Poisson image editing

Dezeming Family

2023 年 2 月 20 日

DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [https://dezeming.top/] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

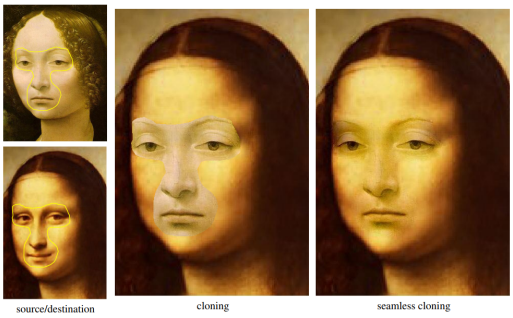
目录

一　引文

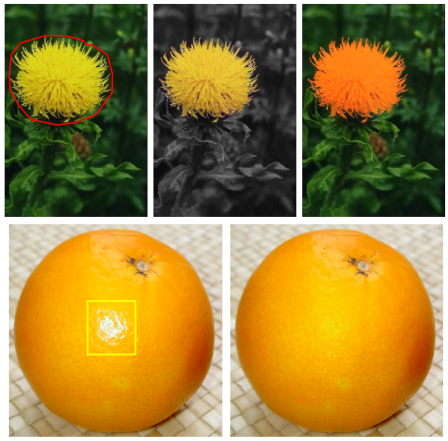
图像编辑任务涉及全局更改（颜色/强度校正、过滤器、变形）或局限于选择区域的局部更改。在这里，我们感兴趣的是以无缝和轻松的方式实现局部更改，这些更改仅限于手动选择的区域，变化的范围从轻微改变到完全被新内容取代。

使用基于求解泊松方程 (Poisson equations) 的通用插值机制，引入了多种新的工具来无缝编辑 (seamless editing) 图像区域。

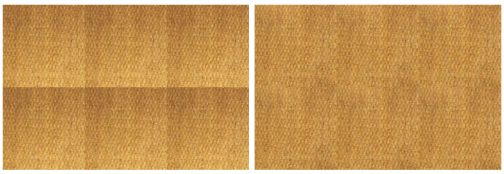
第一组工具允许将不透明和透明源图像区域无缝地导入到目标区域：



第二组工具也是基于类似的数学思想，允许用户在选定区域内无缝地修改图像的外观。可以安排这些更改以影响区域中对象的纹理、照明和颜色：



或使平铺成为矩形：



要实现这一点的经典工具包括仅限于选定内容的图像滤波器（用于轻微更改，比如给新内容一个权重，旧内容一个权重，然后混合），以及使用克隆工具进行交互式剪切和粘贴（用于完全替换）。使用这些经典工具，选定区域中的更改会导致可见接缝。通过沿选定区域的边界羽化（设置边界部分的混合比与中间部分不同），只能部分隐藏这些接缝：

我们提出的一种通用机制，从中可以派生出用于无缝编辑和克隆选择区域的不同工具。该方法的核心数学工具是具有狄利克雷边界条件的泊松偏微分方程 (Poisson partial differential equation with Dirichlet boundary conditions)，它指定了感兴趣域上未知函数的拉普拉斯算子，以及域边界上的未知函数值。

动机是双重的：首先，心理学家 [Land and McCann 1971] 知道，拉普拉斯算子抑制的缓慢强度梯度 (slow gradients of intensity) 可以叠加在图像上，几乎没有明显的效果。相反，拉普拉斯算子提取的二阶变化在感知上是最显著的。其次，有界域上的标量函数由边界上的值和内部的拉普拉斯算子唯一定义。因此，泊松方程有一个唯一的解，这可以导出一个完善的算法。

因此，给定在某个域上构造未知函数的拉普拉斯算子的方法及其边界条件，泊松方程可以通过数值求解来实现该域的无缝填充。这种方式可以在彩色图像的每个通道中独立复制。

求解泊松方程也有一种作为最小化问题的替代解释：它计算在给定边界条件下，在 L_2 范数中，其梯度最接近某个指定向量场（引导向量场 (guidance vector field)）的函数。以这种方式，重构函数向内插入边界条件，同时尽可能密切地跟踪引导场的空间变化。论文 [1] 第 2 节详细介绍了这种引导插值。

