

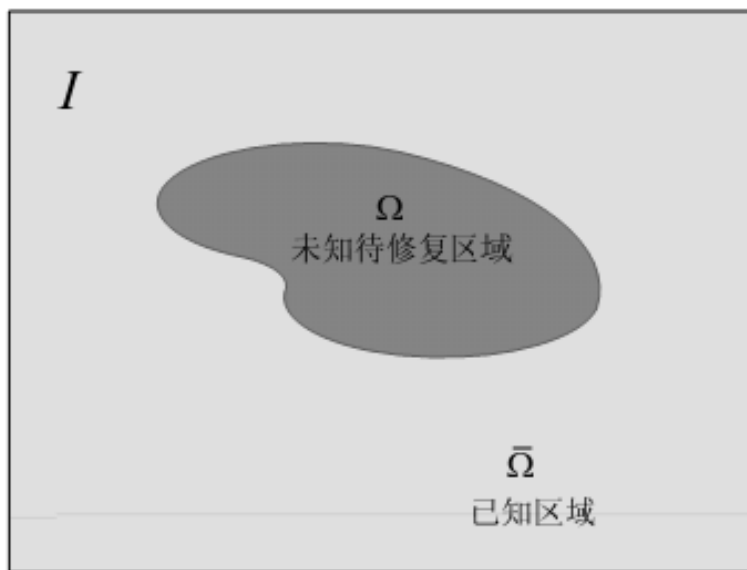


图像补全

章国锋/周晓巍

浙江大学CAD&CG实验室

问题描述



I 代表待修复图像， I 中深色区域 Ω 代表受损区域，也就是需要修补的区域，其余部分 $\bar{\Omega} = I - \Omega$ 为已知区域。Completion 即根据已知区域 $\bar{\Omega}$ 修复未知区域，得到重建区域 Ω' ，使得修复后的图像 $I' = \Omega' \cup \Omega$ 在视觉上自然

图像补全的应用

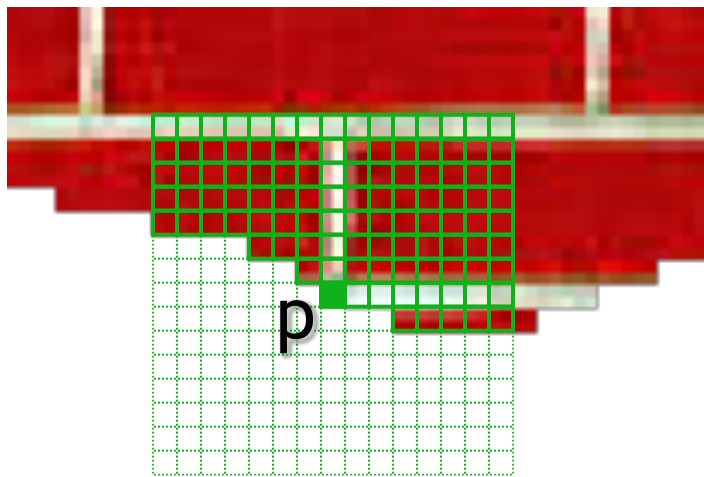
图像修复



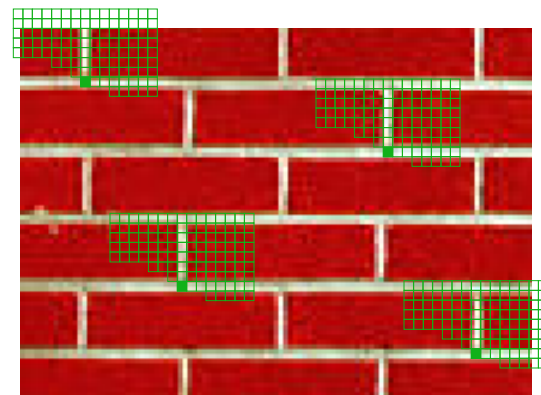
目标移除



基本的纹理合成算法

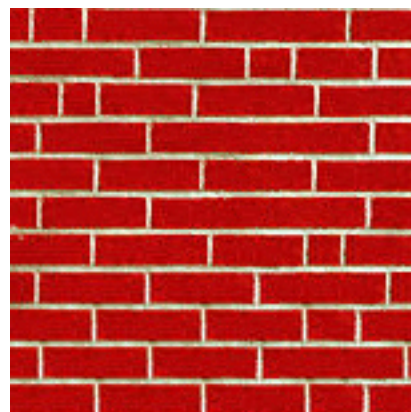


空洞的边界



已知的样本区域

经典的“剥洋葱”式填充顺序



经典的“剥洋葱”式填充顺序

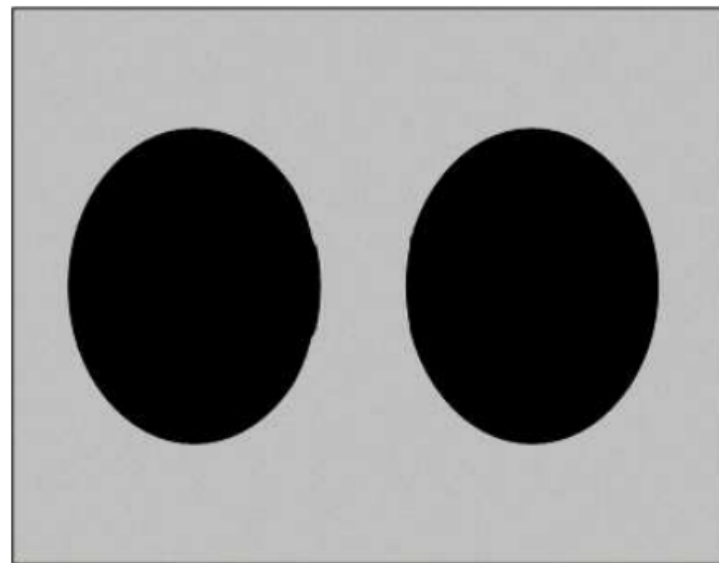
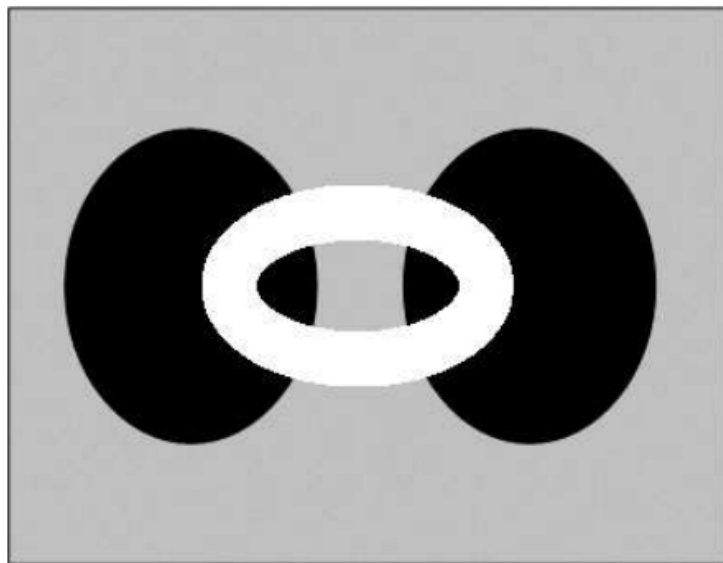


按照何种顺序进行填充是十分重要的

帶有優先級的填充策略

Image Completion by Example-Based Inpainting

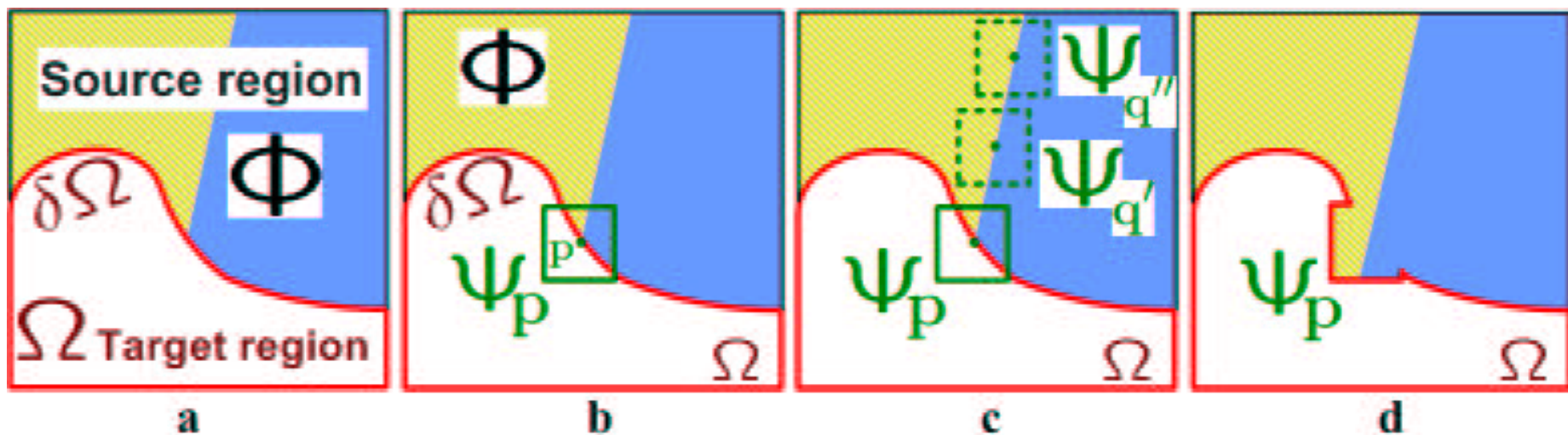
A. Criminisi, P. Perez, and K. Toyama,
CVPR 2003



