

用于内容感知图像大小调整的 Seam Carving

谢阿维丹
三菱电机研究实验室

阿里尔·沙米尔
跨学科中心和 MERL

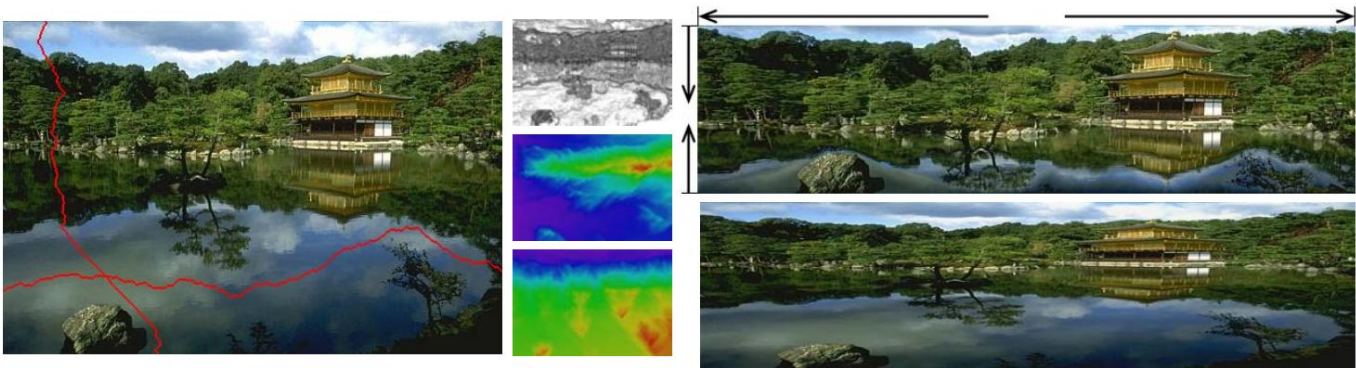


图 1: 接缝是图像中低能量像素的连接路径。左边是一横一纵接缝的原始图像。中间显示了本例中使用的能量函数（梯度的大小），以及用于计算接缝的垂直和水平路径图。通过自动切割接缝以减小图像大小，并插入接缝以扩展图像大小，我们实现了内容感知调整大小。与右下角的标准缩放相比，右上角的示例显示了我们在一个维度上扩展并在另一个维度上减少的结果。

抽象的

有效调整图像大小不仅应使用几何约束，还应考虑图像内容。我们提出了一个简单的图像操作符，称为 seam carving，它支持内容感知图像大小调整以进行缩小和扩展。接缝是单个图像上从上到下或从左到右的最佳 8 连接像素路径，其中最佳性由图像能量函数定义。通过在一个方向上重复切割或插入接缝，我们可以改变图像的纵横比。通过在两个方向上应用这些运算符，我们可以将图像重新定位到新的尺寸。

接缝的选择和顺序保护图像的内容，如能量函数所定义的那样。缝隙雕刻也可用于图像内容增强和对象去除。我们支持用于定义图像能量的各种视觉显着性度量，并且还可以包括用户输入来指导该过程。通过在图像中存储接缝顺序，我们创建了多尺寸图像，这些图像能够实时不断变化以适应给定尺寸。

CR 类别：I.3.0 [计算方法]:计算机图形学 一般； I.4.10 [计算方法]:图像处理 和计算机视觉 图像表示

关键词:图像大小调整,图像重定向,图像接缝,内容感知图像处理,显示设备

1 简介

当今显示设备的多样性和多功能性对数字媒体提出了新的要求。例如，设计人员必须为 Web 内容创建不同的替代方案，并为不同的设备设计不同的布局。此外，HTML 以及其他标准可以支持页面布局和文本的动态变化。然而，到目前为止，图像虽然是数字媒体中的关键元素之一，但通常在尺寸上保持刚性并且不能自动变形以适应不同的布局。在其他情况下，图像的尺寸或纵横比必须改变，以适应不同的显示器，例如手机或 PDA，或者打印在给定的纸张尺寸或分辨率上。

标准图像缩放是不够的，因为它忽略了图像内容并且通常只能统一应用。裁剪是有限的，因为它只能从图像周边移除像素。更有效的调整大小只能通过考虑图像内容而不仅仅是几何约束来实现。

我们提出了一个简单的图像运算符，我们称之为接缝雕刻，它可以通过优雅地雕刻或在图像的不同部分插入像素来改变图像的大小。Seam Carving 使用定义像素重要性的能量函数。接缝是低能量像素从上到下或从左到右穿过图像的连接路径。通过连续移除或插入接缝，我们可以在两个方向上缩小和放大图像的尺寸（见图 1）。对于图像缩减，接缝选择确保在保留图像结构的同时，我们删除更多的低能量像素和更少的高能量像素。

对于图像放大，接缝插入的顺序确保了原始图像内容和人工插入像素之间的平衡。实际上，这些运算符会产生内容感知的图像大小调整。

我们说明了缝隙雕刻和插入在纵横比变化、图像重定向、图像内容增强和对象移除方面的应用。此外，通过存储接缝移除和插入操作的顺序，并小心地将接缝交织在

我们定义了多尺寸图像的垂直和水平方向。这样的图像可以以内容感知的方式不断改变它们的大小。设计人员一次可以制作多尺寸图像,而客户端应用程序可以根据所需尺寸实时调整图像大小以适合确切的布局或显示。

Seam Carving 可以支持多种类型的能量函数,例如梯度大小、熵、视觉显着性、眼睛注视运动等。删除或插入过程是无参数的;然而,为了允许交互式控制,我们还提供了一个基于涂鸦的用户界面,用于为图像的能量添加权重并引导所需的结果。此工具还可用于创作多尺寸图像。总而言之,我们的主要贡献如下:

- 定义seam carving 并展示其属性。
- 目前使用接缝插入的图像放大算法。
- 使用接缝进行内容感知图像大小操作。
- 为连续图像重定向定义多尺寸图像。

