

Image Completion with Structure Propagation [1]

——《计算机应用数学》读书报告

10421038 董子龙

浙江大学 CAD&CG 实验室

浙江大学计算机系

摘要: 报告总结了[1]的算法流程, 设计动态编程, 信任传播, 泊松方程; 同时补充说明 Texture-by-numbers 纹理合成方法[2][3][4]。

1. 简介

Image Completion 也称为 Image Inpainting, 主要功能是利用已知的图像信息, 填充未知的区域像素。已有的方法涉及偏微分方程, 采样, 梯度光滑等技术, 已经能够较好的恢复未知区域的颜色, 保持在纹理上的一致性, 但是不能很好的保持显著的结构, 如栅栏、楼梯、窗户等。这篇文章在一定范围内解决了这个问题。

2. 基本思路

研究者基于以下两个结果:

- 1) 自然图像中的特征通常能够用曲线勾勒出来;
- 2) 在填充其它区域之前, 这些特征的内容应该首先确定。

提出了图像填充的步骤:

- 1) 用户勾勒出特征曲线, 图 1(b);
- 2) 结构扩散, 根据曲线确定特征, 图 1(c);
- 3) 纹理扩散, 填充其它区域, 图 1(d)。

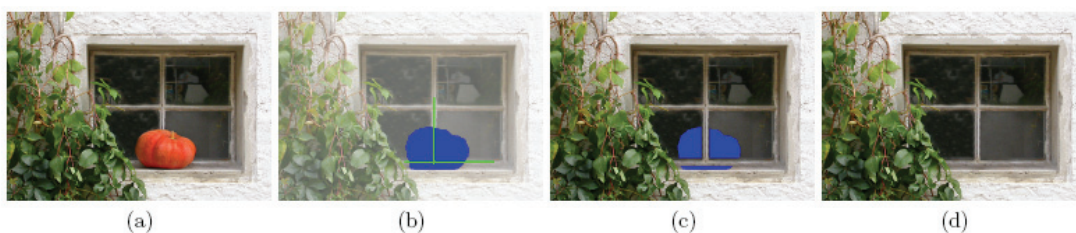


图 1

3. 全局优化模型

研究者建立了一个全局优化的模型, 以最小化合成面片的重叠区域的颜色差异。以最简单的一条曲线为例, 如图 2 :

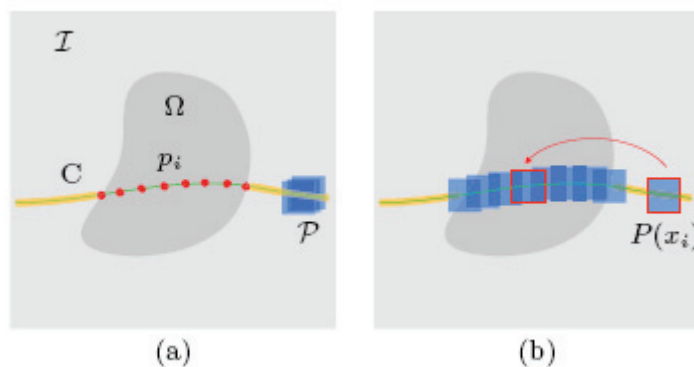


图 2

已知和未知区域分别用 I 和 Ω 表示, C 是用户画的特征曲线, 在 Ω 中对 C 稀疏采样得到

L 个锚点 $\{p_i\}_{i=1}^L$ ，作为合成面片的中心。这些点相邻之间两两相连，构成一个单链图 $G = \{V, \varepsilon\}$ ，V 就是 L 个锚点的集合， ε 是连接相邻节点的边集合。在 Ω 外，沿着 C 的一条窄条对已知区域采样得到样本面片集合 $P = \{P(1), P(2), \dots, P(N)\}$ ，如果结构样本不足够，可以通过旋转获得新的样本。于是问题就转化为，为 L 个合成面片选择相应的样本源面片，即求取路径 $X = \{x_i\}_{i=1}^L$ ，使

$$E(X) = \sum_{i \in V} E_1(x_i) + \sum_{(i,j) \in \varepsilon} E_2(x_i, x_j), \quad (1)$$