带有优先级的填充策略

算法概览

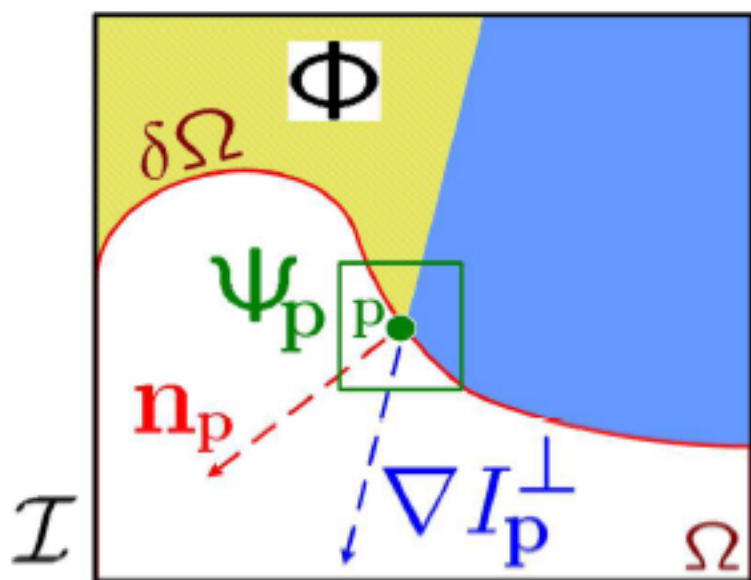
1. 用户选择需要补全的区域
2. 确定目前空洞边缘的像素位置
3. 为每一个像素计算优先级权重
4. 查找到优先级权重最大的像素位置 p , 并确定对应的块 P
5. 从图像已知区域匹配出最相似的块 S , 对 P 中不可见的像素进行补全
6. 更新优先级权重
7. 重复2-6步骤, 直到所有的像素被修复



带有优先级的填充策略

- 优先级度量:

$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p})$$



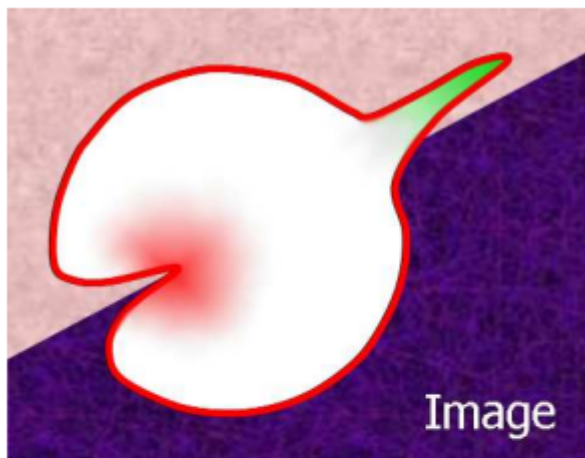
置信度项:

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap \bar{\Omega}} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|}$$

数据项:

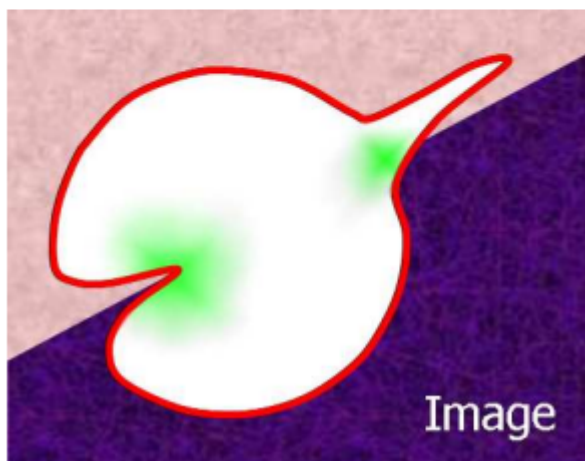
$$D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp} \cdot \mathbf{n}_{\mathbf{p}}|}{\alpha}$$

带有优先级的填充策略



a

a图显示置信度的分布, 绿色表示置信度高的区域
红色表示相对较低的取悦



b

b图显示数据项的分布, 绿色表示置信度高的区域

实验结果



a



b



c



d



e



f

实验结果



a



b

实验结果



a



b



c



d

实验结果



a



b

基本问题: 结构vs纹理



结构信息的补全比纹理信息的补全要困难的多, 能否通过添加一些交互来解决这个问题?



帶交互的補全算法

Image Completion with Structure Propagation

J. Sun, L. Yuan, J. Jia, and H. Shum
SIGGRAPH 2005

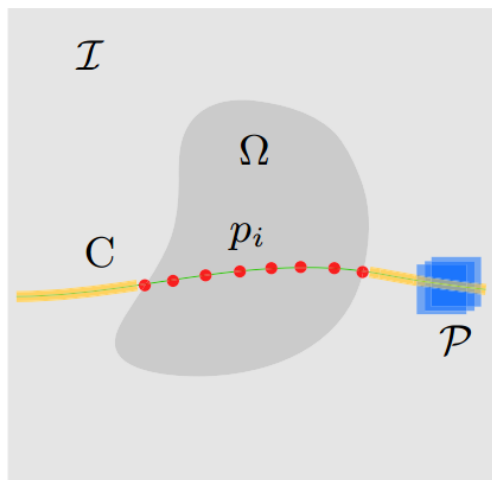
带有优先级的填充策略

算法概览



1. 用户输入：用户在空洞区域以及已知图像区域勾画结构线，
2. 结构补全：该算法在已知图像区域采样，通过优化一个目标能量来决定如何将样本填充被结构线覆盖的空洞区域
3. 纹理补全：补全剩余区域的纹理
4. 光测度修正

采样策略



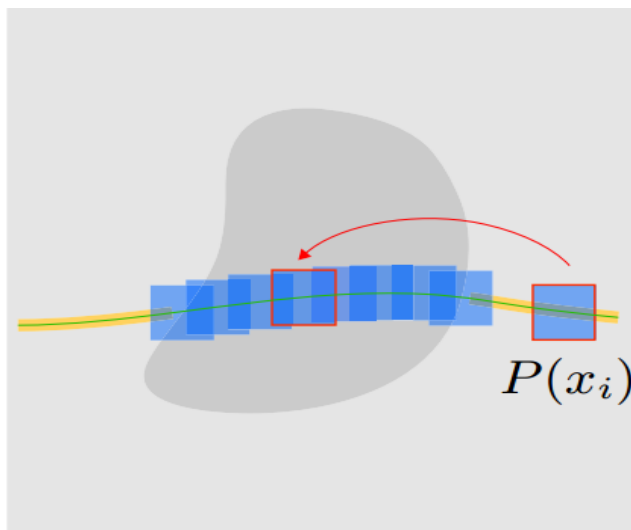
首先对处于 Ω 区域的 C 进行稀疏采样，依次得到 L 个锚点 $\{p_i\}_{i=1}^L$ 如左图所示，之后在 Ω 区域合成的块都是以这些锚点为中心的，这些锚点形成一条链或者可以认为是一个一维的图 $G = \{v, \varepsilon\}$ 。

抽样间隔通常是块大小的一半，这样可以保证有足够的重叠使得修补后的图像结构变的更加的平滑。在 Ω 外，沿着曲线 C 进行采样，一般抽样间隔为 1-5 个像素，得到样本块集

$$P = \{P(1), P(2), \dots, P(N)\} \leftarrow$$

目标能量

对于每一个锚点 p_i 我们找到一个标签 $x_i \in \{1, 2, \dots, N\}$ 对应于其中的一个样本块，
将样本块 $P(x_i)$ 复制到 p_i 的位置如下图所示。