我们将研究制 guidance vector field 的多种可能选择。我们特别指出，这种插值机制在易用性和功能方面利用了经典的克隆工具。生成的新克隆工具允许用户无缝地删除和添加对象。通过适当地混合源图像的梯度与目标图像的梯度，还可以令人信服地添加透明对象。此外，具有复杂轮廓（包括孔）的对象可以自动添加，而无需费力地裁剪。论文 [1] 第 3 节介绍了这些不同的克隆工具方法。

如论文 [1] 第 4 节所示，同样的机制也可用于修改受限域内图像的外观，同时避免域边界上的可以看到的不连续性。特别是，可以很容易地修改对象的颜色、纹理或照明，而无需精确描绘对象边界。此外，还可以无缝地平铺矩形图像区域。

1 1　相关工作

泊松方程在计算机视觉中得到了广泛的应用，它很自然地作为解决某些变分问题的必要条件而出现。由于年代较久远，相关工作中的很多技术我都没有了解过。

在图像编辑应用的特定背景下，前面的三项工作与这里提出的泊松方程的使用有关，比如图像混合插值以及图像克隆区域时的边界无缝衔接。

在 [Fattal 等人 2002] 中，高动态范围（HDR）图像的梯度场被非线性地重新缩放，产生不再是梯度场的矢量场。然后，通过求解泊松方程获得新图像，其中该向量场的散度为 right-hand-side，并且在 Neumann 边界条件下，指定新图像在垂直于边界的方向上的梯度值为零。

相反，我们这里提出的方法可以应用于从图像中选择的任意小块，而不仅仅是整个图像。为了做到这一点，矩形轮廓上的 Neumann 边界条件必须由任意轮廓上的 Dirichlet 条件代替。进一步的概括是扩展应用于梯度的非线性操作的范围，以包括最大操作 (maximum operations) 和小梯度的抑制 (suppression of small gradients)，这两者都具有有用的编辑功能。

在 [Elder 和 Goldberg 2001] 中，引入了一种通过稀疏的边缘元素集（边缘）编辑图像的系统。要抑制对象，将删除关联的边；为了添加对象，将合并关联的边缘以及每个边缘两侧的颜色值。然后通过平滑地插值与新的边缘集合相关联的颜色来获得新的图像。这相当于求解拉普拉斯方程（右手边为空的泊松方程 (a Poisson equation with a null right hand side)），Dirichlet 边界条件由边缘周围的颜色给出。编辑边缘和相关颜色并不总是简单的。此外，当转换到轮廓域或从轮廓域转换时，图像细节会丢失，这可能是不希望的结果。与基于小波极值的相关表示相比，基于稀疏边缘的表示确实是不完整的 [Malat 和 Zhong 1992]，基于小波极值的相关表示是完整的，但不太适合手动编辑。

在 [Lewis 2001] 中，通过从选定区域中的细节中分离出亮度分量并通过选择边界处的亮度的谐波插值（求解拉普拉斯方程）替换亮度，以从毛发图像中去除斑点。

第二种技术是 [Burt 和 Adelson 1983] 中提出的多分辨率图像混合，其思想是使用感兴趣图像的多分辨率表示，即拉普拉斯金字塔。源图像区域的内容在每个分辨率带内独立地与其目标图像中的新环境混合。然后通过将由此获得的新合成拉普拉斯金字塔的不同级别相加来恢复最终合成图像。该技术产生了多分辨率混合，其中最精细的细节在选择的边界附近非常局部地平均，而较低的频率在这些边界周围更大的距离上混合。

这种快速技术实现了源拉普拉斯算子在目标区域的近似插入（在拉普拉斯金字塔的第一层上），而我们通过泊松方程的解精确地执行该拉普拉斯算子插入。更重要的是，多分辨率混合通过金字塔的上层将来自远处源和目标像素的数据合并到最终合成图像中。这种长距离混合可能是不可取的，在我们的技术中也

