

Scene Completion Using Millions of Photographs

论文阅读报告

计 32 黄世宇

2013011304

【论文说明】

我选取的论文名叫 Scene Completion Using Millions of Photographs, 中文翻译为基于海量图片的场景补全。这篇论文主要做的一件事情就是对缺损的图像进行补全。图像补全的论文有很多, 这篇文章之所以能够脱颖而出, 是因为传统的图像补全都只是使用图片自身的一些来对未知的区域进行补全, 但是单张图片的信息是有限的, 有时候只利用图片自身显然是不能修复整张图片的信息的。所以, 作者通过大量外部图片数据库信息对图像进行补全, 取得了传统补全方法所不能达到的效果。

这篇论文发表于 2007 年的 SIGGRAPH, 作者是 James Hays 和他当时的导师 Alexei A. Efros。



James Hays



Alexei A. Efros

我上 James Hays 的个人主页看了一下, 发现这篇论文是他在卡内基梅隆大学读博士期间的一篇论文。这也是他个人主页上最早的一篇论文。读完博士的 James Hays 又到麻省理工学院继续读博士后, 读完博士后又到布朗大学当助理教授, 最后又回到自己读本科的学校——佐治亚理工学院当副教授。

我后来又去谷歌学术上搜索了一下 James Hays, 结果如下:



James Hays
Manning Assistant Professor of Computer Science, Brown University
Computer Graphics, Computer Vision, Computational Photography
在 cs.brown.edu 的电子邮件经过验证 - 首页

[关注](#)

Google 学术搜索

引用指数	总计	2011 年至今
引用	4141	3262
h 指数	22	21
i10 指数	26	25

看来，James Hays 的引用数和各项指标都是大牛级别的啊。既然 James Hays 是大牛，身为 James Hays 导师的 Alexei A. Efros 也可为是大牛中的大牛。Alexei A. Efros 曾在 CMU 任职，现在又在 UC Berkeley 任职。在谷歌学术上查 Alexei A. Efros，其学术影响力也是颇为惊人。



Alexei A. Efros
Associate Professor of Computer Science, UC Berkeley
computer vision, computer graphics, computational photography
在 eecs.berkeley.edu 的电子邮件经过验证 - 首页

关注

Google 学术搜索

引用指数	总计	2011 年至今
引用	16300	9950
h 指数	44	41
i10 指数	59	56

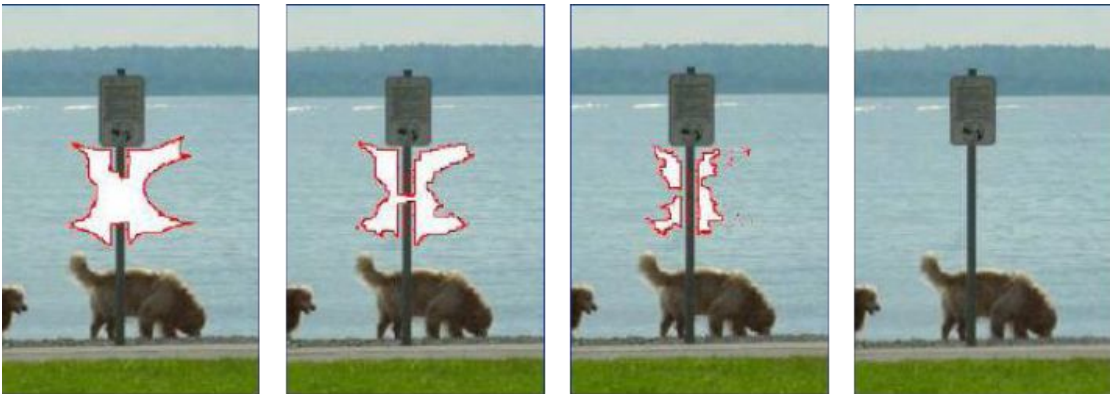
回到这篇论文，在谷歌学术上可以看出，该篇论文的被引用量达 524 次，可见其影响力还是非常大的。接下来我会细细讲解这篇论文的实现原理和流程。最后我会就论文中存在的几个问题提出一些改进意见。

【论文背景】

首先是要为什么要进行图像补全。因为现实生活中，可能因为各种各样的不可抗力导致图像局部被损坏，但是我们仍然希望借助计算机技术，根据现存图片的一些语句信息和上下文关系，尽量恢复出和原始图像相似且合理的图片。