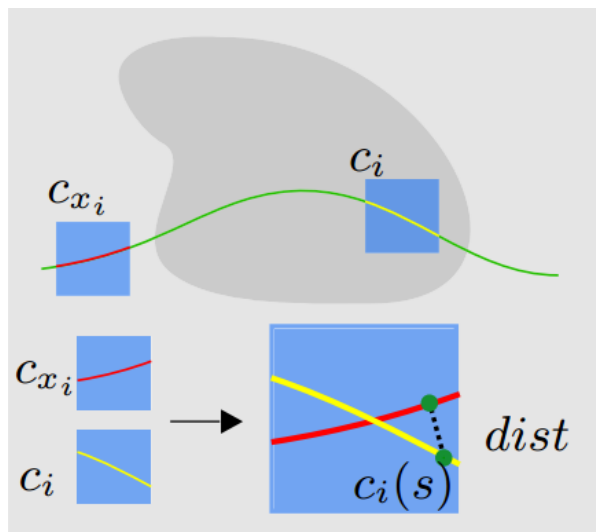
能量函数定义如下:

$$E(X) = \sum_{i \in v} E_1(x_i) + \sum_{(i,j) \in \mathcal{E}} E_2(x_i, x_j),$$

$$E_1(x_i) = k_s E_s(x_i) + k_l E_l(x_i).$$

$E_s(x_i)$, $E_l(x_i)$ 和 $E_2(x_i, x_j)$ 分别表示结构, 边界和一致性约束。

结构约束



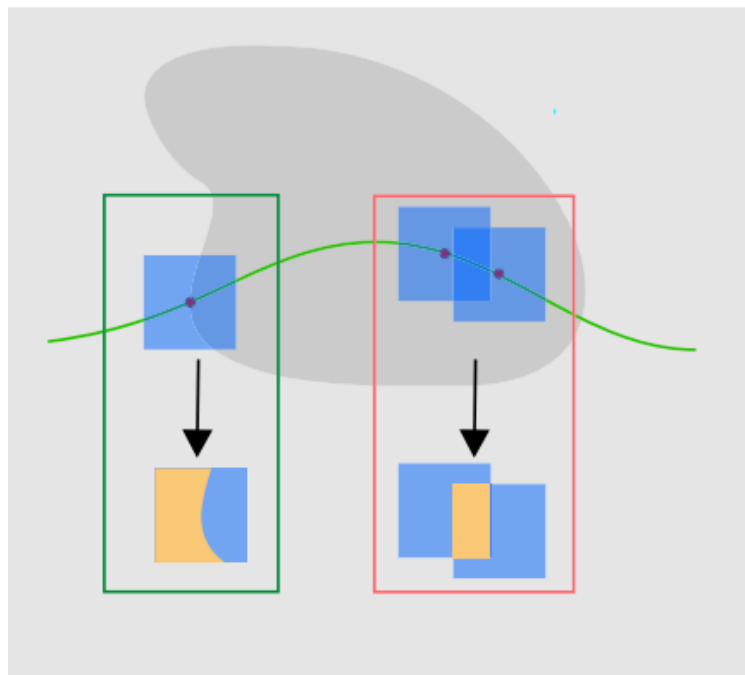
$E_s(x_i)$ 表示样本块 $P(x_i)$ 与节点 i 处结构线的相似

性。假设样本块 $P(x_i)$ 和定位于 p_i 的大小相同的矩形框中分别包含的曲线段 c_{x_i} 和 c_i ，如左图所示。

$$E_s(x_i) = d(c_{x_i}, c_i) + d(c_i, c_{x_i})$$

其中， $d(c_i, c_{x_i}) = \sum_s \|dist(c_i(s), c_{x_i})\|^2$ 是线段 c_i 上所有点到线段 c_{x_i} 的最短距离之和。 s 是线段 c_i 上的点的索引， $dist(c_i(s), c_{x_i})$ 是点 $c_i(s)$ 到线段 c_{x_i} 的最短距离。通过除以 c_i 点的个数进一步被归一化。

边界&一致性约束



$E_I(x_i)$ 约束 Ω 边界上的合成块，目的是为了使得边界上的合成块和边界处的已有图像信息尽可能的相似如左图绿色框所示。 $E_I(x_i)$ 等于样本块放在边界点处时与边界处已知图像区域重叠部分(绿框内黄色部分表示)归一化平方差和。对于 Ω 中的非边界块，该约束置为 0。↵

$E(x_i, x_j)$ 项表示向量两个节点处样本块的一致性约束， x_i 和 x_j 是临近两个节点的标签。该能量项被定义为他们重叠部分的归一化 SSD，如图上中的红框所示。↵

目标能量优化

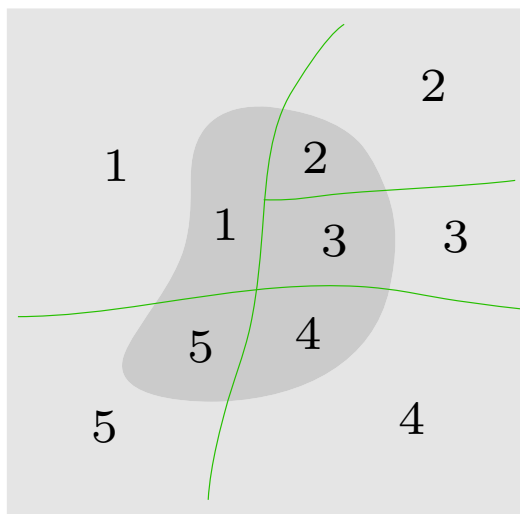
课后思考: 单结构线的优化如何用动态规划求解?

