

Intelligent Scissor for Image Composition

Author: Eric N. Mortensen William A. Barret

Translator: 软件工程卓越班 练孙鸿

0 Abstract

我们提出一种新的，交互式的工具“智能剪刀”(Intelligent Scissors)，用于图像分割与合成。全自动的图像分割还是一个未解决的问题(1995)，人工分割也很累人的。但是呢，智能剪刀就可以让数字图片里面的物体可以被快速提取出来，只需要用鼠标做一点简单的手势(gesture)。这些人工指定的鼠标位置(gestured mouse position)大致地描绘了物体边缘，然后 live-wire 边缘会“捕捉”(“snap” to)到这些位置，然后就可以包裹住目标物体了。

Live-wire 边缘检测把离散的动态规划(DP)当作二维图(graph)搜索问题来解决。DP 在提供一种数学上最优的边缘的同时还能极大地减少对局部噪音或干扰元素的敏感度。鲁棒性(Robustness)还会被进一步提高，这是通过不停机训练(on-the-fly training)使得 boundary 附着(adhere)在特定类型的 edge 而非最强的 edge 来达到的。**Boundary Cooling** 自动冻结那些不再变化的线段，然后自动输入额外的种子点(???automate input of additional seed points)。Cooling 使得用户可以更加自由地指定路径。

被提取的物体可以用 live-wire mask 和 **空间频率等同性(spatial frequency equivalencing)** 被缩放、旋转、组合。Frequency Equivalencing 可以用 Butterworth 滤波来实现，其中 Butterworth 滤波器可让最低频频谱(lowest frequency spectra)与其他图片元素匹配。智能剪刀还可以创造快速、有效的图片组合。

1 Introduction

数字图像合成在电影和桌面应用里很关键...blahblahblah。那么我们提出一个快速的、交互式的(interactive)物体提取工具 Intelligent Scissors。

然后还是在吹逼。

所以这篇论文描述了“智能剪刀”，用 DP 来建模问题。然后用 Boundary Cooling 和 on-the-fly training 来体现。最后我们提出 live-wire masking 和 spatial frequency equivalencing 来做 image composition。

2 Background

数字图像分割很关键，但是现在的方法(1995)都不太方便，就比人工跟踪好一点点。

基于区域(region-based)的魔术棒在很多应用里面都有，它就是要交互式地选择种子点来生长区域(添加相邻的像素)。因为这种区域生长法不提供交互式的视觉反馈(?? 果然是老论文，21 世纪 photoshop 的魔术棒都舒服得不行吧)

其他流行的边界定义方法(boundary definition method)用了动态的轮廓或者“蛇行”(snake)来改进手工输入的粗糙近似，“蛇”会并行地迭代调整边界来**最小化能量泛函(?) (energy functional)**从而得到一个最优的轮廓。能量泛函是内部力量如边界曲率(boundary curvature)，与外部力量如图像梯度模长(image

gradient magnitude)的结合。蛇行(snake)还可以跟踪连续帧之间不剧烈的边界移动。但是, blahblah 不知道在说啥, 反正 snake 的结果不一定令人满意, 如果不满意的话手动调整会很麻烦, 不断重复地手工修改。所以我们在 3.6 提出 intelligent scissor。

其他类型的图像分割技术用了图搜索(用动态规划来搞)来找到全局最优的边界。这些技术不同于蛇行, 因为边界点都是用 Stage-wise 最优开销的方式生成的, 而且蛇行可以并行地、迭代地最小化能量泛函。但是跟 snake 一样, 这些图搜索技术也是需要有一个边界模板的一例如手工输入的近似边界(manual entered rough approximation)、品质因数(a figure of merit)等东西。这些东西都是用来实现定向采样(directional sampling)和搜索限制(searching constraint)。所以图搜索方法也是 non-interactive 的。

所以智能剪刀和上面的方法的**根本区别不是边界定义方法, 而是互动的方法(method of interactive)**。智能剪刀让用户可以在从某个种子点出发、生成的所有最优边界中 INTERACTIVE SELECT 最适合的边界。而且, 之前的方法是不支持 on-the-fly training 和 cooling 的, 所以计算量就没智能剪刀这么少。(最后还发现, 基于空间频率的图像合成这个问题居然还没有被提到过)