然后是要为什么要是用海量图片进行图片补全。其实在这篇文章以前，就有很多关于图像补全的论文了。但是这些论文都有相同的一点，就是通过数学建模，在利用原图的信息进行图像补全。有人就会问，为什么可以用原图的信息就能补全缺失的信息呢。因为这是有局限的，这也是传统图像补全方法的局限。这个局限就是，待补全的图像需要具备比较明显的纹理特征，通过纹理的延生来进行图像补全。

一篇传统的图像补全的论文，如 Region Filling and Object Removal by Exemplar-Based Image Inpainting，就是通过已知纹理的延生，对未知区域进行补全。下面是这篇论文的一些例子：



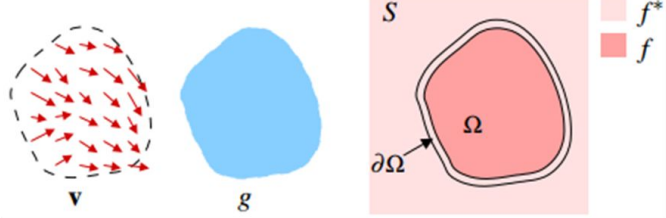


可以看出，这些图像都有很明显的纹理特征，而且补全后的图片就是原图片上的纹理衍生。所以这种方法有很大的局限性。所以这里需要解决两个问题：一、在很多情况下，通过单张原始图像对图片进行修复是不可能的，比如一个去掉屋顶的房屋，不可能通过原始图片补全屋顶。二、即使图中有合适的补全信息，但是会造成不合理的重复，比如，建筑补全，可能被图中其他建筑用来补全，导致建筑重复，补全效果差。所以，原图的信息不足以补全图像，这时，很自然的，我们会想到用其他更多的图片信息，来弥补原图信息的不足。于是，这种思想导致了这篇论文的出现。

【预备知识】

1. 泊松方程

假设目标图像是 S ，源图像是区域 Ω ，它们都是 \mathbb{R}^2 中的闭区域。假设区域 Ω 在 S 中的边界为 $\partial\Omega$ ，如下图所示。



假设 f^* 是 S 的像素函数减去原区域 Ω 内部像素函数的结果， f 为 Ω 区域内目前未知的像素方程。 \vec{v} 为 Ω 上的向量场。

要做到融合的效果，就需要区域内外在边界处梯度一致，以保证不存在跳变的情况。则这个问题的解，就是下面这个最优化问题的解

$$\begin{cases} \min_f \iint_{\Omega} |\nabla f - \vec{v}|^2 \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases} \quad (1)$$

其中 ∇ 是梯度算子。这个问题的最优解泊松方程在狄利克雷边界条件下的解

$$\begin{cases} \Delta f = \text{div } \vec{v} \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases} \quad (2)$$

其中 $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ 是拉普拉斯算子， $\text{div } \vec{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ 是 $\vec{v} = (u, v)$ 的散度。

在离散的情况下，假设 N_p 是像素 p 的四连通相邻像素， $\langle p, q \rangle$ 是两个相邻像素对，令 f 在 p 处的函数值为 f_p ，则 (1) (2) 式可改写成

$$\begin{cases} \min_{f|_{\Omega}} \sum_{\langle p, q \rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2 \\ f_p = f_p^* \quad p \in \partial\Omega \end{cases} \quad (3)$$