不会发生。此外，除了不透明的无缝克隆之外，我们的系统还提供了扩展功能，参见论文 [1] 第 3 节和论文 [1] 第 4 节。

我们提出的引导插值框架 (guided interpolation framework)，其中引导由用户指定。目前研究中已经有不少插值方法来仅使用边界条件的知识自动填充图像区域。第一类此类方法由修补技术组成 [Ballester 等人，2001；Bertalmio 等人，2000]，其中设计了基于 PDE(偏微分方程) 的插值方法，以使等值线继续撞击选定区域的边界。要求解的偏微分方程比泊松方程更复杂，并且只适用于在相对无纹理的区域中弥合相当窄的间隙。基于示例 (Example-based) 的插值方法 [Barret 和 Cheney 2002；Bornard 等人 2002；Efros 和 Leung 1999]，其中使用许多小补丁的排列来合成新图像区域，这是修复方法的有趣的替代方案。这些方法以更令人信服的方式处理大洞和纹理边界。此外，它们还可以用于导入纹理，如 [Efros 和 Freeman 2001；Hertzmann 等人 2001] 所示。

二　引导插值的泊松解

本节我们描述用 guidance vector field 来进行图像插值的方法。由于每个颜色通道可以分开独立进行，所以我们用单通道图来描述。论文公式如图：

$$\min_f \iint_\Omega |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (1) \qquad \Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (2)$$

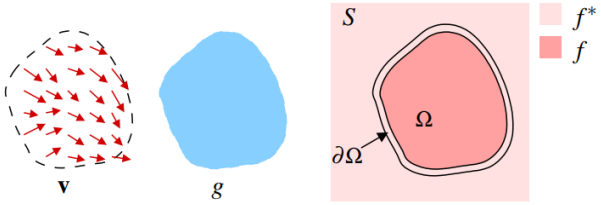
$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (3) \qquad \Delta f = \operatorname{div}\mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}, \quad (4)$$

$$\Delta \tilde{f} = 0 \text{ over } \Omega, \ \tilde{f}|_{\partial\Omega} = (f^* - g)|_{\partial\Omega}. \quad (5) \qquad |N_p| f_p - \sum_{q\in N_p} f_q = \sum_{q\in N_p} v_{pq}. \quad (8)$$

$$\min_{f|_\Omega} \sum_{\langle p,q\rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^*, \text{for all } p \in \partial\Omega, \quad (6)$$

$$\text{for all } p \in \Omega, \quad |N_p| f_p - \sum_{q\in N_p\cap\Omega} f_q = \sum_{q\in N_p\cap\partial\Omega} f_q^* + \sum_{q\in N_p} v_{pq}. \quad (7)$$

下图描绘了符号标识，S 表示 $\mathbb{R}^2$ 上的封闭子集（图像定义域），Ω 是 S 上的一个封闭子集，边界为 $\partial\Omega$。f^* 是定义在 $S-\Omega$ 区域的已知标量函数，f 是定义在 Ω 内部区域的未知标量函数；令 $\mathbf{v}$ 是定义在 Ω 上的向量场。



这里的标量函数可以理解为图像的每个像素值（单通道所以是标量）。Ω 是要被融合的前景区域，$S-\Omega$ 是背景 S 中与 Ω 没有任何重叠的部分，该值是已知的。我们想把前景融合到图像 Ω 区，但是我们不知道融合后的结果应该是什么比较合适，但我们知道边界处的值应该是 $f^*|_{\partial\Omega}$，而且对于 Ω 内部的函数 f，在边界处的值 $f|_{\partial\Omega}$ 也必须要等于 $f^*|_{\partial\Omega}$，这样融合的结果才没有缝隙。图中的 g 表示前景区域的原函数值，是还没有被融合进去的结果。我们下面按照论文的方式来描述。

要在域 Ω 上插值 f 到 f^* 中，最简单的方法是 membrane interpolant，即最小化公式 (1)。公式 (1) 中，$\nabla . = [\frac{\partial .}{\partial x}, \frac{\partial .}{\partial y}]$ 是一个求梯度算子。该算子可以看出，在边界处两个函数值是相同的。这个最小化的目标可以解释为：在保证边界处两个函数值相同的前提下，使得函数 f 的梯度幅值尽可能小。