2 背景

图像大小调整是许多图像处理应用程序中的标准工具。它通过将图像统一调整为目标大小来工作。最近,人们对图像重定向越来越感兴趣,它试图在保持重要特征不变的同时改变图像的大小,这些特征可以自上而下或自下而上检测。自上而下的方法使用面部检测器 [Viola 和 Jones 2001] 等工具来检测图像中的重要区域,而自下而上的方法依赖于视觉显着性方法 [Itti 等人。1999]构建图像的视觉显着图。一旦构建了显着图,就可以使用裁剪来显示图像中最重要的区域。苏等人。[2003] 提议基于显着图或人脸检测器的输出自动创建缩略图。然后裁剪大图像以捕获图像中最显着的区域。同样,Chen 等人。[2003] 考虑了使图像适应移动设备的问题。在他们的方法中,图像中最重要的区域被自动检测并传输到移动设备。刘等人。[2003] 还解决了图像重定向到移动设备的问题,建议以时间换取空间。给定一组感兴趣的区域,他们会构建一条穿过这些区域的最佳路径,并将它们一个接一个地依次显示给用户。桑泰拉等人。[2006] 除了构图规则之外,还使用眼动追踪来智能地裁剪图像。所有这些方法都取得了令人印象深刻的结果,但依赖于传统的图像大小调整和裁剪操作来实际改变图像的大小。

雅各布等人。[2003] 考虑一个自适应的基于网格的文档布局系统,该系统在内容和模板之间保持清晰的分离。页面设计者构建几个可能的模板,当显示内容时,使用最合适的模板。如果提供了模板,则模板可以使用图像的不同离散替代方案,但没有具体提及图像大小调整。

图像大小调整和图像裁剪之间的折衷是引入非线性的、数据相关的缩放。这种方法是由 Liu 和 Gleicher [2005; 2006] 用于图像和视频重定向。对于图像重定向,他们找到感兴趣区域 (ROI) 并构建一个新颖的鱼眼视图扭曲,该扭曲实质上在图像的每个维度上应用分段线性缩放函数。这样,在图像的其余部分被扭曲时,ROI 得以保持。一旦找到 ROI,就可以以交互速率进行重定向,因此用户可以控制所需的图像大小。

通过移动滑块,在他们的视频重定向工作中,他们使用图像和显着图的组合来找到 ROI。然后,他们结合使用裁剪、虚拟平移和镜头剪辑来重新定位视频帧。

Setlur 等人。[2005] 提出了一种自动的、非逼真的算法,用于将大图像重定向到小尺寸显示器。这是通过将图像分解为背景层和前景对象来完成的。重新定位算法将图像分割成多个区域,识别重要区域,删除它们,填充产生的间隙,调整剩余图像的大小,然后重新插入重要区域。

Gal 等人最近提出了在保留用户指定特征的同时将图像变形为任意形状的一般问题的第一个解决方案。[2006]。特征感知变形是通过拉普拉斯编辑技术的特定公式实现的,适用于适应域部分的相似性约束。由于局部约束是通过全局优化过程传播的,因此并非所有约束都可以同时满足。我们的算法是离散的,因此雕刻单个接缝不会影响图像的其余部分。

使用接缝进行图像编辑很普遍。阿加瓦拉等。[2004] 描述了一种交互式数字蒙太奇系统,该系统使用最少的用户帮助找到完美的接缝,将一组照片的部分组合成一张合成图片。贾等。[2006] 提出了扩展泊松图像编辑技术的拖放粘贴 [Perez 等人。2003] 计算源图像和目标图像之间的最佳边界 (即接缝)。罗瑟等。[2006] 开发了 AutoCollage,这是一种从图像集合中自动创建拼贴图像的程序。这个过程需要在许多图像片段之间找到最佳边界或接缝。以上方法都没有讨论图像重定向的问题。一个值得注意的例外是 Wang 和 Cohen [2006] 的工作,他们提出同时解决抠图和合成问题。它们允许用户缩放前景对象的大小并将其粘贴回原始背景。Zomet 等人。[2005] 评估了无缝图像拼接的几个成本函数,并得出结论,最小化拼接图像的梯度和输入图像的梯度之间的L1误差范数通常表现良好。可以通过多种方式计算接缝,包括 Dijkstra 的最短路径算法 [1998]、动态规划 [2001] 或图形切割 [2001]。

改变图像的大小在纹理合成领域得到了广泛的研究,目标是从小图像生成大纹理图像。Efros 等人。[2001] 找到最小化由两个重叠纹理块定义的误差表面的接缝。这样,原始的小纹理图像被接缝以形成更大的纹理图像。这后来被 Kwatra 等人扩展到处理图像和视频纹理合成。[2003] 展示了如何增加原始纹理视频的空间和时间维度。

至于物体移除,Bertalmio 等人。[2000] 提出了一种图像修复方法,可以从边界向内平滑地传播信息,模拟专业修复者使用的技术。基于补丁的方法 [Drori 等人。2003年;克里米尼西等人。2003年;贝塔尔米奥等。2003] 使用自动指导来确定矿山合成顺序,这大大提高了结果的质量。最近,Sun 等人。[2005] 提出了一种交互式方法,通过沿使用指定的曲线传播结构,在缺少强烈视觉结构的情况下处理修复。

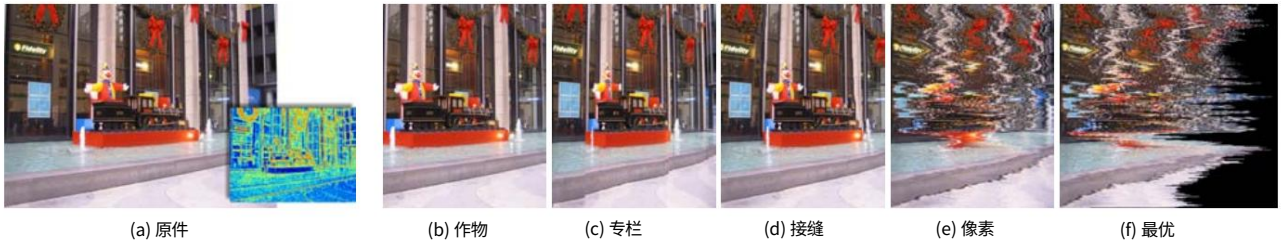


图 2:减少图像宽度的 5 种不同策略的结果。(a) 原始图像及其e1能量函数,(b) 最佳裁剪,(c) 移除能量最小的列,(d) 移除接缝,(e) 移除每行中能量最少的像素,最后,(f) 全局移除能量最低的像素,不管它们的位置如何。图 3 显示了每种策略的能量保存。