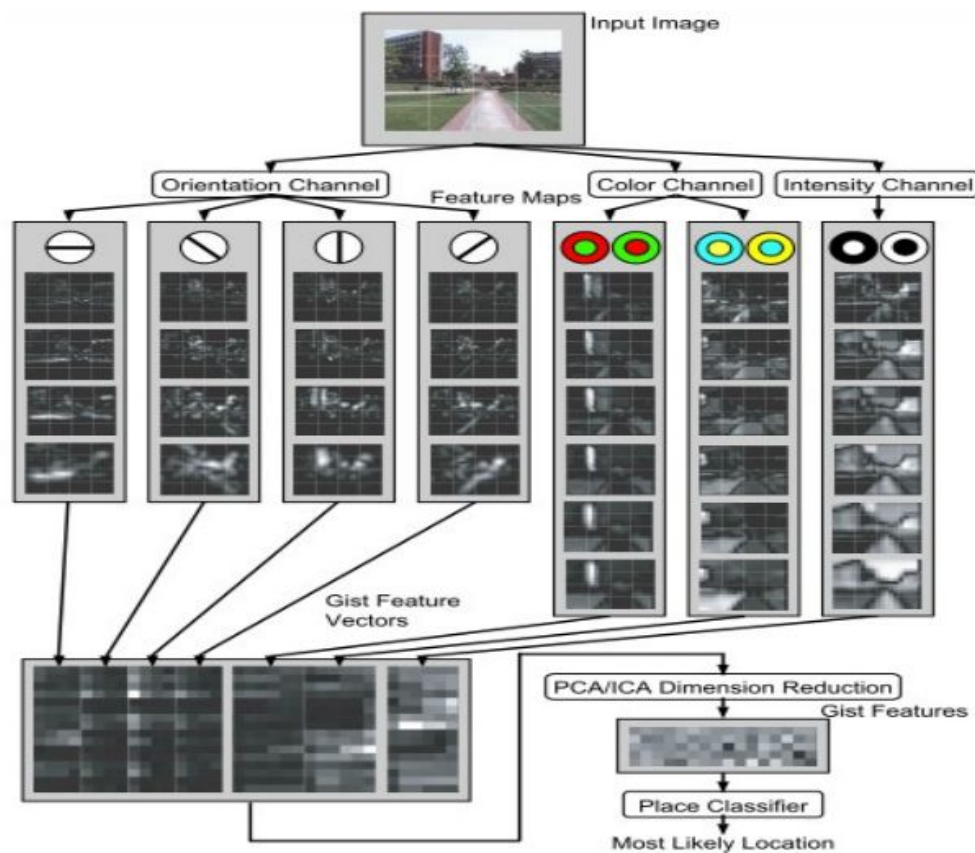
其中 $v_{pq} = \vec{v} \left(\frac{p+q}{2} \right) \cdot \overrightarrow{pq}$ ，可以理解为原像素函数之差。而 (3) 式中问题满足下面的线性方程组

$$|N_p|f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq} \quad \forall p \in \Omega$$

这个方程组直接求解将是 $O(N)$ 左右的复杂度。其系数矩阵是一个稀疏矩阵，每行非零系数最多为5个，又是对角优势矩阵，因此可以通过高斯-赛德尔迭代法来快速求得其近似解。

2. G i s t 特征

采取多尺度的滤波器对图像进行锐化预处理，再利用 G a b o r 变换来提取图像的 G i s t 特征。如下图所示：

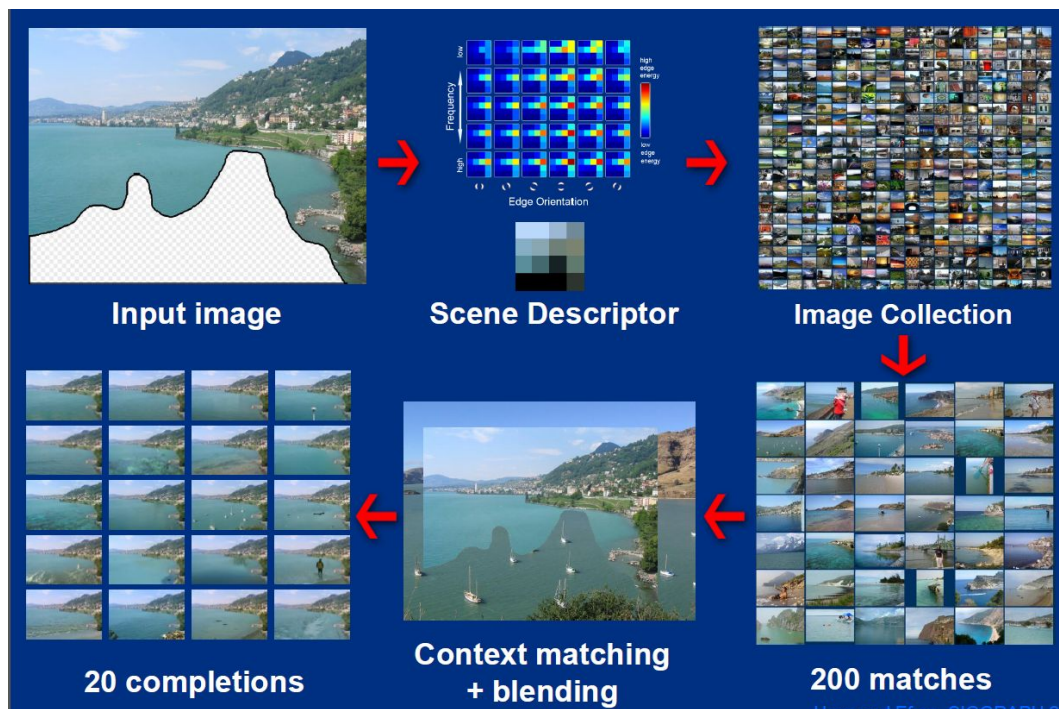


【数据处理】

从 flickr 上下载的高清图片以及用关键字搜索出的图片。去除重复图像，最大尺寸小于 800，最小尺寸小于 500 的图片去掉，所有图像进行降采样。图像下载，处理和场景匹配在 15 个机器上完成，一共得到 2.3 百万个不同的图像，一共 396G。然后把这些图片按照 gist scene detector 进行分类。