这个最小化器必须要满足对应的 Euler-Lagrange 方程，见公式 (2)。其中，$\Delta . = \frac{\partial^2 .}{\partial x^2} + \frac{\partial^2 .}{\partial y^2}$ 表示 Laplacian 算子，也就是说公式 (2) 是有 Dirichlet 边界条件 (boundary conditions) 的 Laplacian 方程。

对于图像编辑任务，这种简单的方法通常不会得到使人满意的插值结果，而且还会有模糊（毕竟这里限制了被插入的图像的梯度 ∇f，而且要使得拉普拉斯算子为 0）。也有不少人改进了这种方案，我们论文中提出的路线是通过引入引导场形式的进一步约束来修改问题。我们知道，泊松图像编辑的目的是保留源图像的纹理，无缝融入到新图像中。那么一个切实可行的方案就是将源图像的梯度保留，应用到目标图像的边界中，解出同时满足梯度和边缘约束条件的方程，得到目标区域像素。

我们已经知道了前景图像的原来的值，那么我们就能求出前景图像的梯度 $\mathbf{v}$，我们称其为引导梯度场，这样就可以得到新的最小化公式 (3)。公式 (3) 的解是有 Dirichlet 边界条件的泊松方程的唯一解，见公式 (4)，其中，$div\mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} = \Delta g$ 是 $\mathbf{v} = (u, v)$ 的散度 (divergence)。也就是说，根据公式 (4)，在保证边界值对应相等的前提下，f 的 Laplacian 要和 g 的 Laplacian 相等。

这是彩色图像泊松编辑的基本机制：公式 (4) 的三个泊松方程在所选颜色空间的三个颜色通道中独立求解。本文中报告的所有结果都是在 RGB 颜色空间中获得的，但例如在 CIE 实验 (CIE-Lab) 中也获得了类似的结果。

当引导场 $\mathbf{v}$ 是保守的 (conservative)（即它是函数 g 的梯度）时，可以用另一种有用方法，在 Ω 上定义校正函数 $\widetilde{f}$ 来理解泊松插值，此时，$f = g + \widetilde{f}$，此时泊松方程 (4) 就变为了公式 (5) 所示的有边界条件的 Laplace 方程。根据公式 (5)，在 Ω 内，加性校正项 $\widetilde{f}$ 是一个原图像与目的图像之间沿着边界 $\partial\Omega$ 处的差值 $(f^* - g)$ 的 membrane interpolant。前面说过，由于 f 的 Laplacian 要和 g 的 Laplacian 相等（$\Delta f = \Delta g$），又因为 $f = g + \widetilde{f}$，所以可以得到 $\Delta\widetilde{f} = 0$。 在论文的 section 3 中，引导内插 (guided interpolation) 会用于无缝克隆。

2 1　离散泊松求解器

对于变分问题（公式 (3)）及其有 Dirichlet 边界条件的关联的泊松方程，可以被离散化并通过多种方式求解。

对于离散图像，这种离散化是很自然的。不失泛化性，我们用相同的符号来描述，S 和 Ω 现在表示的是定义在无限离散网格（像素网格）上的有限点集。S 不但可以包括一整幅图像，也可以是一个图像的子集（比如我们很明确前景图像只会在背景图中占很小的区域，我们就可以让 S 作为把这个小区域圈起来的框）。

对于一个 S 内的像素 p，设 N_p 是它的四连接邻域集（上下左右四个像素），设 $< p, q >$ 表示像素对 (pixel pair)，$q \in N_p$。此时 Ω 的边界 $\partial\Omega = \{p \in S \setminus \Omega : N_p \cap \Omega \neq \varnothing\}$（也就是说像素点 p 在 S 而不在 Ω 处，如果 p 的邻域与 Ω 有交集，就认为是边界）。设 p 的像素值为 f_p。

对于任意形状的 Dirichlet 边界，最好的方法就是离散化，而不是用公式 (4) 所示的泊松方程求解。因此可以得到公式 (6) 所示的离散形式，其中，v_{pq} 表示 $\mathbf{v}(\frac{p+q}{2})$ 在边 $[p, q]$ 处的投影，即 $v_{pq} = \mathbf{v}(\frac{p+q}{2}) \cdot \overrightarrow{pq}$。其解满足公式 (7) 所示的联立线性方程。