3 操作员

我们的内容感知调整大小的方法是以明智的方式删除像素。因此,问题是如何选择要去除的像素点?直觉上,我们的目标是去除与周围环境融为一体的不引人注意的像素。这导致在本文的许多图中使用了以下简单的能量函数,例如图 1.6.5.8.11.12.13 (我们在 3.2 小节中探讨了其他能量函数) :

$$\partial e(l)=\frac{\partial}{\partial x}|x|+\frac{\partial}{\partial y}|y| \tag{1}$$

给定能量函数,假设我们需要减小图像宽度。人们可以想出几种策略来实现这一目标。例如,保存能量 (即保留具有高能量值的像素) 的最佳策略是按升序移除具有最低能量的像素。这会破坏图像的矩形形状,因为我们可能会从每一行中移除不同数量的像素 (参见图 2(f))。如果我们想防止图像损坏,我们可以从每一行中删除相同数量的低能量像素。这保留了图像的矩形形状,但通过创建之字形效果破坏了图像内容 (图 2(e))。

为了保持图像的形状和视觉连贯性,我们可以使用自动裁剪。也就是说,寻找包含最高能量的子窗口,即目标图像的大小 (图 2(b))。

介于移除像素和裁剪之间的另一种可能策略是移除具有最低能量的整列。尽管如此,伪影可能会出现在生成的图像中 (图 2(c))。因此,我们需要一个调整大小运算符,它比裁剪或列移除限制更少,但可以比单个像素移除更好地保留图像内容。这导致我们的接缝雕刻策略 (图 2 (d)) 和内部接缝的定义。

形式上,设为 $n \times m$ 图像,定义垂直接缝为:

$$X=\{s_x^n\}_{x=1}^n=\{(x(i),i)\}_{i=1}^n, \text{ st } \forall i, |x(i)-x(i-1)| \leq 1, \tag{2}$$

其中 x 是映射 $x: [1,...,n] \rightarrow [1,...,m]$ 。也就是说,垂直接缝是图像中从上到下的 8 连接像素路径,在图像的每一行中包含一个且仅包含一个像素 (参见图 1)。类似地,如果 y 是映射 $y: [1,...,m] \rightarrow [1,...,n]$,则水平接缝是:

$$Y=\{s_y^m\}_{y=1}^m=\{(j,y(j))\}_{j=1}^m, \text{ st } \forall j, |y(j)-y(j-1)| \leq 1. \tag{3}$$

因此接缝 s (例如垂直接缝 $\{s_i\}$) 路径的像素将是 $l_s=\{l(s_i)\}_{i=1}^n=\{l(x(i),i)\}_{i=1}^n$ 。请注意,类似于从图像中移除行或列,从图像中移除接缝的像素仅具有局部效果:图像的所有像素都向左 (或向上) 移动以补偿丢失的路径。

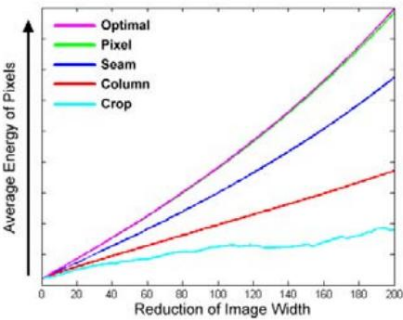


图 3:图像能量保存。使用五种不同的调整大小策略通过平均像素能量测量的内容保存比较。实际图像如图 2 所示。

视觉效果仅沿接缝路径可见,而图像的其余部分完好无损。另请注意,可以替换约束 $|x(i)-x(i-1)| \leq 1$ 为 $|x(i)-x(i-1)| \leq k$,并为 $k=0$ 获得一个简单的列 (或行),为任何值 $1 \leq k \leq m$ 获得分段连接或甚至完全不连接的像素集。

给定能量函数 e ,我们可以将接缝的成本定义为 $E(s)=E(l_s)=\sum_{i=1}^n e(l(s_i))$ 。我们寻找最佳接缝最小化这个接缝成本:

$$E(s)=\min_{s \in S} \sum_{i=1}^n e(l(s_i)) \tag{4}$$

可以使用动态规划找到最佳接缝。第一步是从第二行到最后一行遍历图像,并计算每个条目 (i,j) 的所有可能连接接缝的累积最小能量 M :