【算法流程】

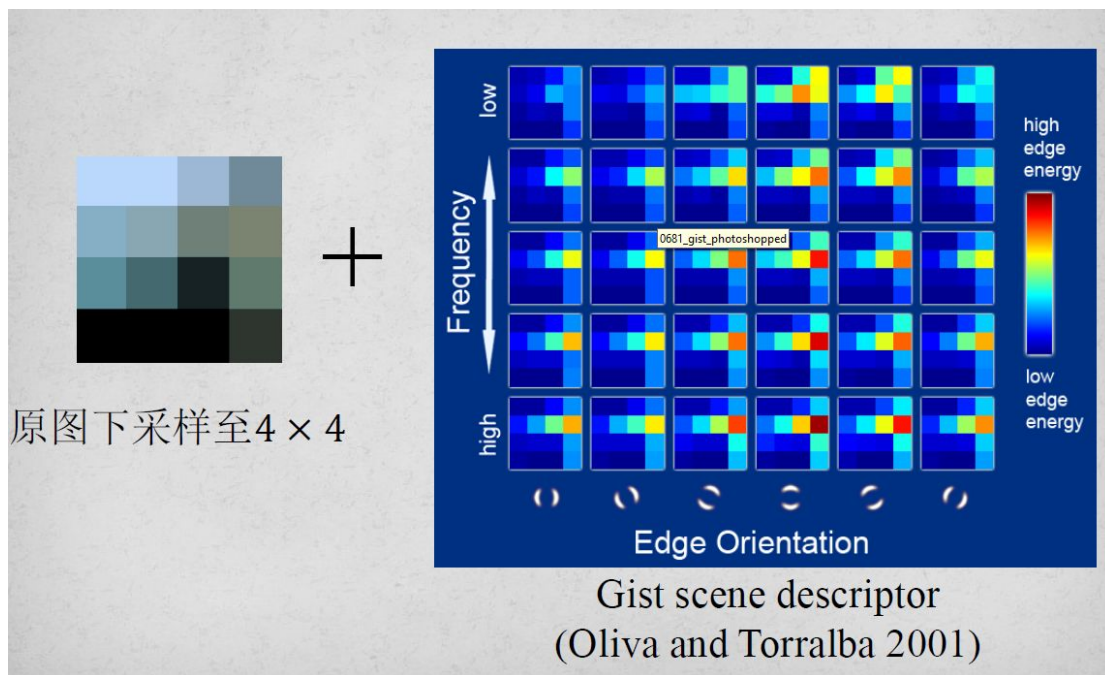
算法总体流程如下图：



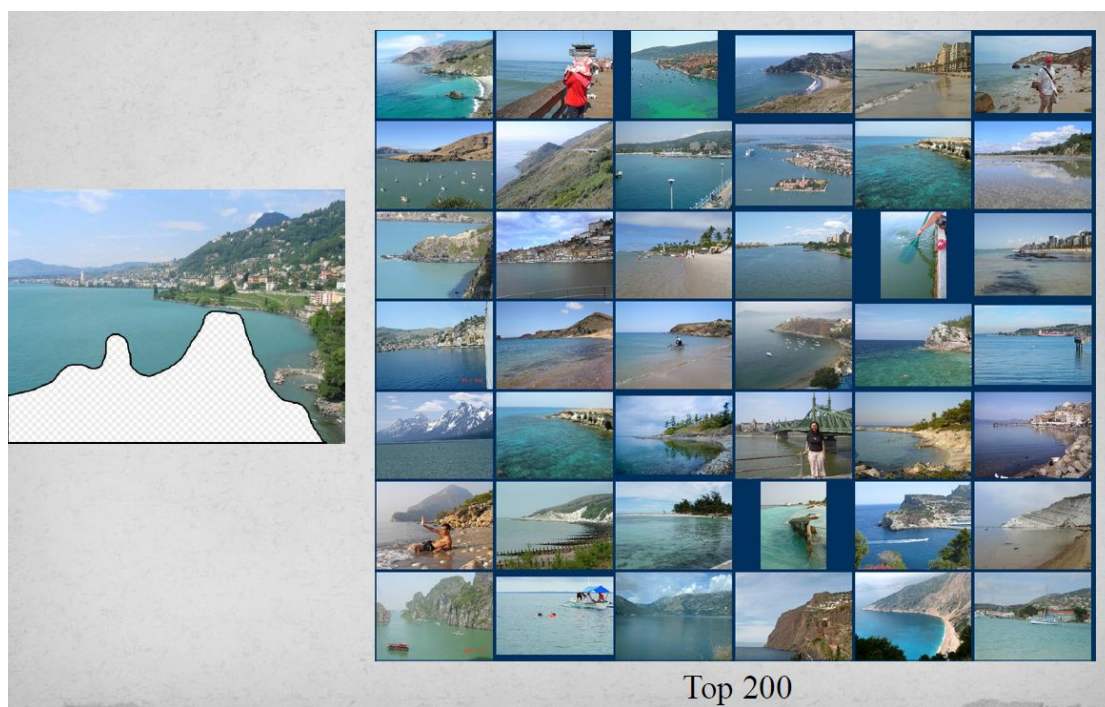
算法流程文字描述：

1. 从互联网中下载海量的图片，主要内容是室外场景，包括风光名胜、建筑、街道等，存入数据库。
2. 为数据库中的每一张图片生成一个描述符。
3. 输入待补全图片和需要剔除的区域，为待补全图片生成描述符。
4. 在数据库中寻找与待补全图片描述符最相似的图片 200 张。
5. 在这 200 张图片中，为每一张图片寻找最佳的合成位置。
