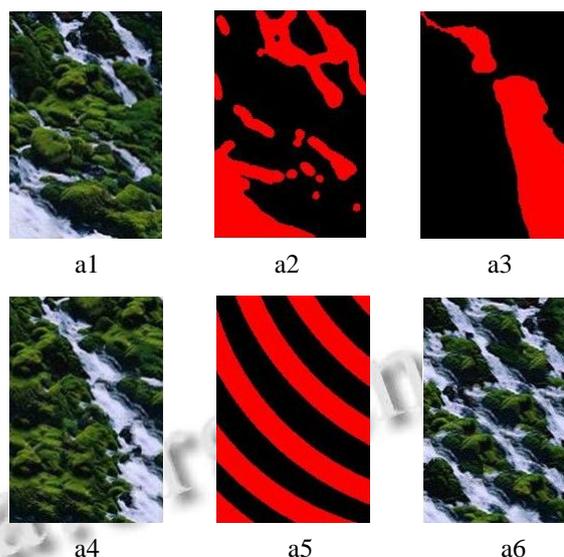


图 2 实验结果 1

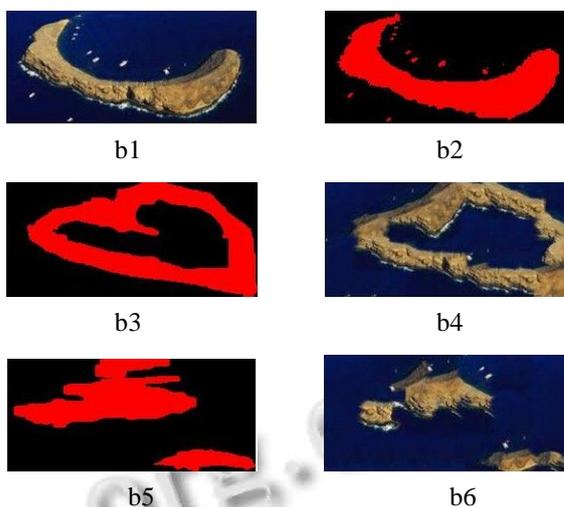


图 3 实验结果 2

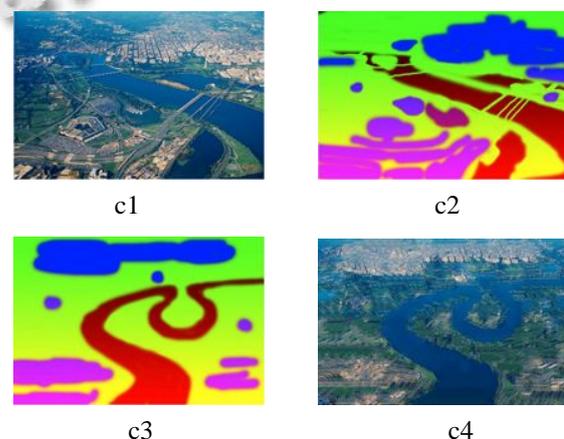


图 4 实验结果 3

在实验结果 1 中, 图像大小为 150×220 像素, a1 为

源图, a2 为源图经过纹理分割得到的表达示意图, 图中红色表示水流, 黑色表示青苔岩石, 用户可以模拟瀑布或者其他自然景观进行设计, 得到设计示意图, 如 a3,a5 等, 合成结果图分别为 a4,a6; 实验结果 2 中, 图像大小为 260*120 像素, b1 为源图, b2 为源图的表达示意图, 图中红色表示岛屿, 黑色表示海洋, 用户可以模拟珊瑚礁的形状进行设计, 设计示意图为 b3,b5, 合成结果分别为 b4, b6. 从图中可以看出, 青苔和流水之间或者岛屿和海洋之间几乎看不到人工拼接的痕迹; 合成结果一样保持着用户设计的基本形状, 同时保持了纹理局部信息完整性. 实验结果 3 中, 图像大小为 200*134 像素, c1 为源图, c2 为表达示意图, 对于同一区域增加梯度图后使用一种渐变的颜色进行表达, c3 为设计图, c4 为结果图, 从图中可以看出, 对于源图分割的结果越精细, 不仅有利于用户设计, 也可以使算法在一些复杂的图像编辑中取得不错的结果. 实验结果表明, 本文的算法可以根据用户的设计进行场景编辑, 合成结果在保持用户设计的基本形状的情况下, 边界缝合自然.

6 结语

本文在 Hertzmann 提出的图像类推算法的基础上, 提出了一个有效的场景编辑算法, 实验表明, 本文的算法可以取得令人满意的结果. 在今后的工作中需要进一步研究的工作包括以下几个方面: 1) 针对源图像对的预处理, 提出一种自适应的纹理分割算法, 同时可以为每一个子块计算梯度图, 梯度可以使用同一颜色的深浅进行表示, 这样使分割结果更细致, 更方便于用户设计, 合成效果也更加逼真; 2) 将块缝合算法和点合成算法相结合, 在场景设计过程中, 对于大块的子块采用块合成算法, 而对于比较小块子块以及编辑改用点合成方式, 这样不仅可以更好的保持用户的设计形状, 也可以从源图中获取更适合结果图的边界.

参考文献

- 1 Hertzmann A, Jacobs C E, Oliver N, et al. Image analogies. Proc. of the 28th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 2001: 327-340.
- 2 陈海旭,李文辉,张岩,孟宇,王彤.用图像类比方法实现艺术风格学习.吉林大学学报(理学版),2006,44(2):237-241.
- 3 朱文浩,魏宝刚.基于样本的纹理合成技术综述.中国图象图形学报,2008,13(11):2063-2069.
- 4 沈海.一种基于类推思想的图像分割方法.计算机工程与应用,2006,42(9):45-47.
- 5 古元亭,吴恩华.基于非线性卷积的可控图像类推和自类推技术.计算机辅助设计与图形学学报,2008,20(03):332-336.
- 6 李民,程建,汤万琼.基于学习字典的图像类推方法.计算机应用研究,2011,28(8):3171-3177.
- 7 古元亭,吴恩华.基于图像类推的超分辨率技术.软件学报,2008,19(4):851-860.
- 8 郑秋旭,陈昭炯.基于约束的多样图纹理设计与合成算法.计算机系统应用,2010,19(1).
- 9 Efros AA, Freeman WT. Image quilting for texture synthesis and transfer. SIGGRAPH 2001, Proc. of the 28th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press, 2001. 341-346.
- 10 Wei LY, Levoy M. Fast texture synthesis using tree-structured vector quantization. SIGGRAPH'00, Proc. of the 27th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison Publishing Co. 2000. 479-488.
- 11 Ashikhmin M. Synthesizing Natural Textures. 2001 ACM Symposium on Interactive 3D Graphics, 2001,(3): 217-226.
- 12 Efros AA, Leung TK. Texture synthesis by non-parametric sampling. Proc. of International Conference on Computer Vision, Kerkyra: IEEE Computer Society, 1999: 1033-1038.