其中

$$E_1(x_i) = k_s \cdot E_S(x_i) + k_i \cdot E_I(x_i). \quad (2)$$

最小。 E_S, E_I, E_2 分别代表结构，完整，耦合约束； k_s, k_i 是调节权值。 E_S 是源面片与合成面片结构上的差异，即所包含的特征曲线之间的差异。

$$E_S(x_i) = d(c_i, c_{x_i}) + d(c_{x_i}, c_i), \quad (3)$$

d 函数计算第一个曲线参数上的每一个点到第二个曲线参数上的距离之和，如图 3(a)。

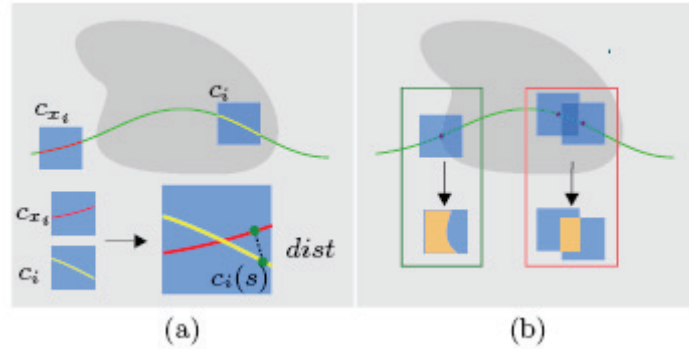


图 3

E_I 约束与 I 存在交集的合成面片的源面片在交集上必须一致，即在图 3(b) 绿色框中的橙黄色部分。 E_I 等于交集上的归一化平方差(SSD)的和。

E_2 指明了相邻的两个合成面片之间的重叠区域的差异，即在图 3(b) 红色框中的橙黄色部分。同样是计算归一化平方差(SSD) 的和。

4. 动态编程——单曲线的结构扩散

对于只有一条曲线的情况，可以看作单链图求解一个最小值路径的动态编程问题。定义 $M_i(x_i)$ 为从 1 到 i 所有可能的 x_i 的积累最小值，动态编程遍历 2 到 L ，递归计算所有路径的

$M_i(x_i)$ ：

$$M_i(x_i) = E_1(x_i) + \min_{x_{i-1}} \{E_2(x_{i-1}, x_i) + M_{i-1}(x_{i-1})\}, \quad (4)$$

其中 $M_1(x_1) = E_1(x_1)$ 。 L 的对应的 x_i 即是 $x_L^* = \arg \min_{x_L} M_L(x_L)$ ，在计算 $M_i(x_i)$ 的过程

中维护一张记录路径的表，就可以回溯得到所有的源面片。

5. 信任传播——多曲线的结构扩散

对于更加复杂的情况，单链图不足以表示，如图 4(a)。为了构造符合要求的图 G ，首先在所有的曲线交点上设置锚点，然后对剩下的曲线采样，如图 4(b)。对于这样的图，动态编程的代价非常高，所以研究者采用了信任传播。

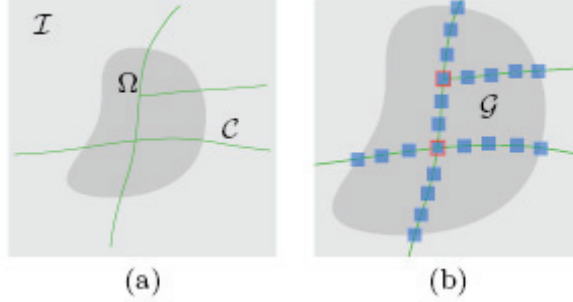


图 4

信任传播是两两相连的无向图上的局部消息传递算法，旨在使定义的Gibbs能量最小。其基本思想：图中的每一个节点从它的邻居接收消息，根据自身状态更新消息，然后把消息返还各个邻居。如下：

Algorithm 1 Completion using Belief Propagation.

- 1: Initialize all messages $M_{ij}^0 = 0$ between any two adjacent nodes i and j in graph \mathcal{G} .
- 2: Update all messages M_{ij}^t iteratively from $t = 1$ to T :

$$M_{ij}^t = \min_{x_i} \{E_1(x_i) + E_2(x_i, x_j) + \sum_{k \neq j, k \in \mathcal{N}(i)} M_{ki}^{t-1}\} \quad (5)$$

where $\mathcal{N}(i)$ are all adjacent nodes of i .

- 3: Compute optimal label x_i^* for each node i :

$$x_i^* = \arg \min_{x_i} \{E_1(x_i) + \sum_{k \in \mathcal{N}(i)} M_{ki}^T\} \quad (6)$$

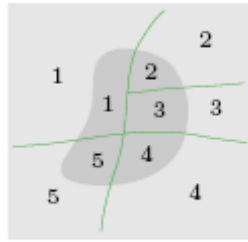
M_{ij} 是从 i 节点传给 j 节点的消息，包含 N 个元素的列向量， N 是样本源面片的个数，表示 j 节点对应 x_j 的可能性，在算法实现过程中，只保存最大可能值。公式(5)就是消息传递的循环过程，研究者利用带负号的 \log 函数把概率转换成和的最小值，事实上是求最大积 (max-product)。