$$M(i,j)=e(i,j)+\min(M(i-1,j-1),M(i-1,j),M(i-1,j+1))$$

在此过程结束时, M 中最后一行的最小值将指示最小连接垂直缝的结束。

因此,在第二步中,我们从 M 上的这个最小条目回溯以找到最佳接缝的路径 (见图 1)。水平接缝的 M 的定义是类似的。

3.1 节能措施

为了评估内容感知调整大小的不同策略的有效性,我们可以检查所有像素的平均能量

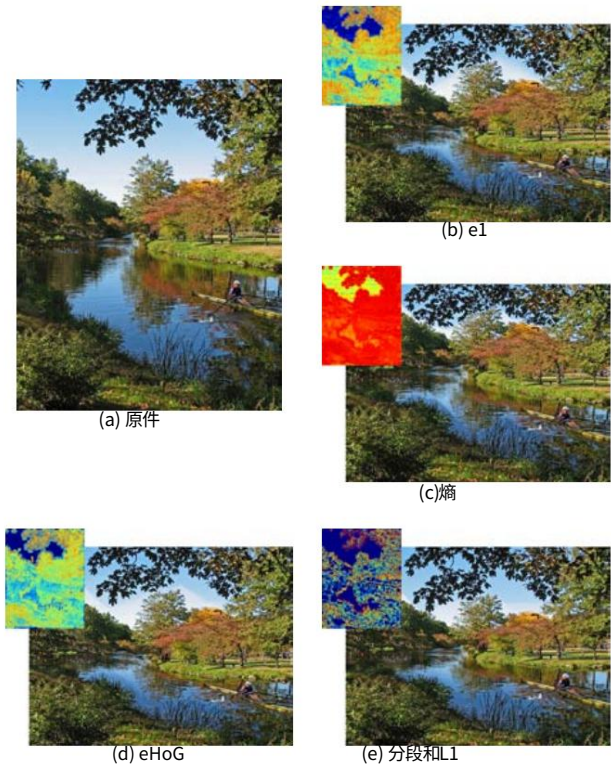


图 4:比较用于内容感知大小调整的不同能量函数。

在图像中 [1] $\sum p \in I e(p)$ 在调整大小时,随机移除像素应该保持平均值不变,但是内容感知调整大小应该提高平均值,因为它移除了低能量像素并保留了高能量像素。图 3 显示了使用上述五种不同策略改变图 2 的图像宽度时平均能量的变化图。正如预期的那样,按升序移除低能量像素会得到最佳结果。紧随其后的是像素移除,但是这两种方法都破坏了图像的视觉连贯性。裁剪显示出最差能量保存。列移除在保存能量方面做得更好,但仍然会引入视觉伪影。Seam Carving 在能量保存和视觉连贯性需求之间取得了最佳平衡。这个图表结果通常是许多图像的特征。

3.2 图像能量函数

我们已经检查了文献中发现的几种可能的图像重要性度量,作为指导接缝雕刻的能量函数。我们已经测试了梯度的L1和L2范数,显着性度量 [Itti 等人。1999] 和 Harris 角测量 [Harris 和 Stephens 1988]。我们还使用了眼睛注视测量 [DeCarlo 和 Santella 2002],以及面部检测器的输出。

图 4 比较了e1误差、熵、分割和梯度直方图 (HoG)的结果。熵能计算 9×9 窗口上的熵并将其添加到e1。分割方法首先分割图像[Christoudias et al. 2002] 然后对结果应用e1误差范数,有效地只留下线段之间的边缘。最后, eHoG定义如下:

我| ∂y $\frac{\partial}{\partial x} \text{我} | \frac{\partial}{\partial x}$
 $e_{\text{HoG}}(I) = \max(\text{HoG}(I(x, y)),$

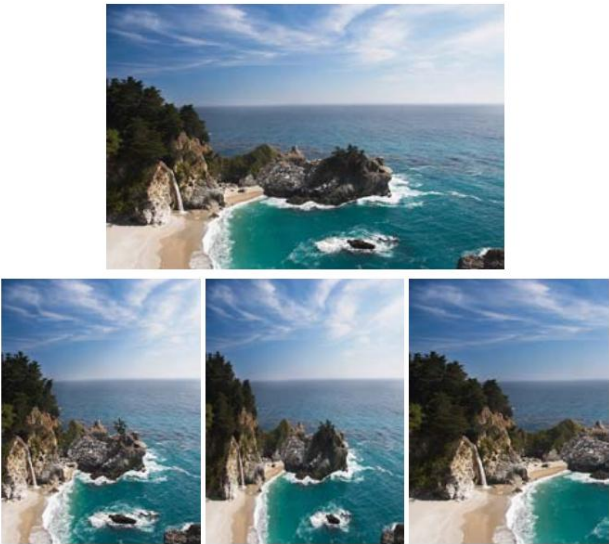


图 5:比较纵横比变化。底部从左到右:使用接缝去除、缩放和裁剪调整图像大小。

其中 $\text{HoG}(I(x, y))$ 被视为每个像素的定向梯度直方图 [Dalal 和 Triggs 2005]。我们使用在像素周围的 11×11 窗口上计算的 8-bin 直方图。因此,在分母处取 HoG 的最大值会将接缝吸引到图像中的边缘,而分子确保接缝将平行于边缘运行并且不会穿过它。图 9 和 10 中也使用了eHoG。

正如预期的那样,没有单一的能量函数在所有图像上都表现良好,但通常它们都适应相似的调整大小范围。它们在引入视觉伪影的速度以及它们影响的图像部分方面各不相同。我们发现e1或eHoG工作得很好。

4 离散图像大小调整

4.1 纵横比变化

假设我们想要将给定图像的纵横比从 $n \times m$ 更改为 $n \times m$,其中 $m - m = c$ 。这可以简单地通过从 I 中连续移除 c 个垂直接缝来实现。与简单缩放相反,此操作不会改变图像的重要部分 (由能量函数定义),并且实际上创建了一个不均匀的、内容感知的调整图像大小 (图 5)。

从 $n \times m$ 到 $n \times m$ 的相同纵横比校正也可以通过将行数增加 m/m 倍来实现 (图 6)。这种方法的附加值是它不会从图像中删除任何信息。我们在 4.3 小节中详细讨论了增加图像大小的策略。

4.2 使用最佳接缝顺序进行重定向

图像重定向将纵横比从一维变化推广到二维,这样大小为 $n \times m$ 的图像将被重定向到大小 $n \times m$,暂时,我们假设 $m < m$ 且 $n < n$ 。这就引出了一个问题,缝雕的正确顺序是什么?先去除垂直接缝?先水平接缝?还是两者交替?我们将搜索最优顺序定义为以下目标函数的优化:



图6:日本大师歌川广重的画面长宽比变化。在这两个示例中,原始图像都通过接缝插入加宽了。

$$\sum_{x=y, a, s=1}^k E(a_i s + (1-a_i) s_{n-1}) \quad (5)$$

其中 $k=r+c$, $r=(m-m)$, $c=(n-n)$ 并且 a_i 用作确定在第 i 步我们是否移除水平或垂直接缝的参数: $a_i \in \{0,1\}$, $\sum_{i=1}^k a_i = r$, $\sum_{i=1}^k (1-a_i) = c$