如果 Ω 包含了 S 边界上的像素，那么有的像素 $N_p < 4$（称为截断邻域 (truncated neighborhood)）。

对于在 Ω 内部的像素，$N_p \subset \Omega$，在公示 (7) 的右侧没有边界项，即对应于公式 (8)。

公式 (7) 构成了一个经典的、稀疏的 (sparse(banded))、对称的、正定的系统。由于边界 $\partial\Omega$ 可以是任意形状的，我们必须使用迭代求解器。

本文所示的结果是使用具有连续超松弛 (successive overrelaxation) 的 Gauss-Seidel 迭代或 V-cycle multigrid 计算的。这两种方法对于中等大小彩色图像区域的交互式编辑都足够快，例如，对于 60000 像素的圆盘区域，Pentium 4 上执行需要 0.4 秒。如 [Bolz 等人 2003] 所示，GPU 上的多重网格实现将为更大的区域提供解决方案。

三　无缝克隆

本节公式如下：

$$\mathbf{v} = \nabla g, \quad (9) \qquad \Delta f = \Delta g \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}. \quad (10)$$

$$\text{for all } \langle p, q \rangle, v_{pq} = g_p - g_q, \quad (11)$$

$$\text{for all } \mathbf{x} \in \Omega, \mathbf{v}(\mathbf{x}) = \begin{cases} \nabla f^*(\mathbf{x}) & \text{if } |\nabla f^*(\mathbf{x})| > |\nabla g(\mathbf{x})|, \\ \nabla g(\mathbf{x}) & \text{otherwise.} \end{cases} \quad (12)$$

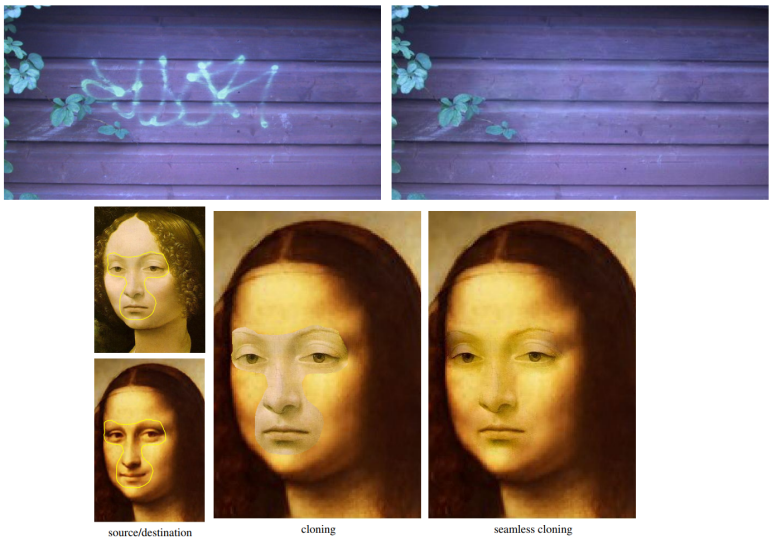
$$v_{pq} = \begin{cases} f_p^* - f_q^* & \text{if } |f_p^* - f_q^*| > |g_p - g_q|, \\ g_p - g_q & \text{otherwise,} \end{cases} \quad (13)$$

使用源图像的梯度的融合

上一节我们说过，我们希望在保证边界相同的前提下，使得前景和背景的引导场也要相同。这个引导场可以是梯度场，即公式 (9)。那么公式 (4) 就可以写为公式 (10) 的形式。从数值实现上，对于每个像素对 $< p, q >$ 而言，v_{pq} 可见公式 (11)。图像像素梯度 $[g_x\ g_y]^T = [f(x+1,y)-f(x-1,y)\quad f(x,y+1)-f(x,y-1)]^T$，代入到 $\mathbf{v}$ 中就能得到相应结果。公式 (11) 可以代入到公式 (9) 中。