公式(5)的具体意义：对所有可能的 x_i ，计算相对于 j 节点的 E 能量，加上 i 节点的除 j 之外的所有相邻节点传给的最大可能值，选择使这个值最小的 x_i ；在下次循环中，这个选择出来的 x_i 又自动影响 j 节点的邻居。

6. 纹理扩散

结构扩散完成后，继续填充剩余的区域，一般来说，这些结构曲线会把 Ω 分成不相连的子区域，而且每一个子区域都相邻 I 的一个子区域，研究者将这些两两对应的子区域编号，如图 5，然后应用 Texture-by-numbers 技术，使纹理从 I 扩散到 Ω 。

子区域处理的次序根据信任图(Confidence map)确定，是一种水平集方法，从最靠近已知区域的像素开始处理，向内扩散。



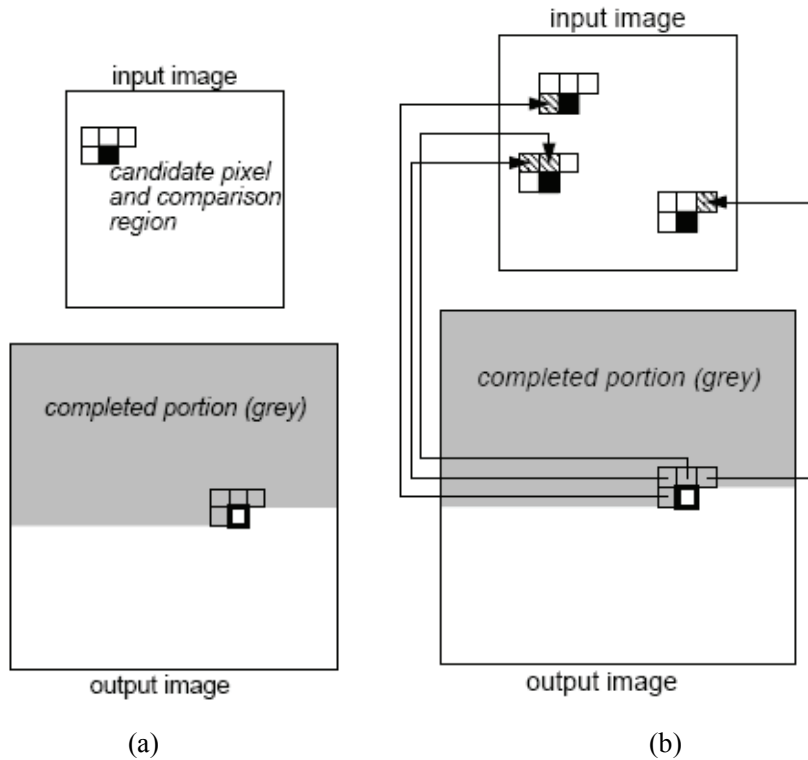
(a)

图 5

6.1 Texture-by-numbers

WEI 和 LEVOY[2]提出纹理合成的 WL 算法：沿着扫描线，对输出图像的每一个像素建立一个 L 型的相邻像素区域，然后对输入图像的每一个像素比较 L 区域，选择差异最小的像素，如图 6(a)。算法的缺陷在物体边界形状的模糊。

Michael [3]利用相邻像素之间的耦合性，改进 WL 算法，在速度和边界保持上有很大的改进：同样建立 L 区域，但是对源像素的选择局限在 L 区域的每一个像素的源像素的相应邻像素上，如图 6(b)。缺点是源像素基本上从上到下选取，当到达底部时，重新开始的策略决定比较麻烦。



(a)

(b)

图 6

Hertzmann 等把改进的 WL 方法引入[4]的框架中,其中一个应用就是 Texture-by-numbers (TBN)方法。TBN允许用户从标记过的图像样例中学习数据,然后根据新的标记合成新的图像,如图7。 A 和 A' 是图像样例, A 是 A' 对应的标记, B 是根据新的标记 B 合成的图像。