我们使用传输图 T 找到最佳顺序,对于每个所需的目标图像大小 $n \times m$,指定水平和垂直接缝去除操作的最佳序列的成本。也就是说,条目 $T(r, c)$ 包含获得大小为 $n-r \times m-c$ 的图像所需的最小成本。我们使用动态规划计算 T 。

从 $T(0,0) = 0$ 开始,我们填充每个条目 (r, c) ,选择两个选项中的最佳选项。从大小为 $n-r \times m-c+1$ 的图像中删除水平接缝或删除垂直接缝来自大小为 $n-r+1 \times m-c$ 的图像:

$$T(r, c) = \min(T(r-1, c) + E(s_x(n-r-1 \times m-c)), T(r, c-1) + E(s_y(n-r \times m-c-1))) \quad (6)$$

其中 $n-r \times m-c$ 表示大小为 $n-r \times m-c$ 的图像, $E(s_x(l))$ 和 $E(s_y(l))$ 是相应的接缝去除操作的成本。

我们存储一个简单的 $n \times m+1$ 位图,它指示在动态规划的每个步骤中选择了两个选项中的哪一个。

选择左邻居对应于垂直接缝去除,而选择顶部邻居对应于水平接缝去除。给定目标大小 $n \times m$,其中 $n=n-r$ 和 $m=m-c$,我们从 $T(r, c)$ 回溯到 $T(0,0)$ 并应用相应的删除操作。图 7 显示了图像上不同的重定向策略的示例。

4.3 图像放大

去除垂直和水平接缝的过程可以看作是一个时间演化过程。我们将 $l(t)$ 表示为较小的图像



图 7:最佳顺序重定向:顶部是原始图像及其传输图 T 。给定目标大小,我们遵循最佳路径 (T 上的白色路径)以获得重定向图像 (顶行,右)。为了进行比较,我们通过水平和垂直接缝去除之间交替 (顶行,左)、先去除垂直接缝 (底行,左)和先去除水平接缝 (底行,右)来显示重定向结果

在 t 接缝已从 l 中移除后创建。为了放大图像,我们近似于这次演化的“反转”,并在图像中插入新的“人工”接缝。因此,为了将图像 l 的大小放大一倍,我们计算 l 上的最佳垂直 (水平)接缝 s 并通过将 s 的像素与其左右邻居 (水平情况下的顶部和底部)平均来复制它们。

使用时间演化符号,我们将生成的图像表示为 $l(-1)$

不幸的是,通过选择相同的接缝,重复此过程很可能会产生拉伸伪像 (图 8(b))。为了实现有效放大,重要的是要在原始图像内容和人为插入的部分之间取得平衡。因此,为了将图像放大 k ,我们找到前 k 个要删除的接缝,并复制它们以达到 $(-k)$ (图 8(c))。这可以看作是从一个更大的图像中恢复像素的时间回溯的过程,这些像素会被 seam removal 移除 (尽管不能保证一定是这种情况)。

复制图像中的所有接缝相当于标准缩放 (见图 8(e))。为了继续以内容感知的方式过度放大图像 (例如,大于 50%),我们将过程分为几个步骤。每一步都不会将图像的大小放大到前一步的一小部分,从本质上防止重要内容被拉伸。然而,极端放大图像很可能会产生明显的伪像 (图 8(f))。

4.4 内容放大

seam carving 可以用来放大图像的内容,同时保持其大小,而不是扩大图像的大小。这可以通过结合缝隙雕刻和缩放来实现。为了尽可能保留图像内容,我们首先使用标准

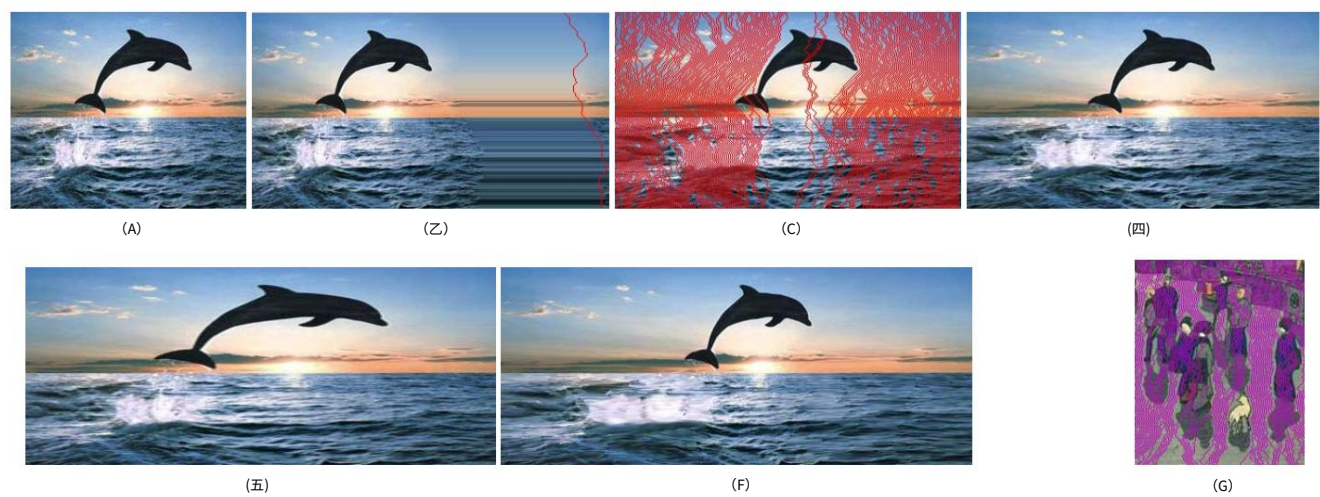


图 8:接缝插入:在放大的图像上找到并插入最佳接缝很可能会像 (b) 中那样一次又一次地插入相同的接缝。按移除顺序插入接缝 (c) 可实现所需的 50% 放大 (d)。(e) 在 (f) 中使用两步 50% 的接缝插入比缩放 (e) 获得更好的结果。在 (g) 中,显示了为展开图 6 而插入的接缝的近视图。