由此获得的无缝克隆工具确保了前景图和背景图的边界的一致性。如下面的图所示，它可以用于隐藏不期望的图像特征或在图像中插入新的元素，但比传统的克隆更灵活和容易。从用户输入的角度来看，大多数任务只需要非常宽松的套索选择，如下图所示：

然而，当源的特征必须与目标中的相应特征对齐时，如下图中的围栏示例和面部示例，源和目标区域的定位必须更加精确：



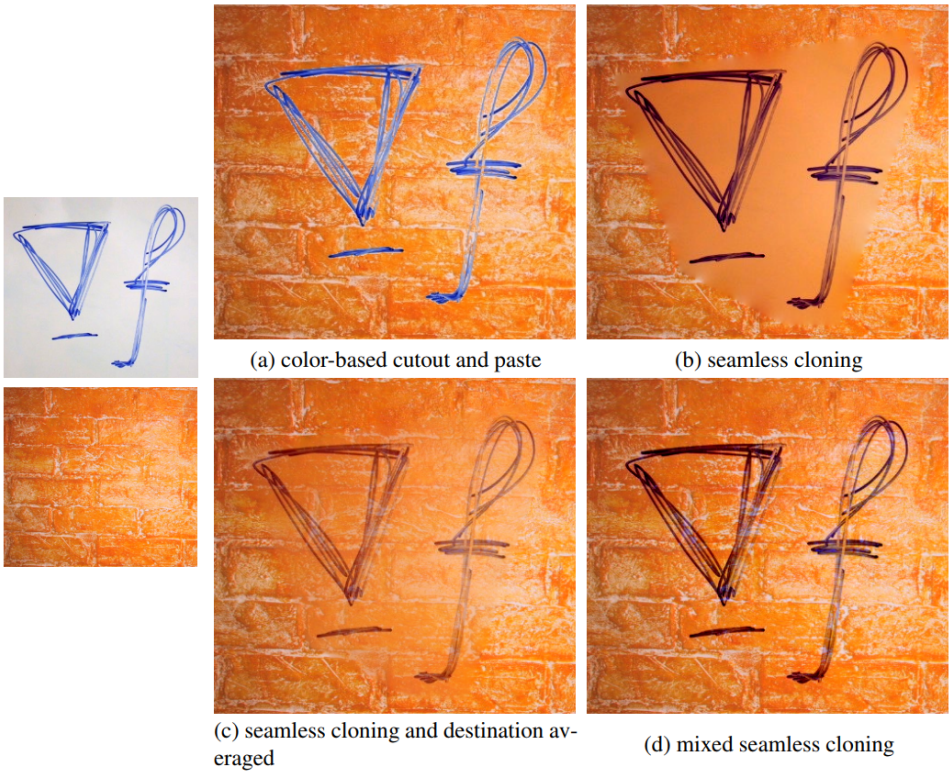
最后，在无缝克隆主要涉及纹理片段的情况下，如面部修饰示例，以及下图中纹理交换示例，重复应用宽笔刷笔划是更有效的方法。

直到插值过程引起的全局变化，源图像（前景图像）的全部内容都会被保留。在某些情况下，最好只传输部分源内容。这个问题最常见的例子是只转移源图像的强度而不是颜色，一个简单的解决方案是预先将源图像变成单色（灰度图），然后再应用泊松融合，就能只迁移图像纹理，而不迁移颜色，见下图：

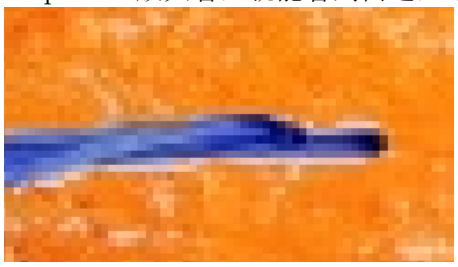
混合梯度

前面的技术不会保留背景图像 f^* 的纹理，但在一些情况下，我们希望保留目的图像的纹理，比如：

我们希望源图像的文本部分从源图像剥离，并应用于目标图像，而不需要复杂的或者精细的选择操作。一种可能的方法是将引导场 $\mathbf{v}$ 定义为源和目标梯度场的线性组合，但这会影响纹理。比如下图中的“seamless cloning and destination averaged”：