6. 计算拼合边缘，完成图片拼接，返回最优的 20 个结果供用户选取。

所谓的描述符就是前文提到过的 **gist** 特征加上原图的下采样，具体如下图所示。



然后通过在数据库中进行搜索，找出描述符最相近的 200 个待选图片。



基于颜色

基于梯度

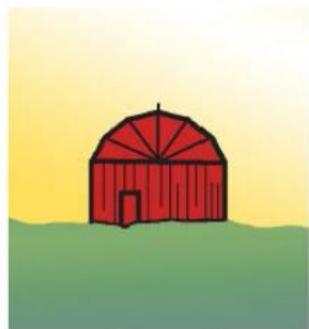


Image 1



(a)



(b)

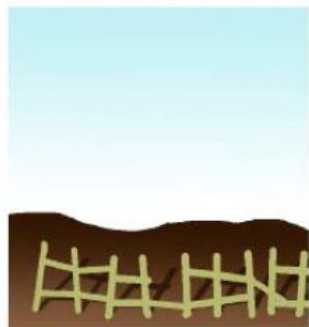


Image 2



(c)





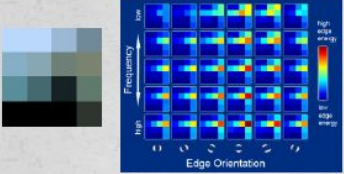
(d)