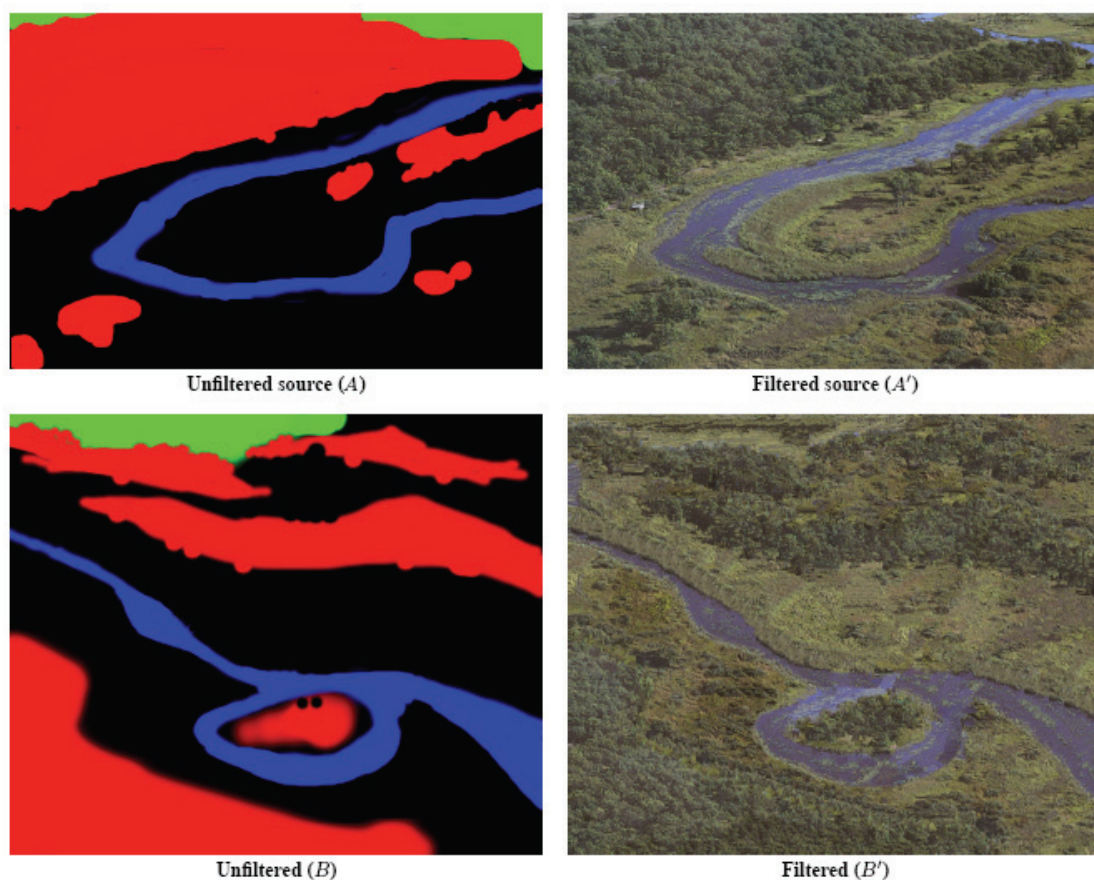


图 7

7. 光度校正

为了保证重叠合成面片之间的接缝的一致连贯,如图 8 中黑色线段,研究者提出光度校正法,即在狄利克雷边界条件下,从校正的梯度场解泊松方程。

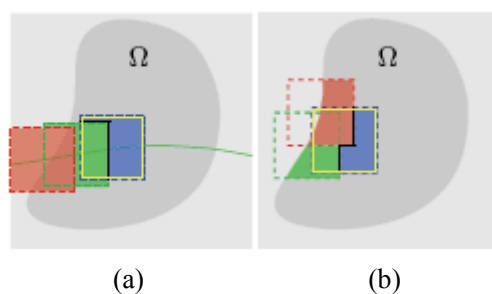


图 8

假设红色和绿色面片已经合成完毕,直接从源片面复制像素颜色到蓝色的面片,生成 J ,在 J 上构建一个二元遮罩 B_M ,如果在重叠区域,设为 0,反之为 1。然后根据 ∇J 计算 ∇J^* :

$$\nabla J_x(x, y)^* = \begin{cases} \nabla J_x(x, y) & B_M(x, y) = B_M(x + 1, y) \\ 0 & B_M(x, y) \neq B_M(x + 1, y) \end{cases}$$

这个过程相当于在接缝处消除变化，保持一致。图中的黄色边框内部就是狄利克雷边界条件，像素的每个颜色通道分别在 ∇J^* 上解泊松方程得到 J^* ，就是最终的蓝色面片。

图 9 表明光度校正法的效果，左下没有应用，右下经过校正效果就完全不一样了。



图 9

8. 文献

- [1] Jian Sun, Lu Yuan, Jiaya Jia, Heung-Yeung, Shum, Image Completion with Structure Propagation, *Proceedings of SIGGRAPH 2005*.
- [2] WEI, L.-Y., AND LEVOY, M. Fast texture synthesis using tree-structured vector quantization. *Proceedings of SIGGRAPH 2000* (July 2000), 479–488.
- [3] Michael Ashikhmin. Synthesizing Natural Textures. *ACM Symposium on Interactive 3D Graphics*, pages 217–226, March 2001.
- [4] Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. Image analogies. *In Proceedings of ACM SIGGRAPH 2001*, 327–340.