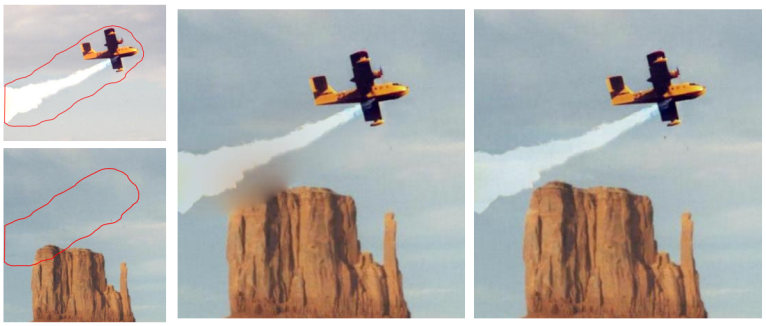


经典的方法，基于颜色的选择和阿尔法掩模可能很耗时，并且经常会留下不希望的光晕 (halo)。我们把上图中的“color-based cutout and paste”放大看，就能看到白边，这就是光晕。



由于泊松方法可以允许使用非能量保守的引导场，因此可以得到更吸引人的效果。我们保持 f^* 或 g 中变化较大的梯度，即公式 (12)，就能得到对应的引导场 v_{pq} 值，见公式 (13)。得到的克隆效果可以参考上面的“mixed seamless cloning”。

当将源图像中的一个对象添加到非常接近目标图像中的另一个对象时，这种混合无缝克隆也很有用，见下图。



source/destination　　seamless cloning　　mixed seamless cloning

四　选择编辑

前面的方法都会依赖于源图像的梯度，我们也可以只用目标图像的梯度作为引导场来做一些事情，比如 texture flattening(纹理展平), spatially selective illumination changes(改变选定区域的照明), background or foreground color modifications(修改前景或背景), and seamless tiling(无缝平铺)。

前两种，即纹理展平和改变选定区域的照明，是依赖于原始梯度场 ∇f^* 在选定区域的非线性修改。后两种，依赖于在域内部或外部对原始图像进行修改后的就地无缝克隆，以提供新的源图像，或提供新的边界条件。

公式如下：

$$\text{for all } \mathbf{x} \in \Omega, \mathbf{v}(\mathbf{x}) = M(\mathbf{x})\nabla f^*(\mathbf{x}), \quad (14)$$

$$v_{pq} = \begin{cases} f_p - f_q & \text{if an edge lies between } p \text{ and } q, \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

$$\mathbf{v} = \alpha^{\beta} |\nabla f^*|^{-\beta} \nabla f^*, \quad (16)$$

4 1　Texture flattening

给图像梯度 ∇f^* 一个稀疏过滤器，使得仅仅保留最显著的特征，见公式 (14)，其中 M 是一个二元 mask，仅仅在某些区域有值。比如 M 可以是一个边缘检测器，即公式 (15)，这样就可以起到平滑效果：

4 2　Local illumination changes

[Fattal et al. 2002] 的方法可以用于 HDR 图像，以修改动态范围。首先，对图像的对数的梯度场进行变换，以减少大的梯度并增加小的梯度。变换后的向量场 $\mathbf{v}$ 被用于重建图像的对数 f，通过求解在 Neumann 边界条件下整个图像域的泊松方程 $\Delta f = div\mathbf{v}$ 来得到。

上述方法的一个自然扩展是使用 $\partial\Omega$ 上的适当的 Dirichlet 条件将校正限制在选定区域 Ω 上，简化的引导场见公式 (16)。α 等于 0.2 乘以 Ω 区域的 f^* 的平均梯度；$\beta = 0.2$。该方法不但可以提高曝光，也可以降低镜面反射：