Graph-Cut 算法的两种实现

对于其中一个待选图片，通过基于梯度的 Graph-Cut 算法，寻找原图在待选图片中的最佳位置，如下图所示：



通过泊松融合使得原图和待选图片进行融合。然后通过以下公式计算出每张待选图片的代价。选出代价最小的 20 个作为返回结果，供用户返回。



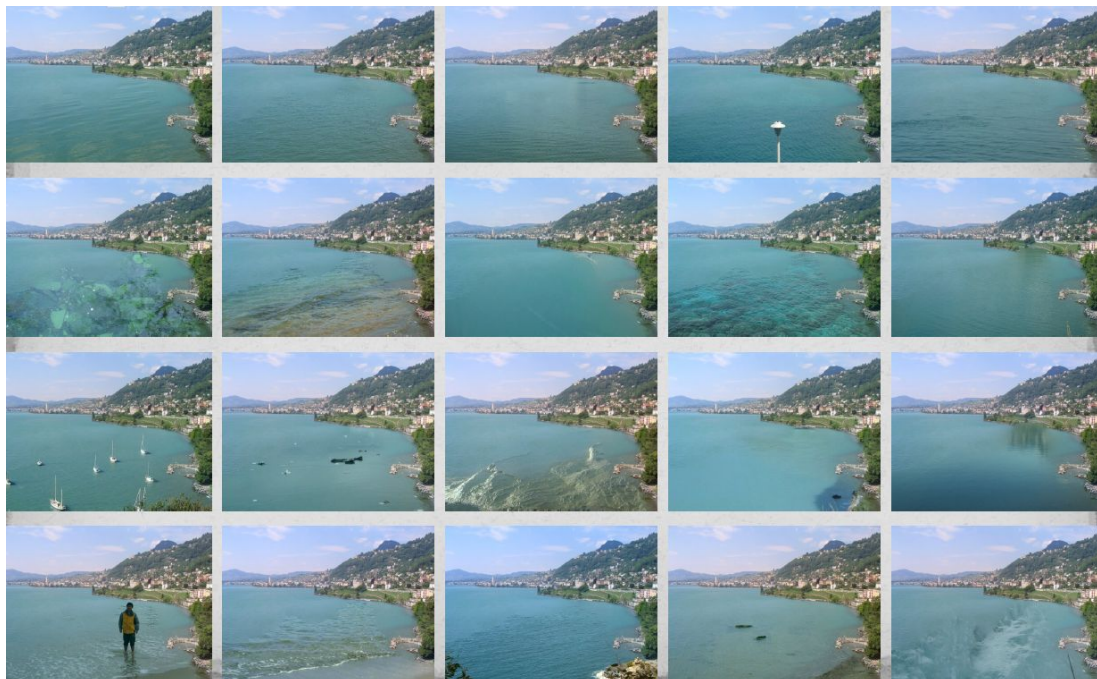
1、描述符的距离

+

2、重合区域的匹配程度

+

3、Graph-Cut的代价



top 20 结果

【结果分析】

先看看该算法结果和传统图像补全算法的结果对比：



原图



待补全



传统算法



该论文算法

可以看出，该论文的算法效果明显好于传统图像补全算法。

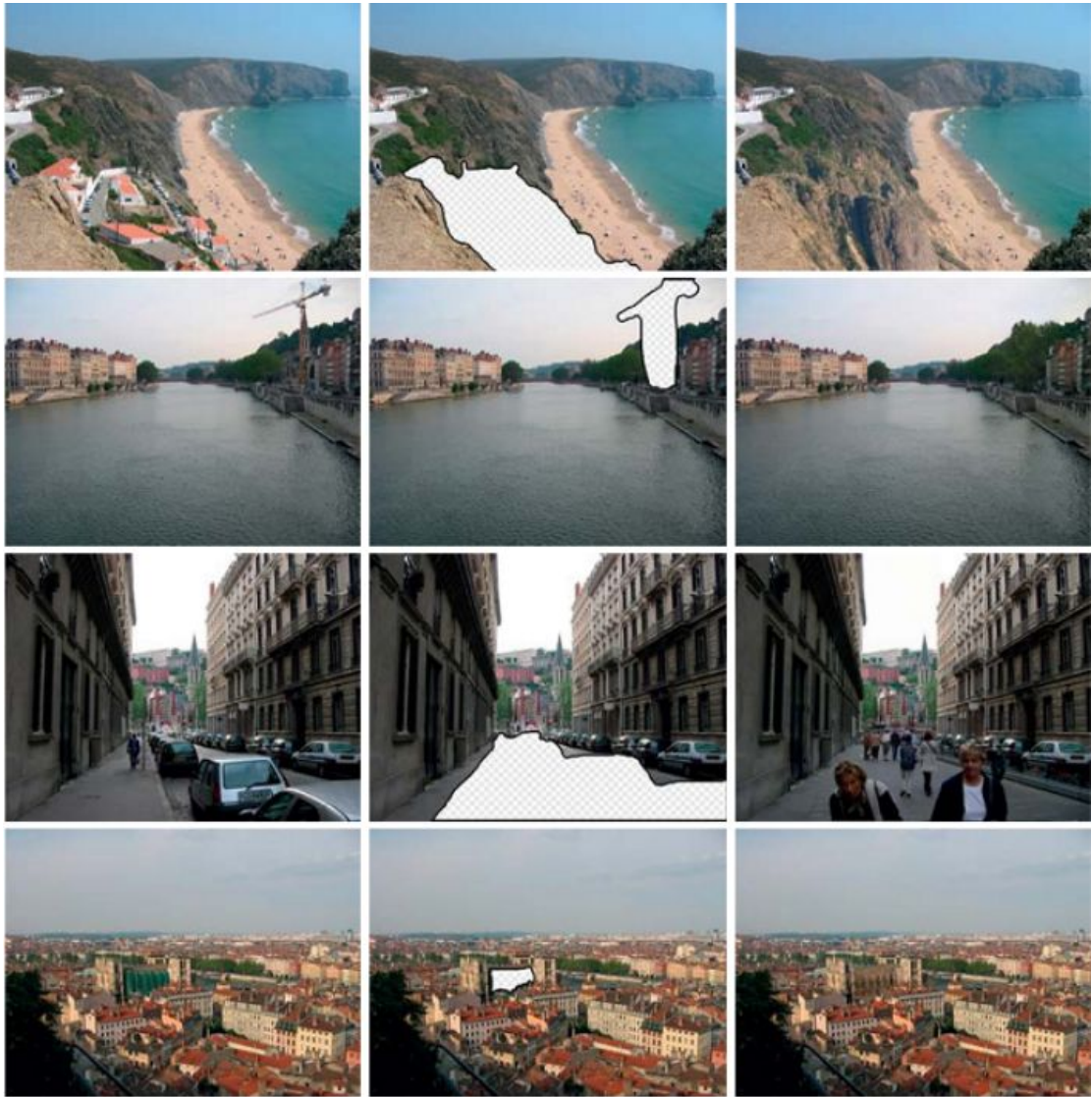
再看看一些其他比较好的结果：



Original

Input

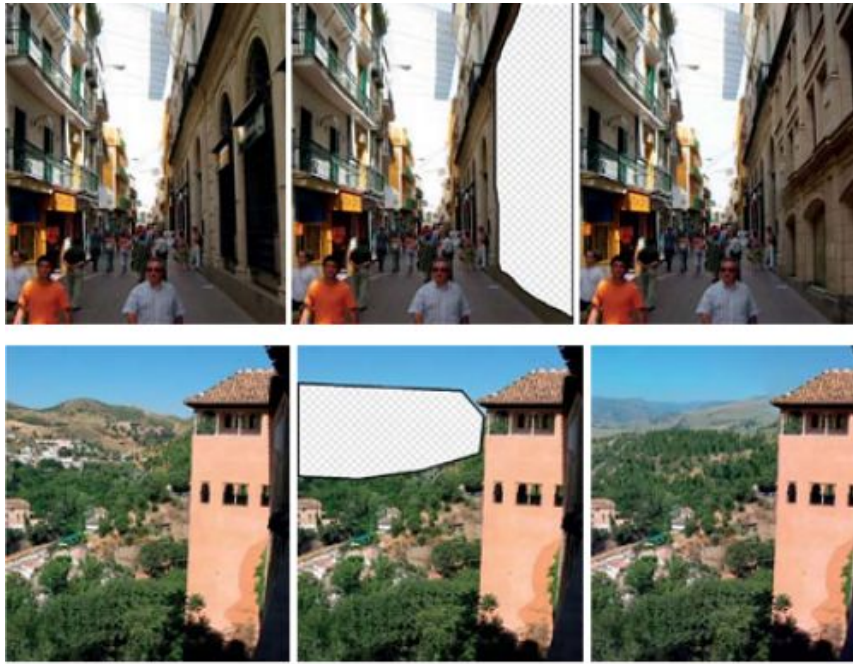
Alternative completions



Original image

Input

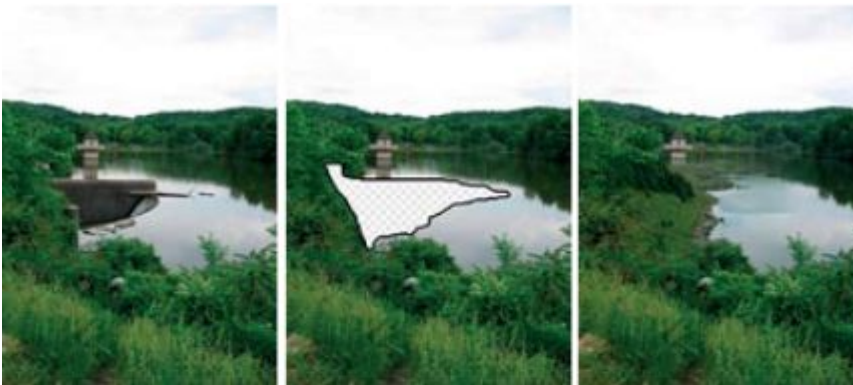
Output



Original image

Input

Output



Original

Input

Output

下面，我们再来看一下一些不太好的结果，我会分析一下为什么会出现这样的结果。