图 9:内容放大。右侧:缝隙雕刻和缩放的组合放大了原始图像的内容 (左侧)。

缩放以放大图像,然后才在较大的图像上应用接缝雕刻以将图像雕刻回其原始大小 (参见图 9)。请注意,移除的像素实际上是原始图像的子像素。



图 10:梯度域中的 Seam Carving。使用标准技术 (右上)和梯度域 (右下)重新定位原始图像 (左上)。放大比较显示在左下角。

4.5 梯度域中的Seam Carving

有时,从图像中移除多个接缝仍然会在调整大小后的图像中产生明显的视觉伪影。为了克服这个问题,我们可以将缝隙雕刻与泊松重建结合起来 ([Perez et al. 2003])。具体来说,我们像以前一样计算能量函数图像,但不是从原始图像中移除接缝,而是在梯度域中工作,并从原始图像的 x 和 y 导数中移除接缝。在此过程结束时,我们使用泊松求解器重建图像。图 10 显示了该技术的示例。

4.6 对象移除

我们使用一个简单的用户界面来移除对象。用户标记要移除的目标对象,然后从图像中移除接缝,直到所有标记的像素都消失。系统可以自动计算目标去除区域的垂直或水平直径中较小的一个 (以像素为单位),并相应地进行垂直或水平去除 (图11)。此外,要恢复图像的原始大小,可以在生成的 (较小的)图像上使用接缝插入 (参见图 12)。注意,相反



图 11:简单对象移除:用户在原始图像上标记一个要移除的区域 (绿色),并可能标记一个要保护的区域 (红色) (见左图中的插图)。在右图中,连续的垂直接缝被移除,直到没有“绿色”像素留下。



图 12:对象移除:找到丢失的鞋子! (原始图像在左上角)。在此示例中,除了移除对象(一只鞋)之外,图像还被放大回其原始大小。请注意,此示例很难使用绘画内画或纹理合成来完成。

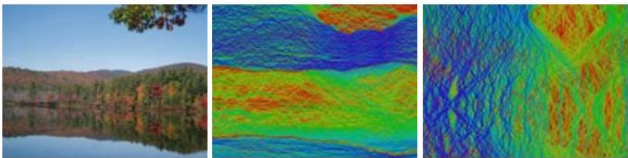


图 13:具有垂直和水平接缝索引V和H 的图像,按索引从蓝色(第一个接缝)到红色(最后一个接缝)着色。

到以前的对象移除技术 [Drori 等人。2003年;克里米尼西等人。2003年;贝塔米奥等。2003],该方案改变了整个图像(如果调整大小,则改变其大小或内容)。这是因为移除和插入的接缝都可能经过图像中的任何位置。

5 张多尺寸图像

到目前为止,我们假设用户提前知道目标大小,但在某些情况下这可能是不可可能的。例如,考虑嵌入在网页中的图像。网页设计师事先不知道页面将以什么分辨率显示,因此无法生成单个目标图像。在不同的场景中,用户可能想尝试不同的目标尺寸并选择最适合他或她需要的尺寸。

接缝雕刻与像素数量成线性关系,因此调整大小与要移除或插入的接缝数量成线性关系。平均而言,我们在大约 2.2 秒内将尺寸为 400×500 的图像重新定位为 100×100 。然而,实时计算数十或数百个接缝是一项具有挑战性的任务。为了解决这个问题,我们提出了一种多尺寸图像的表示形式,它为尺寸为 $(m \times n)$ 的图像编码从 1×1 到 $m \times n$ 的整个重定向尺寸范围,当 N 时甚至进一步到 $N \times M > n, M > m$ 。此信息的内存占用非常低,可以在几秒钟内作为预处理步骤计算出来,并允许用户实时连续重新获取图像。

从不同的角度来看,这可以看作是存储了接缝移除和插入的时间演化隐式过程的显式表示。首先考虑改变图像宽度的情况。我们定义一个大小为 $n \times m$ 的索引图V,它为每个像素编码移除它的接缝的索引,即 $V(i, j) = t$ 表示像素 (i, j) 被 t 移除-th seam re



图 14:单独使用e1 (右上)和人脸检测器 (右下)重新定位左图像。



图 15:重新定位佛陀。顶部是原始图像、装饰物消失的裁剪版本和内容被拉长的缩放版本。使用简单的自下而上特征检测进行自动重定向无法保护佛像的面部结构(左下),这对面部检测器来说也是一个具有挑战性的图像。通过添加简单的用户约束来保护面部(底部,中间)或面部和花朵(底部,右侧),可以获得更好的效果。

移动(图 13)。要获得宽度为 m 的图像,我们只需要在每一行中收集接缝索引大于或等于的所有像素至

此表示支持图像放大和缩小。例如,如果我们想要支持将图像放大到 $M > m$,我们使用接缝插入过程将图像放大到 $n \times M$,类似于第 4.3 节。然而,我们不是对第 k 个接缝及其两个邻居进行平均,而是不修改接缝中的原始图像像素,而是将新像素插入图像作为第 k 个接缝及其左侧(或右侧)的平均值像素邻居。插入的接缝被赋予从 -1 开始的负索引。因此,为了将原始图像放大 $k, (m < k \leq M)$,我们使用完全相同的过程收集(从放大的图像中)接缝索引大于 $(m - (m + k))$ 的所有像素 $= -k$,得到大小为 $m - (-k) = m + k$ 的图像。

计算用于图像高度放大和缩小的水平索引图H以类似的方式实现(参见图 13)。然而,在独立计算H和V的同时支持两个维度调整大小是不通的。这是因为水平和垂直接缝可以在多个地方发生碰撞,并且在一个方向上移除接缝可能会破坏另一个方向的索引图。

可以在附录中找到更多详细信息。然而,避免这种情况的一种简单方法是允许在一个方向上移除接缝,并在另一个方向上使用生成接缝,即行或列。请注意,虽然将多尺寸图像重定向到任何尺寸是即时的