4 3 Local color changes

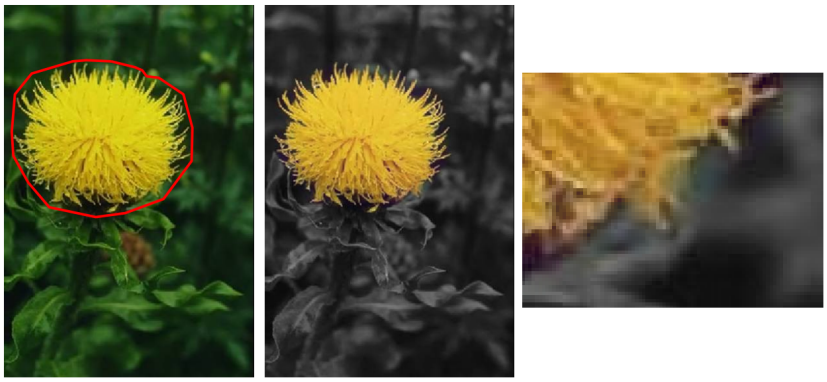
泊松编辑也可以用于修改颜色。给定一个原始的彩色图像，选择一个区域 Ω，两种不同的颜色变换可以无缝混合：一个版本提供了在 Ω 外的目的函数 f^*，另一个版本提供了在 Ω 内要被通过公式 (10) 修改的源图像 g。

例如，将图像中除某个感兴趣的对象外的所有对象都变成单色的任务通常通过精确选择一个对象，然后将其补色设置为单色 (setting its complement to monochrome) 来执行。相比之下，Poisson 编辑将用户从繁琐的精确选择中解放出来：

- 给定一个源彩色图像 g，目的图像 f^* 被设置为 g 的亮度 (luminance)（意味着目的图像 f^* 被设置为了三通道都是相同值的图像，因此表现为灰度图像）。
- 用户选择包含对象的区域 Ω，这可能比实际对象稍大。
- 对每个通道都求解泊松方程 (10)。（意味着如果我们要把前景图像无缝地贴到背景上，要保证前景梯度被保留下来。）

比如下面的图，左边选择的区域是前景 g，右边是目的图 f^*：

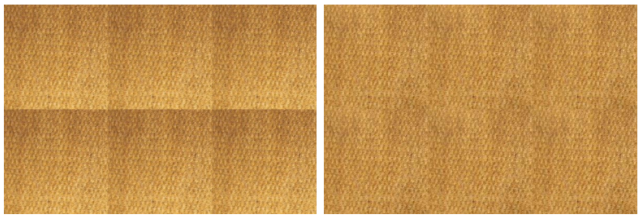
结果如下。注意，尽管结果似乎还免费提供了对象的精确分割，但事实并非如此，因为对象外部存在一些目标图像的残留污染。放大后可以看到一些绿色的痕迹：



也可以改变前景图的颜色。将前景图原图 g 的每个通道分别乘以 1.5，0.5 和 0.5，得到 g'；目的图 f^* 就是前景图 g。然后求解泊松方程，就可以只改颜色了：

4 4 Seamless tiling

当域 Ω 为矩形时，可以通过使用泊松求解器强制执行周期性边界条件来使其内容可平铺。源图像 g 是原始图像，边界条件从 g 的边界值导出，使得矩形域的相对侧对应于相同的 Dirichlet 条件。比如对于上下衔接的图像，设置 $f^*_{up} = f^*_{down} = 0.5(g_{up}+g_{down})$，对于左右衔接的图像，设置 $f^*_{left} = f^*_{right} = 0.5(g_{left}+g_{right})$ 即可：



五　小结

总的来说是个挺容易理解的技术，难点在于约束问题的求解，当找到合适的求解方案时，问题就可以解决了。以前早期的计算机视觉算法很多都依赖于建模和约束求解，主要拼的是数学思想和方法，因此具有很高的参考价值，也很有趣味性。现在很多视觉的东西都交给了神经网络和大数据，导致人们失去了对基本方法的思考和探索，这也是不太好的地方。

借助图像梯度的泊松求解器的实现并不困难，我们会在本专栏的后续其他文章中介绍实现过程。

参考文献

[1] Pérez, Patrick, Michel Gangnet, and Andrew Blake. "Poisson image editing." ACM SIGGRAPH 2003 Papers. 2003. 313-318.

[2] https://zhuanlan.zhihu.com/p/355055346

[3] https://cloud.tencent.com/developer/article/2066941