3 Intelligent Scissors

用动态规划来进行边界定义可以看作是一个动态规划(DP)问题, 问题的目标是找到一条出发点的一组目标点的最优路径。具体到图像的边界搜索, 图搜索就是要找到由像素到像素的全局最优路径了。在这篇论文里面, 最优性(optimality)就被定义成从一个起始像素到目标像素的**最小的累计代价路径(minimum cumulative cost path)**, 而这里的累积代价(cost)指的是路径局部边(local edge)的代价的和。

3.1 Local Cost

因为最小代价路径要对应着图像元素的**【边缘】**, 所以拥有强边缘特征的 edge 就要有低 cost。local cost 的计算分成了三个部分(1995 这篇是这样, 1998 那篇长的就有 6 个部分)。Edge cost 的计算由下面三个指标加权求和得到

1. 拉普拉斯过零点(Laplacian Zero Crossing), 记为 f_z
2. 图像梯度的模长(Gradient Magnitude), 记为 f_G
3. 图像梯度的方向(Gradient Direction), 记为 f_D

设 $l(p, q)$ 是像素 p 到像素 q 的 local cost (边的权重), 则我们可以**给上面几个指标加权求和一波**, 即

$$l(p, q) = w_z f_z(q) + w_D f_D(p, q) + w_G f_G(q)$$

原文给出的三个权重 w_z, w_D, w_G 的经验值(Empirical value)分别为 0.43, 0.14, 0.43。

3.1.1 Laplacian Zero Crossing

这是在一副图像的经过 Laplacian 算子操作之后（实际当中一般用 Difference of Gaussian 来模拟，实现的时候可以用滑动窗口/模板，但是这次实验用的卷积核只是 3x3，可能就对噪声比较敏感了），寻找他的过零点 (Zero-Crossing)。那么如果相邻像素 p, q 的 Laplacian 符号不同的话，过零点就会在中间。而过零点意味着这可能是个边缘的 Good Indicator，那么应该给它少一点的 cost。所以

$$f_z(q) = \begin{cases} 0, & q \text{ 是过零点} \\ 1, & q \text{ 不是过零点} \end{cases}$$

3.1.2 Gradient Magnitude

拉普拉斯过零点被定义为了一个 binary operator，只判断两种情况，返回两种值，这样子它就不对边缘的强弱进行区分了。所以为了区分强/弱边缘，还是得搞一下梯度吧。所以设像素 q 处的梯度为 I_x, I_y ，则梯度模长

$$G = \sqrt{I_x^2 + I_y^2}$$

则：

$$f_G = 1 - \frac{G}{\max(G)}$$

3.1.3 Gradient Direction

梯度方向 indicator 的想法大概是，好边缘的走向一般都和梯度方向垂直，所以这里构造了一个函数 f_D ，使得路径 p→q 与梯度平时 cost 最大，与梯度垂直时 cost 最小。因为打起来比较麻烦，这里就不贴公式了。（然而我在写的时候这个 indicator 有点干扰，所以就直接设为常数了 2333333）

3.2 2D Dynamic Programming

没什么好说的，就是从种子点 (Seed Point) 到自由点 (Free Point) 的 Dijkstra 寻路。原文给出了伪代码。每个像素就是一个 node，每个 node 和相邻（纵横+斜）8 个 node 都有连接边，而连接边的 cost 就按 3.1 的来求就好了。这里的 Dijkstra 单源正权重寻路的 source 就是指 Seed Point。

3.3 其他

可以用鼠标点击指定第一个 Seed Point。后面每隔一段距离 Free Path 就会“冷却” (Cooling) 下来，于是新的 Seed Point 就产生在冷却路径的末尾。Photoshop 里面这个冷却距离是可以调整的，可以很密也可以很疏。我在这里实现就 free path 达到 100 个 pixel 就自动冷却，当然也可以按下 Space 来手动冷却。

4 实现

平台：Visual Studio 2015 + Win 8.1

框架：Noise3D（自研的图形渲染框架）

<https://github.com/CHINA-JIGE/Noise3D-DirectX11>

语言：C++

注：第一次在图片上单击左键可以设置第一个 seed point，移动鼠标来进行路径搜索。路径会自动冷却（标为蓝色）。可以按 Space 键手动冷却。再按一次鼠标左键可以停止这一轮的 intelligent scissor。

未来的优化：

1. 因为路径可以每到一定距离就 cool down，所以每次 cool down 的时候重新计算 dijkstra 寻路图的时候就可以限制寻路的范围