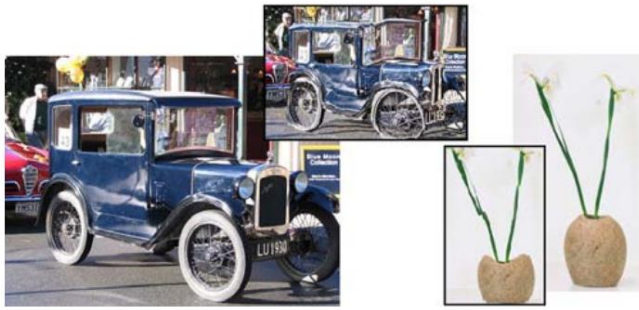


图 16:使用接缝调整大小时失败的示例:图像过于紧凑(左)或内容布局阻止接缝绕过重要部分(右)。在这种情况下,最好的策略是使用缩放。

重要的是,由于额外的约束,生成的图像将不同于 4.2 小节中隐式时间演化过程诱导的最佳顺序创建的图像。读者可以参考随附的视频,例如实时连续将多尺寸图像调整为各种尺寸。

6 限制

到目前为止,本文中显示的所有示例都是自动计算的,但我们的方法并不适用于所有图像。这可以通过添加更高级别的提示(手动或自动)来纠正。例如,在图 14 中, e_1 错误函数失败了,但结合人脸检测器,我们得到了更好的结果。

图 15 显示了添加用户约束的示例。

其他时候,即使是高级信息也无法解决问题。我们可以描述限制我们的 seam carving 方法的两个主要因素。第一个是图像中的内容量。如果图像太紧凑,从某种意义上说它不包含“不太重要”的区域,那么任何类型的内容感知调整策略都不会成功。第二种限制是图像内容的布局。在某些类型的图像中,虽然没有压缩,但内容的布局方式可以防止接缝绕过重要部分(图 16)。

7 结论和未来工作

我们提出了一个操作符,用于使用 seam carving 对图像进行内容感知调整大小。接缝被计算为单个图像上的最佳路径,并从图像中删除或插入。

该运算符可用于各种图像操作,包括:宽高比更改、图像重定向、内容放大和对象移除。操作员可以轻松地与各种显着性措施以及用户输入集成,以指导调整大小的过程。此外,我们为支持实时连续调整大小的多尺寸图像定义了一个数据结构。

这项工作有许多可能的扩展。我们希望将我们的方法扩展到其他领域,首先是调整视频大小。由于在某些情况下缩放可以实现更好的大小调整结果,我们想研究结合这两种方法的可能性,特别是定义更强大的多尺寸图像。我们还找到一种更好的方法来组合多尺寸图像中的水平和垂直接缝。

致谢

我们要感谢 Fredo Durand 和麻省理工学院的图形组审阅了这项工作的早期版本。我们感谢 Stark Draper 为视频解说。我们感谢 Eric Chan 使用瀑布图像,感谢众多 flickr (<http://www.flickr.com/>) 成员通过创意共同权利 (<http://creativecommons.org/>) 提供他们的图像: crazyegg95 (佛陀)、Gustty (情侣和冲浪者)、JeffKubina (国会大厦)、mykaul (汉努卡和汽车)、o2ma (花瓶)、sigs66 (长滩和海边的两人)。我们也感谢匿名审稿人和审稿人的意见。

参考

- AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D. 和 COHEN, M. 2004。交互式数字蒙太奇。ACM 跨。图形。23.3, 294–302。
- BERTALMIO, M., SAPIRO, G., CASELLES, V. 和 BALLESTER, C. 2000。图像修复。在 ACM SIGGRAPH 会议记录中, 417–424。
- BERTALMIO, M., VESE, L., SAPIRO, G. 和 OSHER, S. 2003。同步结构和纹理图像修复。在过程中。IEEE 计算机视觉和模式识别会议, 707–714。
- BOYKOV, Y., 和 JOLLY, M.-P. 2001。Interactive graph cuts for optimal boundary & region segmentation of objects in nd images。在国际计算机视觉会议 (ICCV) 中, 第一卷。我, 105–112。
- CHEN, L., XIE, X., FAN, X., MA, W., ZHANG, H., AND ZHOU, H. 2003。在小型显示器上调整图像的视觉注意模型。多媒体系统 9.4, 353–364。
- CHRISTOUDIAS, C., GEORGESCU, B., AND MEER, P. 2002。低水平视觉的协同作用。在第 16 届国际模式识别会议上, 卷。四, 150–155。
- CRIMINISI, A., PEREZ, P. 和 TOYAMA, K. 2003。通过基于示例的修复进行对象移除。在 IEEE 计算机视觉和模式识别会议中, 417–424。
- DALAL, N., AND TRIGGS, B. 2005。用于人体检测的定向梯度直方图。在计算机视觉与模式识别国际会议上, 卷。2, 886–893。
- DAVIS, J. 1998。带有移动物体的场景马赛克。在 CVPR 会议记录中。
- DECARLO, D., AND SANTELLA, A. 2002。照片的程式化和抽象化。在 SIGGRAPH 会议记录中, 769–776。
- DRORI, I., COHEN-OR, D. 和 YESHURUN, Y. 2003。基于片段的图像补全。在 ACM SIGGRAPH 会议记录中, 303–312。
- EFROS, AA, AND FREEMAN, WT 2001。用于纹理合成和传输的图像衍缝。在 SIGGRAPH 2001, Computer Graphics Proceedings, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., 341–346。
- GAL, R., SORKINE, O., AND COHEN-OR, D. 2006。特征感知纹理。在 Eurographics 渲染研讨会上。

HARRIS, C., AND STEPHENS, M. 1988. 组合角和边缘检测器。在第四届 Alvey Vision 会议记录中,147-151。

ITTI, L., KOCH, C., AND NEIBUR, E. 1999。用于快速场景分析的基于显着性的视觉注意模型。PAMI 20,11,1254-1259。

JACOBS, C., LI, W., SCHRIER, E., BARGERON, D和SALESIN, D. 2003。自适应基于网格的文档布局。在 ACM SIGGRAPH 会议记录中,838-847。