多结构线的优化需要,用Belief Propagation算法求解,
详见作者原文

纹理补全

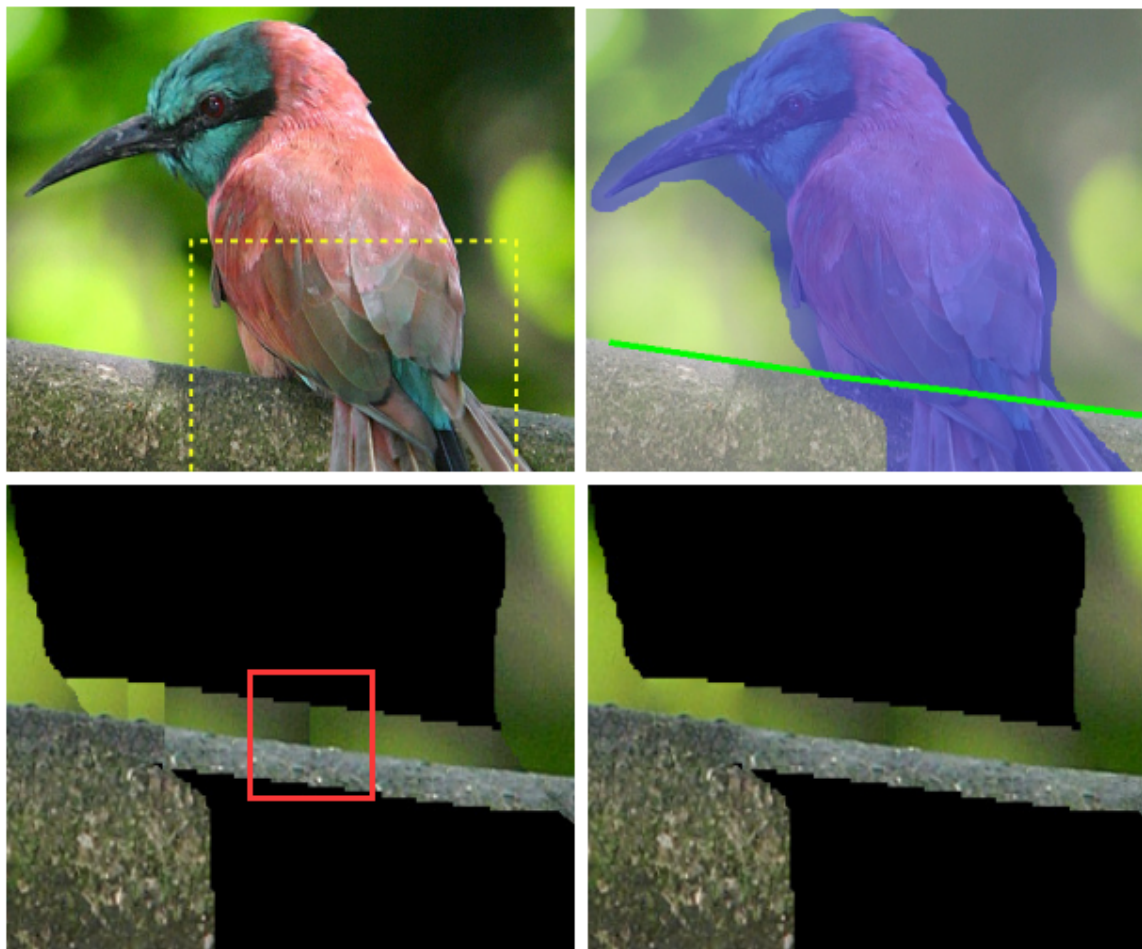


用户输入的曲线已经对图像做了分区，只从对应区域选取图像块对未知区域进行填充



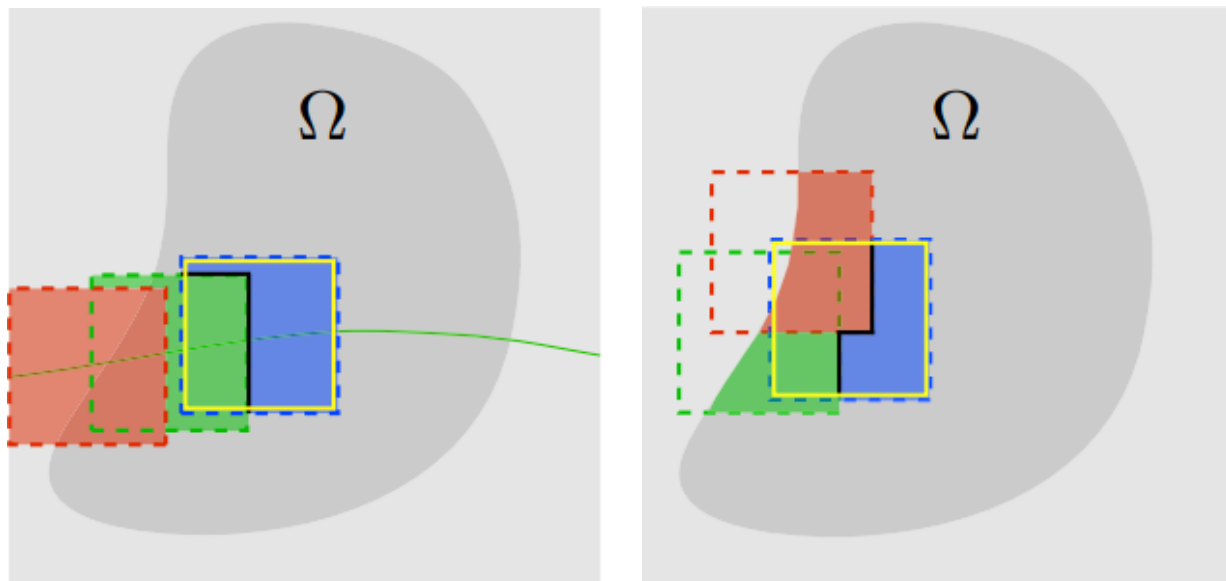
光测度修正

由于光照变化的影响, 容易产生色差缝



光测度修正

假设红色和绿色矩形块是已经被合成好的结构块，蓝色的块是正要合成的块。首先我们先将样本块中的像素拷贝到对应的蓝色区域的位置，得到一个新的在蓝色矩形区域的合成块 J 。之后我们构造一个二值化的掩膜块 B_M ，在绿色区域的值都置为 0，在蓝色区域的值都为 1。

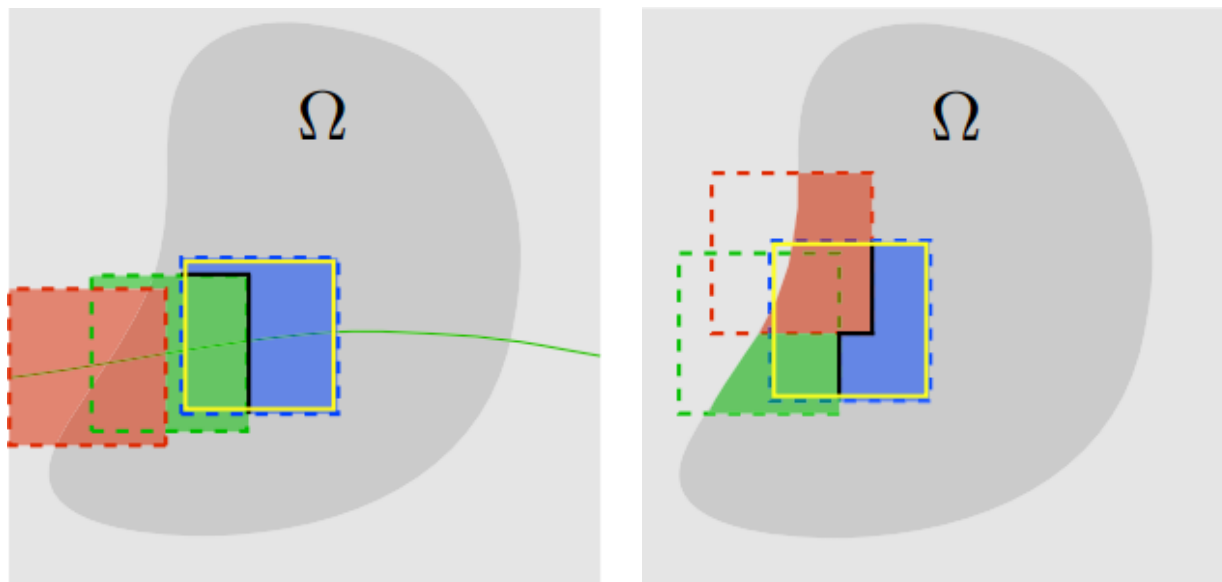


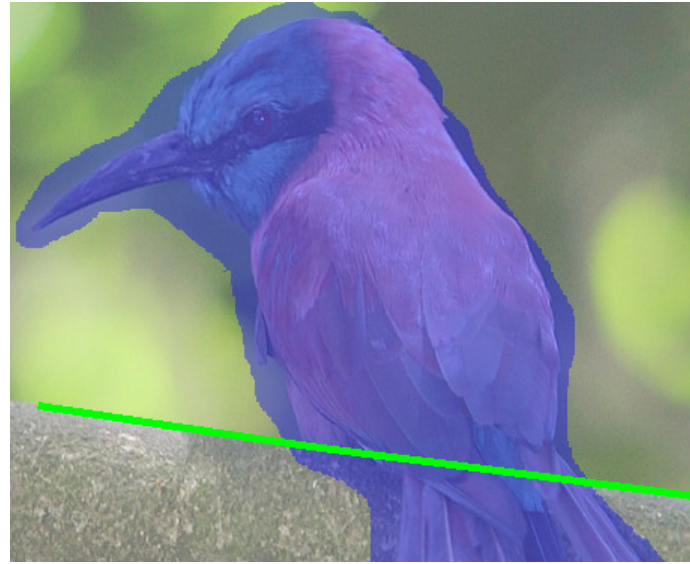
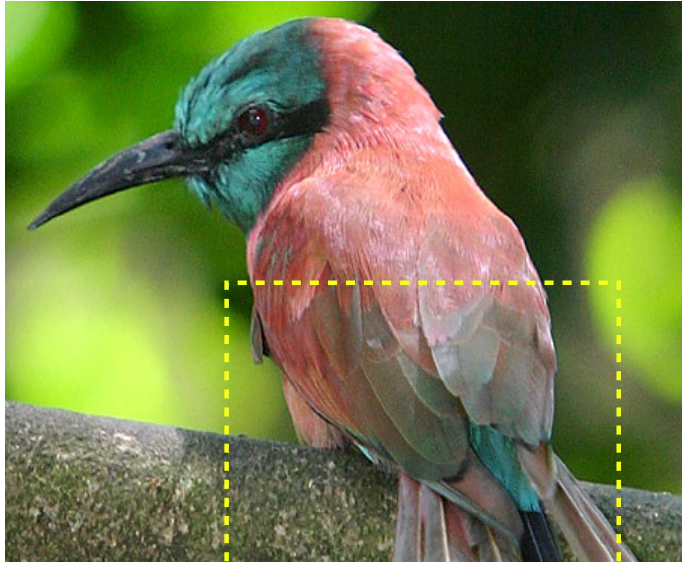
光测度修正

最后，我们通过修正 J 的梯度场 ∇J 重新构造一个新的块 J^* ，修正规则如下：↵

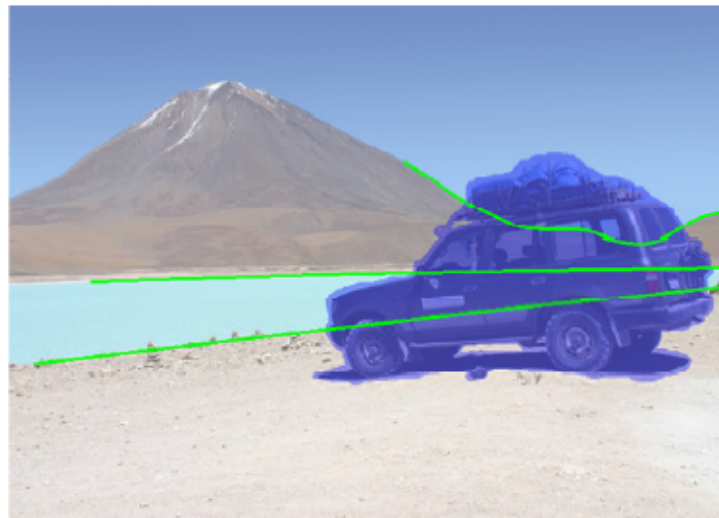
$$\nabla J_x(x, y)^* = \begin{cases} \nabla J_x(x, y) & B_M(x, y) = B_M(x+1, y) \\ 0 & B_M(x, y) \neq B_M(x+1, y) \end{cases}$$