可以看到，这里匹配的场景有点牵强，这是因为这种场景不够典型，过于独特，数据库里面确实找不到比较好的匹配场景和匹配点，导致匹配效果不好。

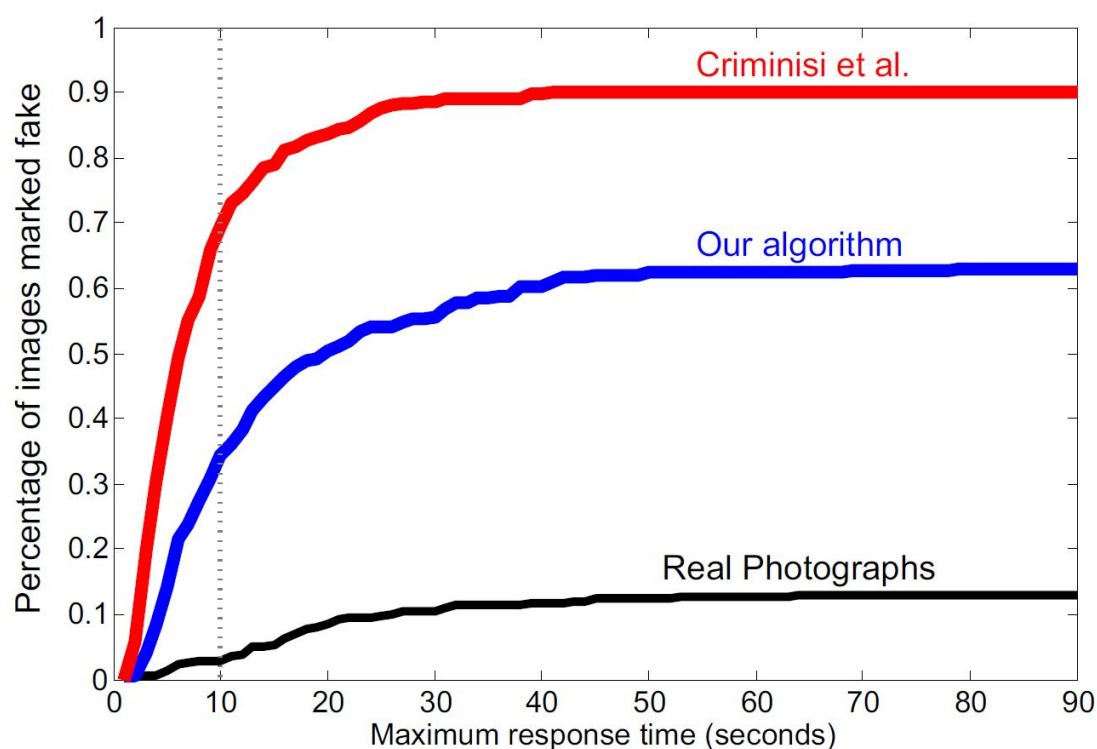


比如这张匹配结果，可以看到，人和周围的景色大小比例不合适。这是因为该算法并没有判断这种语义的功能，算法只管把补全的场景和原图尽量的进行无缝连接，并不检查连接后的语境和语义。



从这张图可以看出，红色箭头处有明显的 mismatch。这是因为图像由于分辨率不同，泊松混合以及纹理不同导致边界处模糊所致。

最后，我们来看一下论文中的一个趣味实验的结果：



该实验是请 20 个参与者对 51 张图片进行真假辨认。这 51 张图片有三个版本，分别是传统图像补全的算法结果（对应图中红色曲线），该论文算法结果（对应图中蓝色曲线），真实的图片（对应图中黑色曲线）。可以看出，在第 10s 的时候，有 2% 的人认为真实图片是假的，有 35% 的人认为该论文的结果是假的，有 70% 的人认为传统算法的结果是假的。作者通过这个实验来说明该算法结果的质量高于传统算法。

【论文创新点和优点】

1. 论文突破传统的图像补全思想，提出一种全新的图像补全理念，并成功完成了工程实现，取得了传统算法所不能达到的结果。

2. 算法中选用了 `gist` 特征，而不是一般的 `PCA` 等降维算法。这就使得特征提取可以使用于不同分辨率，不同尺寸的图像，使得各种各样的图像可以进行特征比较。而用 `PCA` 这种特征很难做到维度统一。这也是作者能够取得比较好效果的原因之一。

【论文存在的问题和改进方法】

1. 由于原始图片有接近 400G 的数据，所以对待补全的图像进行匹配时，搜索空间极大，耗时严重。

改进：对图像特征进行排序，构建 `kd-tree` 或二叉树，对图像搜索进行加速。

2. 该算法对图像进行拼接的时候，是直接把两张图片进行泊松融合，这就导致了前面结果分析当中出现的大小比例不合适的问题。

改进：对每张候选图像，进行等比例缩放，生成 50 张不同大小的图像，再和待补全图像进行泊松融合，最后选择 50 张融合图像中代价最小的一张作为这种候选图片的结果。

【参考文献】

[1] Scene Completion Using Millions of Photographs. J. Hays & A. A. Efros. ACM Transactions on Graphics 2007.

[2] Region Filling and Object Removal by Exemplar-Based Image Inpainting. A Criminisi et. al. Image Processing 2004.

[3] Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope, A Oliva, A Torralba, International Journal of Computer Vision, 2001, 42(3):145-175.