JIA, J., SUN, J., TANG, C.-K.和SHUM, H.-Y. 2006. 拖放粘贴。在 SIGGRAPH 的会议记录中。

KUHN, HW 1955。分配问题的匈牙利方法。在海军研究后勤季刊,2:83-97。

LIU, F.和GLEICHER, M. 2005。使用鱼眼视图变形进行自动图像重定向。在 ACM UIST,153-162。

LIU, F., AND GLEICHER, M. 2006。视频重新定位:自动平移和扫描。在 ACM 多媒体国际会议上,241-250。

LIU, H., XIE, X., MA, W., AND ZHANG, H. 2003。移动设备上大图片的自动浏览。第 11 届 ACM 多媒体国际会议论文集,148-155。

PEREZ, P., GANGNET, M., AND BLAKE, A. 2003。泊松图像编辑。ACM 跨。图形。22,3,313-318。

ROTHER, C., BORDEAUX, L., HAMADI, Y和BLAKE, A. 2006. 自动拼贴画。在 SIGGRAPH 2006 会议记录中。

SANTELLA, A., AGRAWALA, M., DECARLO, D., SALESIN, D.和COHEN, M. 2006。用于半自动照片裁剪的基于注视的交互。在 ACM 计算系统中的人为因素 (CHI),771-780。

SETLUR, V., TAKAGI, S., RASKAR, R., GLEICHER, M.和GOOCH, B. 2005。自动图像重定向。在移动和无处不在的多媒体 (MUM) 中,ACM 出版社。

SUH, B., LING, H., BEDERSON, BB和JACOBS, DW 2003. 自动缩略图裁剪及其有效性。在 UIST 03:第 16 届 ACM 用户界面软件和技术年度研讨会论文集,ACM 出版社,美国纽约州纽约市,95-104。

SUN, J., YUAN, L., JIA, J., AND SHUM, H. 2005。用结构传播完成图像。在 ACM SIG GRAPH 的会议记录中。

V. KWATRA, A. SCHDL, IEGT和BOBICK, A. 2003。Graphcut 纹理:使用图形切割的图像和视频合成。在 SIGGRAPH 的会议记录中。

VIOLA, P., AND JONES, M. 2001。使用简单特征的增强级联进行快速对象检测。在计算机视觉和模式识别 (CVPR) 会议上。

WANG, J.和COHEN, M. 2006。同步抠图和合成。Microsoft Research 技术报告, MSR-TR 2006-63 (5 月)。

ZOMET, A., LEVIN, A., PELEG, S.和WEISS, Y. 2005。通过最小化错误边缘进行无缝图像拼接。IEEE 图像处理汇刊 15, 4, 969-977。

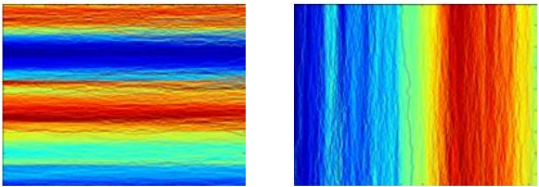


图 17:图 13 中图像的水平和垂直一致索引图。颜色索引从蓝色 (第一个接缝)变为红色 (最后一个接缝)。

构建一致的索引图

为多尺寸图像独立计算水平索引图H和垂直索引图V将不起作用。如果每个水平接缝都与所有垂直接缝索引相交 (或接触)并且每个垂直接缝都与所有水平接缝索引相交,则我们说H和V是一致的。一致性确保在任何维度上移除一个接缝都会从另一个维度上的所有接缝中恰好移除一个像素,同时保留索引图结构。如果不保持一致性,那么在去除一个水平接缝后,我们可能会留下像素数不同的垂直接缝,图像的矩形结构将被破坏。

除了将接缝限制为一个或两个维度中的行或列之外,我们在这里提出另一种方法。我们仅使用时间上 0 连接的接缝,即在原始尺寸图像I (0)上空间连接的接缝

对于这样的接缝,唯一可能违反H和V图之间一致性的可能发生在对角线接缝步骤中。我们的方法首先在一个方向上计算时间上 0 连接的接缝,然后在计算另一个方向的接缝时对对角线施加约束。

为了理解为什么唯一违反一致性发生在对角线上,不失一般性地假设某些垂直接缝 $j \in \{1...m\}$ 违反了一致性约束。这意味着它必须多次接触某个水平接缝 $i \in \{1...n\}$ 。将接缝 j 与接缝 i 相交的那些像素表示为 p 和 q。因为 p 和 q 是垂直接缝 j 的一部分,所以它们不能在同一行。然而,它们也是水平缝 i 的一部分,并且不能在同一列中。让我们检查一下由 p 和 q 定义的矩形的角。接缝 i 和 j 必须在这个矩形内连接,并且它们都接触到它的角。但是,一个是垂直接缝,另一个是水平接缝。发生这种情况的唯一可能性是矩形实际上是正方形,并且两条接缝都穿过它的对角线。

请注意,上述声明依赖于原始图像中所有接缝都连接的事实,如果我们使用非 0 连接接缝,则情况并非如此。然而,因为我们使用的是 0-connected 接缝,我们可以同时计算一个方向上的所有接缝。在不失一般性的情况下,对于 0 连接的垂直接缝,我们独立地检查原始图像的所有行对。对于每个这样的对,我们找到一组最优的单边路径,将一行的所有像素连接到下一行的所有像素。从图像顶部到底部的全局多条接缝路径只是这些单边路径的串联。

寻找一对行之间的最佳单边路径类似于加权分配问题,其中一行中的每个像素都连接到另一行中的三个相邻像素。我们使用匈牙利算法 [Kuhn 1955] 来解决这个加权分配问题。一旦我们在一个方向上找到了接缝,我们就会在另一个方向上重复这个过程,但是我们会屏蔽掉已经被任何第一个方向的接缝使用的每条对角线边缘。这保证了第二个方向的接缝将与第一个方向一致 (图 17)。