$\nabla J_y(x, y)^*$ 采用相似的规则进行调整。狄利克雷边界条件即是内部的边界(J 最外面的一层像素，如黄色矩形框所示)。↵





实验结果

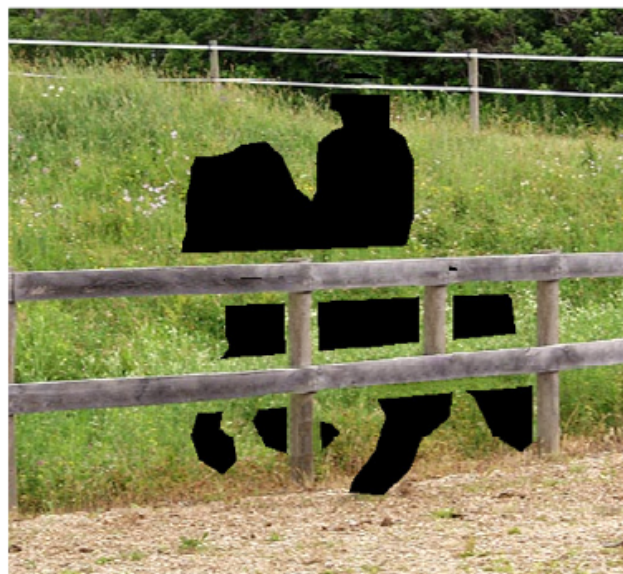
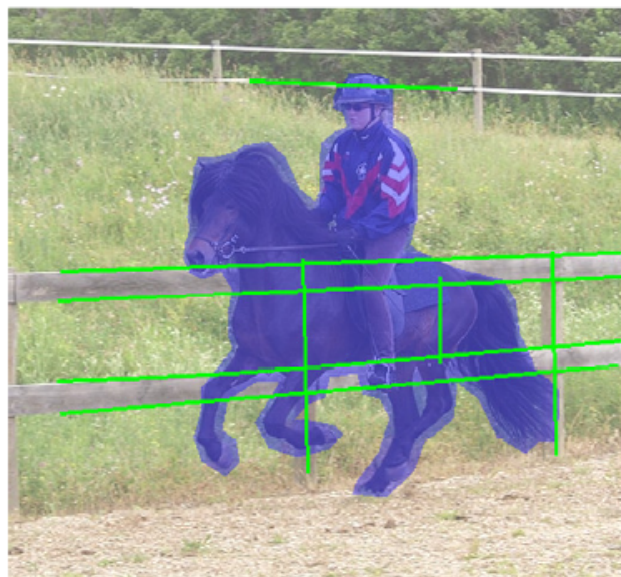


实验结果

Criminisi等人的方法



实验结果



实验结果

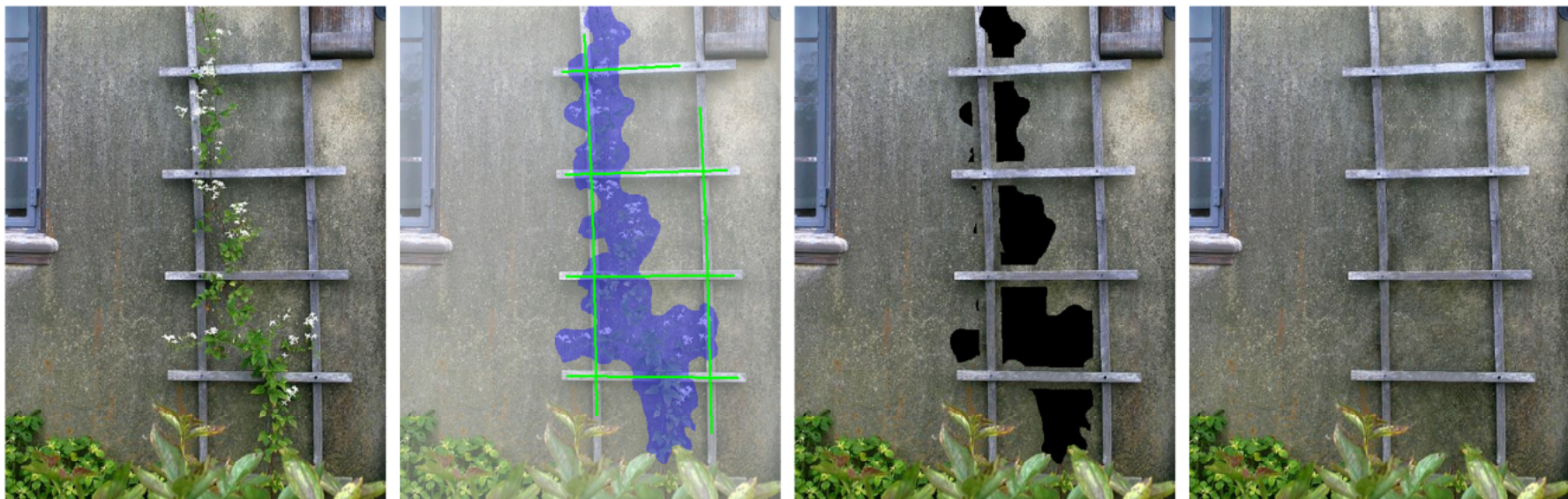
Criminisi等人的方法




实验结果



实验结果

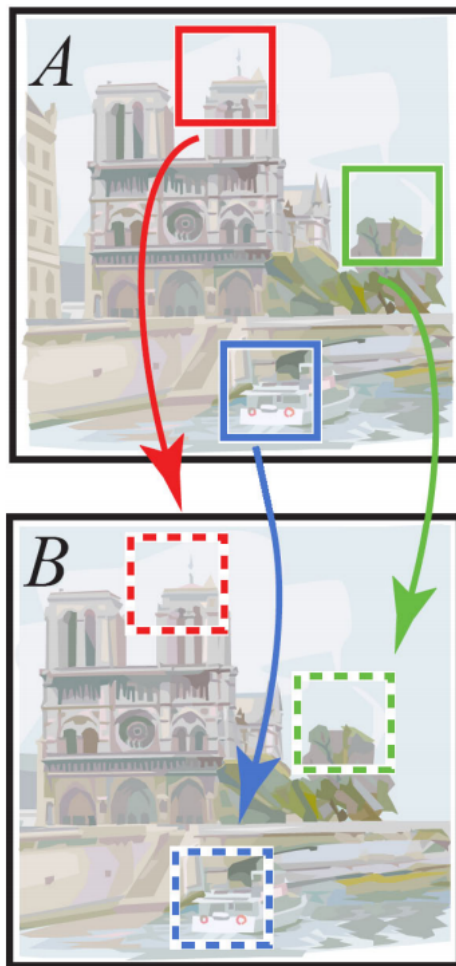




PatchMatch: A randomized correspondence algorithm for structural image editing

Barnes, C., Shechtman, E., Finkelstein, A
SIGGRAPH 2009

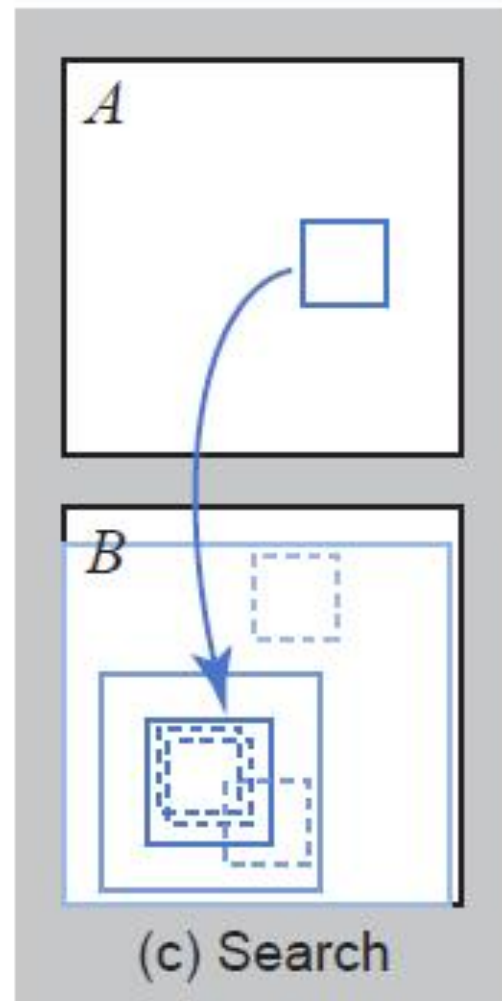
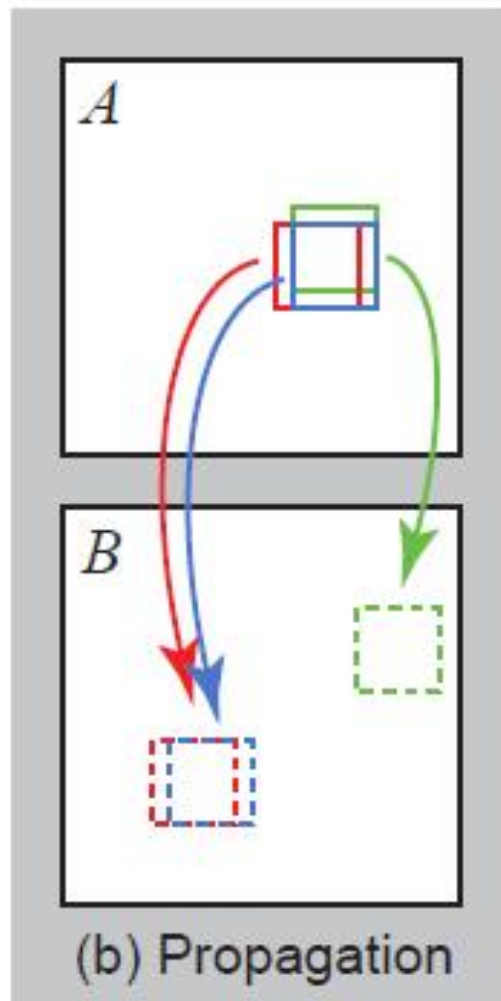
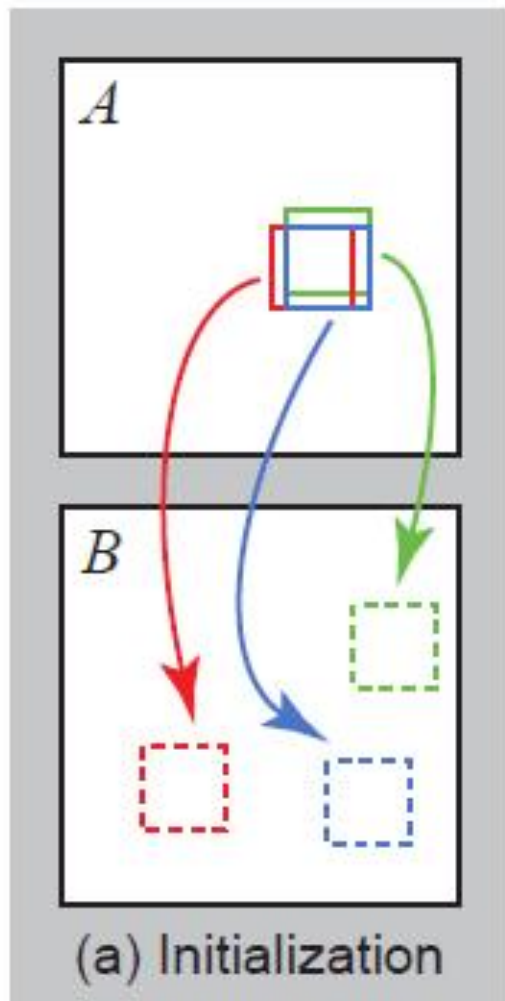
最近邻域(Nearest-Neighbor Field)



基本思想

1. NNF的计算可以转换为**offset**场的计算
2. 由于结构的重复性, 大部分的**patch**的**offset**都是相同的
3. 算法的开始可以为每个像素随机产生一个**offset**, 产生一个正确的**offset**的概率是很大的

算法流程



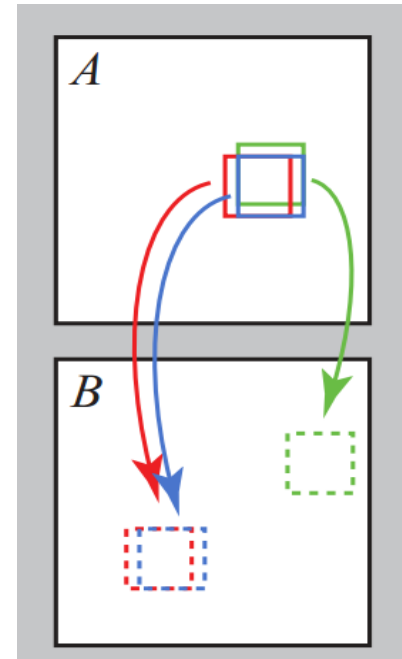
算法流程

迭代部分按照P1, S1, P2, S2,.....Pn, Sn进行,
其中P为扩散过程, 将周围好的offset扩散到当前像素
S为随机搜索过程, 在当前最好的匹配块的位置周围
多次随机寻找更优的匹配块

算法流程

扩散过程Pn:

- 1.当n为奇数的时候, 从左到右, 从上到下遍历所有的patch, 将上面和左边相邻的patch的offset传递到目前的patch, 选择最优的offset保留, 如图所示
- 2.当n为偶数的时候, 从下到上, 从右到左遍历, 选择下面和右边相邻的patch的offset传递到目前的patch

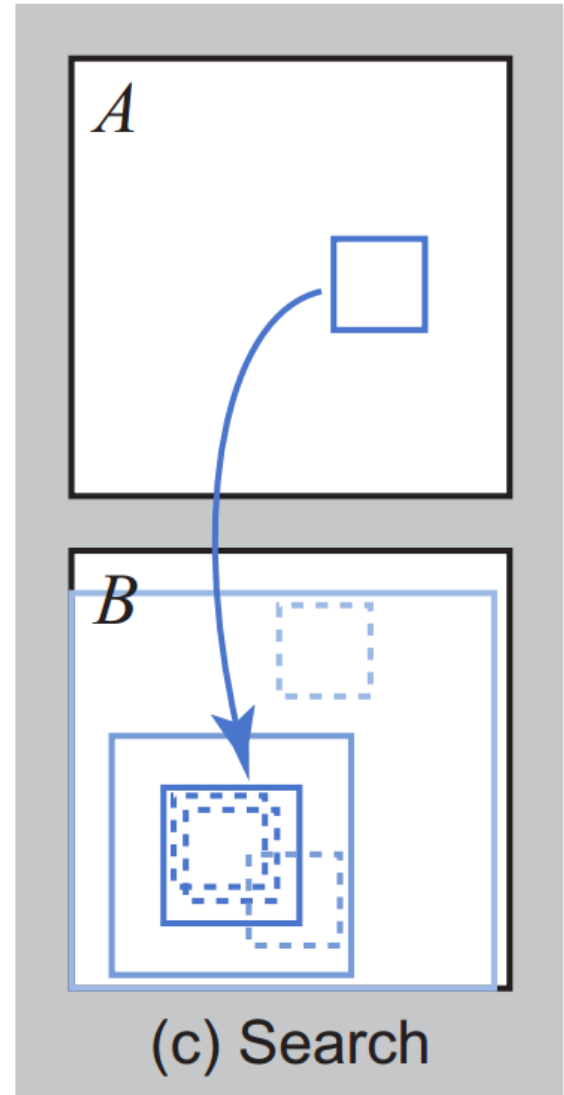


算法流程

随机搜索过程Sn:

$$\mathbf{u}_i = \mathbf{v}_0 + w\alpha^i \mathbf{R}_i$$

其中 \mathbf{v}_0 为原始的offset,新的offset \mathbf{u}_i 是以 \mathbf{v}_0 为中心, \mathbf{R}_i 由 $[-1,1] \times [-1,1]$ 上均匀分布产生的随机数, w 为最大搜索块半径, a 为收缩系数



PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing



(a) Original



(b) Inpainting



(c) Retarget



(d) Reshuffle

Connelly Barnes¹, Eli Shechtman^{2,3}, Adam Finkelstein¹, Dan B Goldman²

¹Princeton University, ²Adobe Systems, ³University of Washington



Scene Completion Using Millions of Photographs

James Hays and Alexei A. Efros
SIGGRAPH 2007







Criminisi et al. result



Criminisi et al. result



Scene Matching for Image Completion



Data

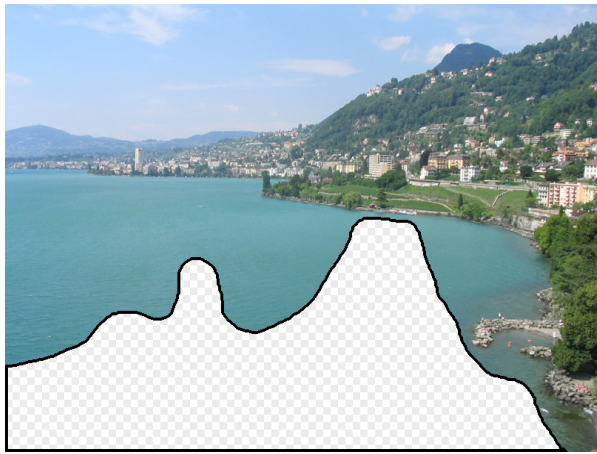
2.3 Million unique images from Flickr groups and keyword searches.



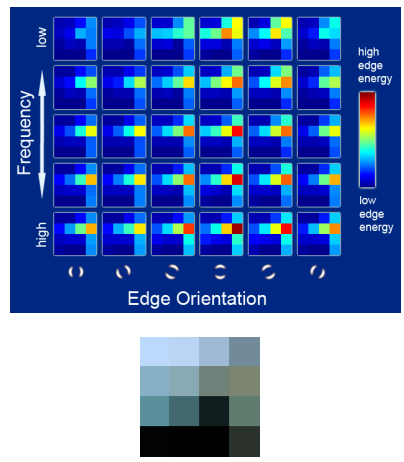


Scene Completion Result

The Algorithm



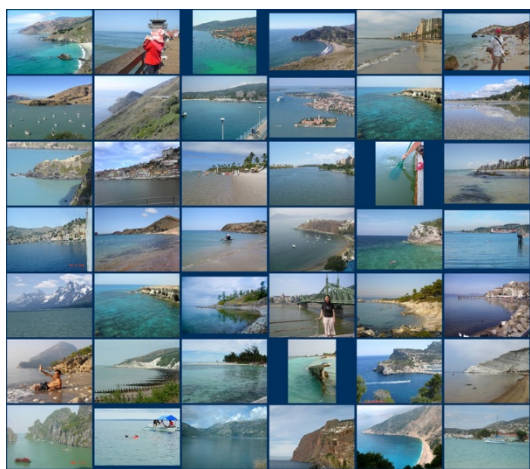
Input image



Scene Descriptor



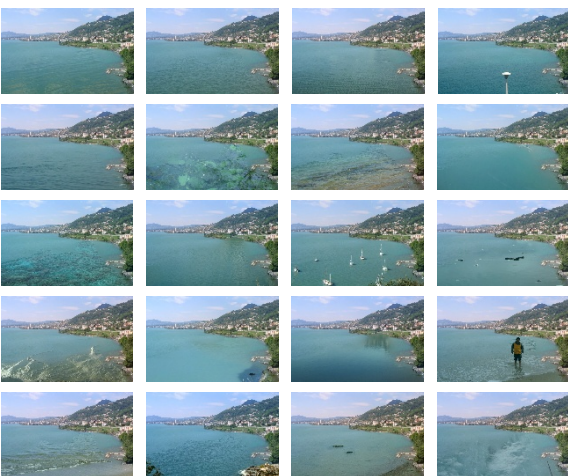
Image Collection



200 matches

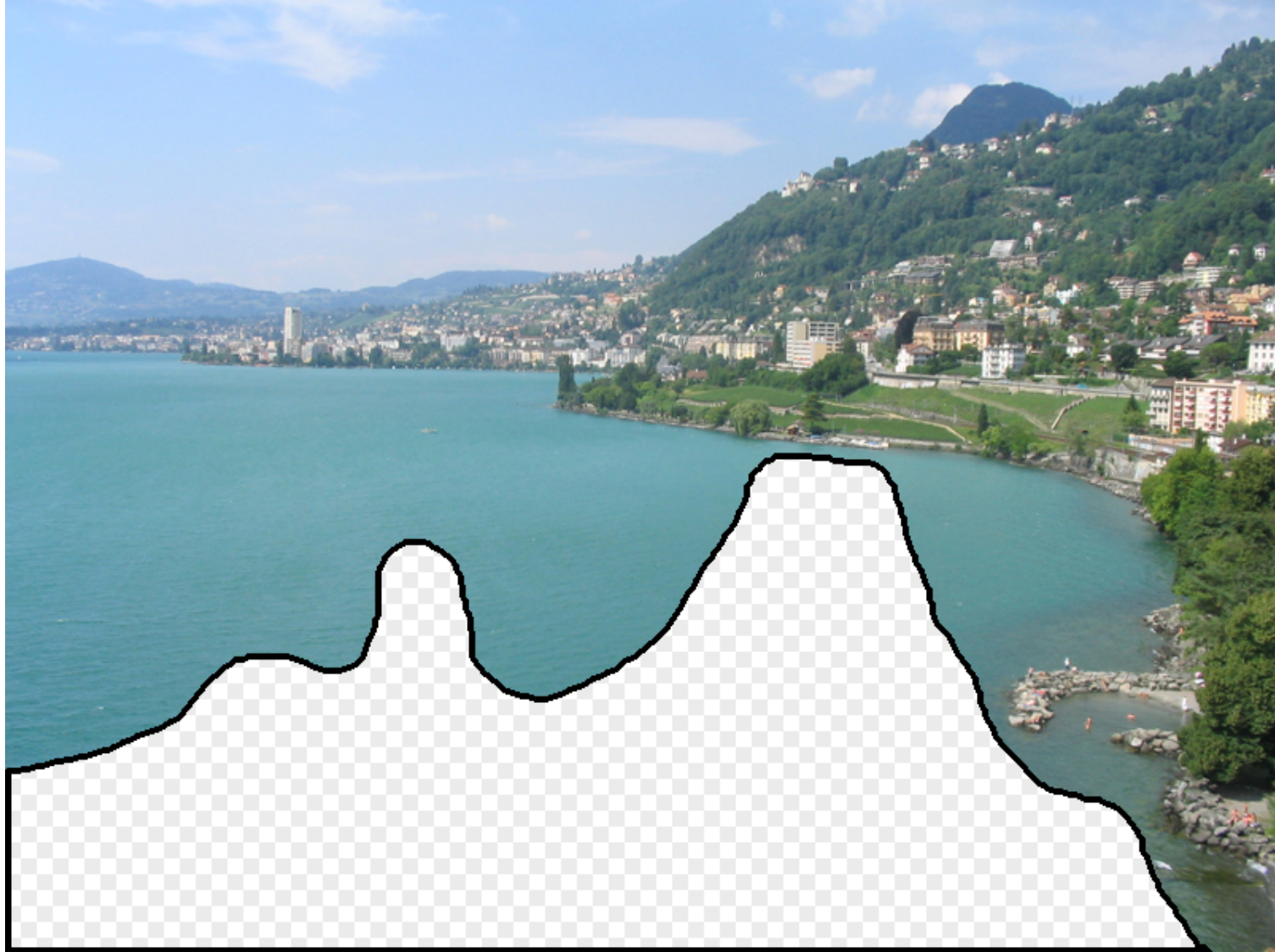


Context matching + blending

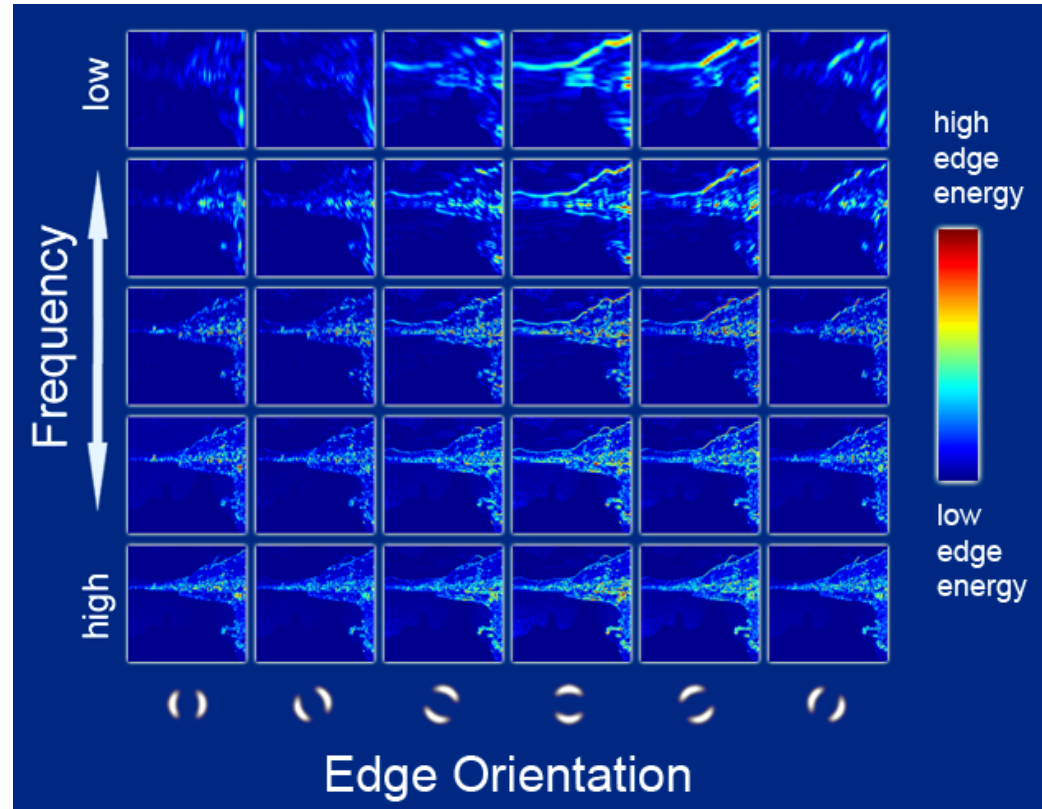
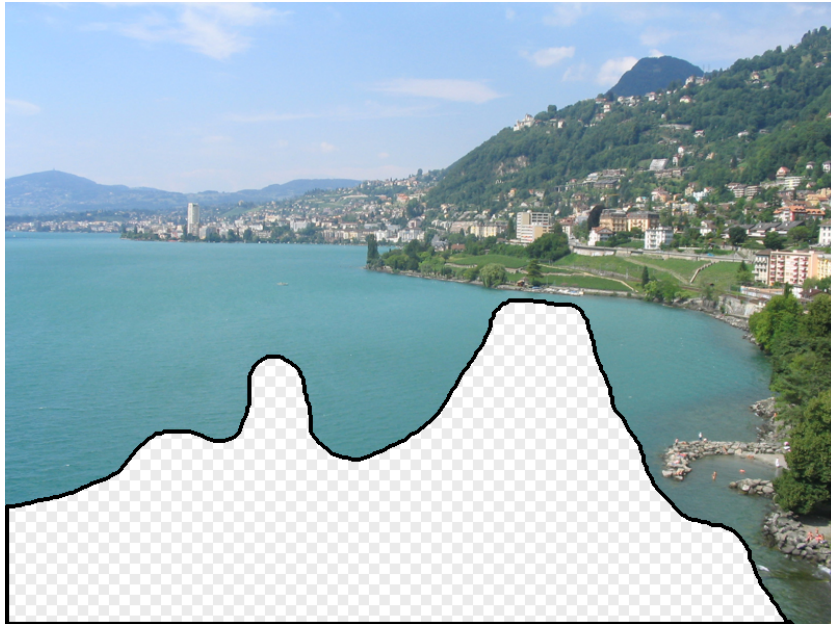


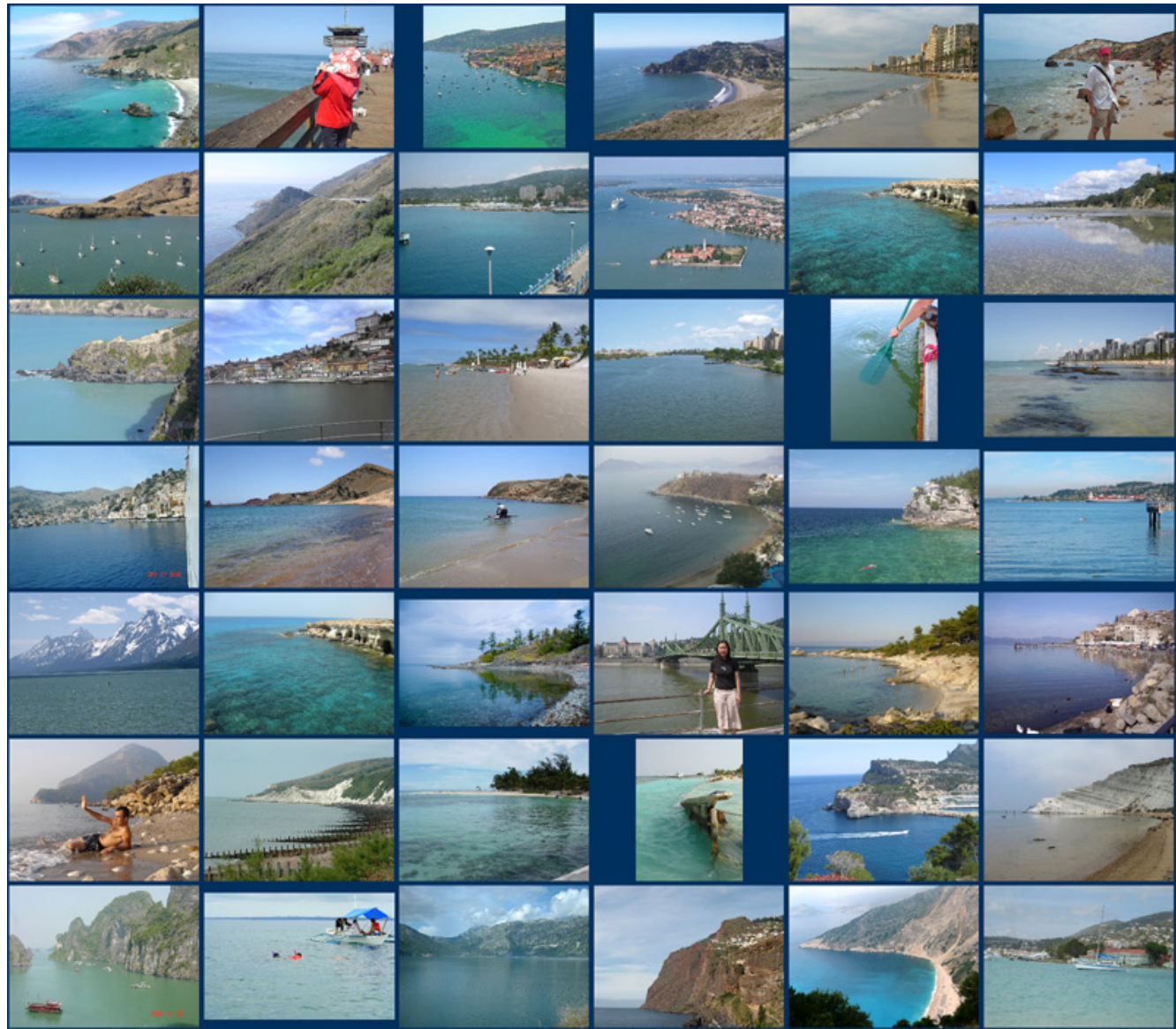
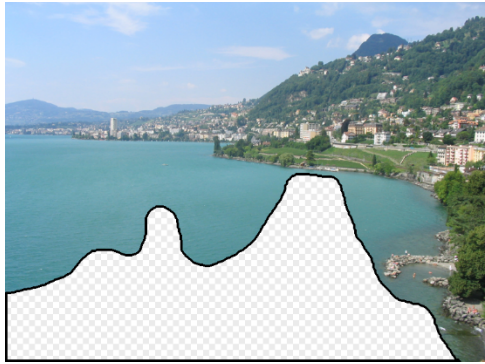
20 completions

Scene Matching



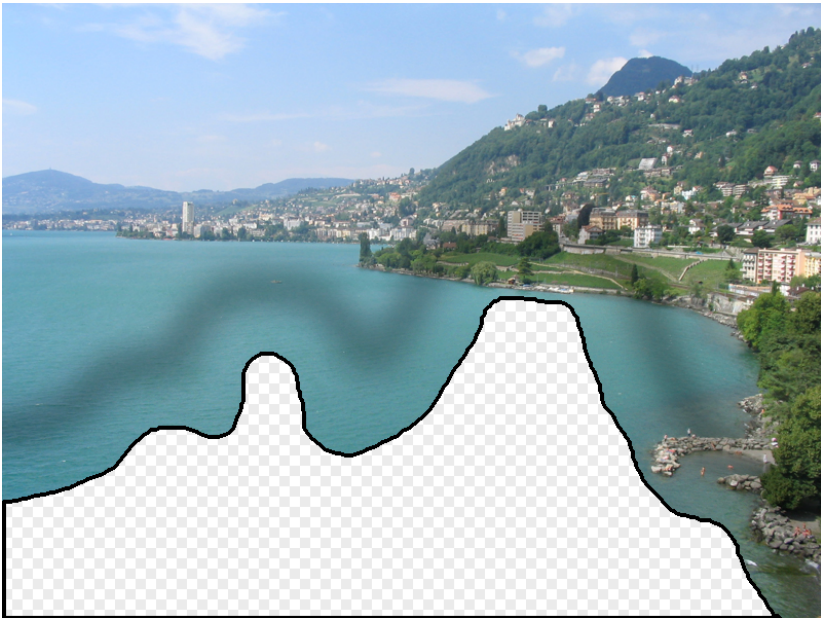
Scene Descriptor





... 200 total

Context Matching

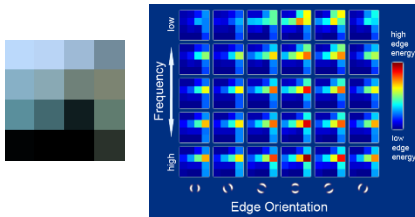






Result Ranking

We assign each of the 200 results a score which is the sum of:



The scene matching distance

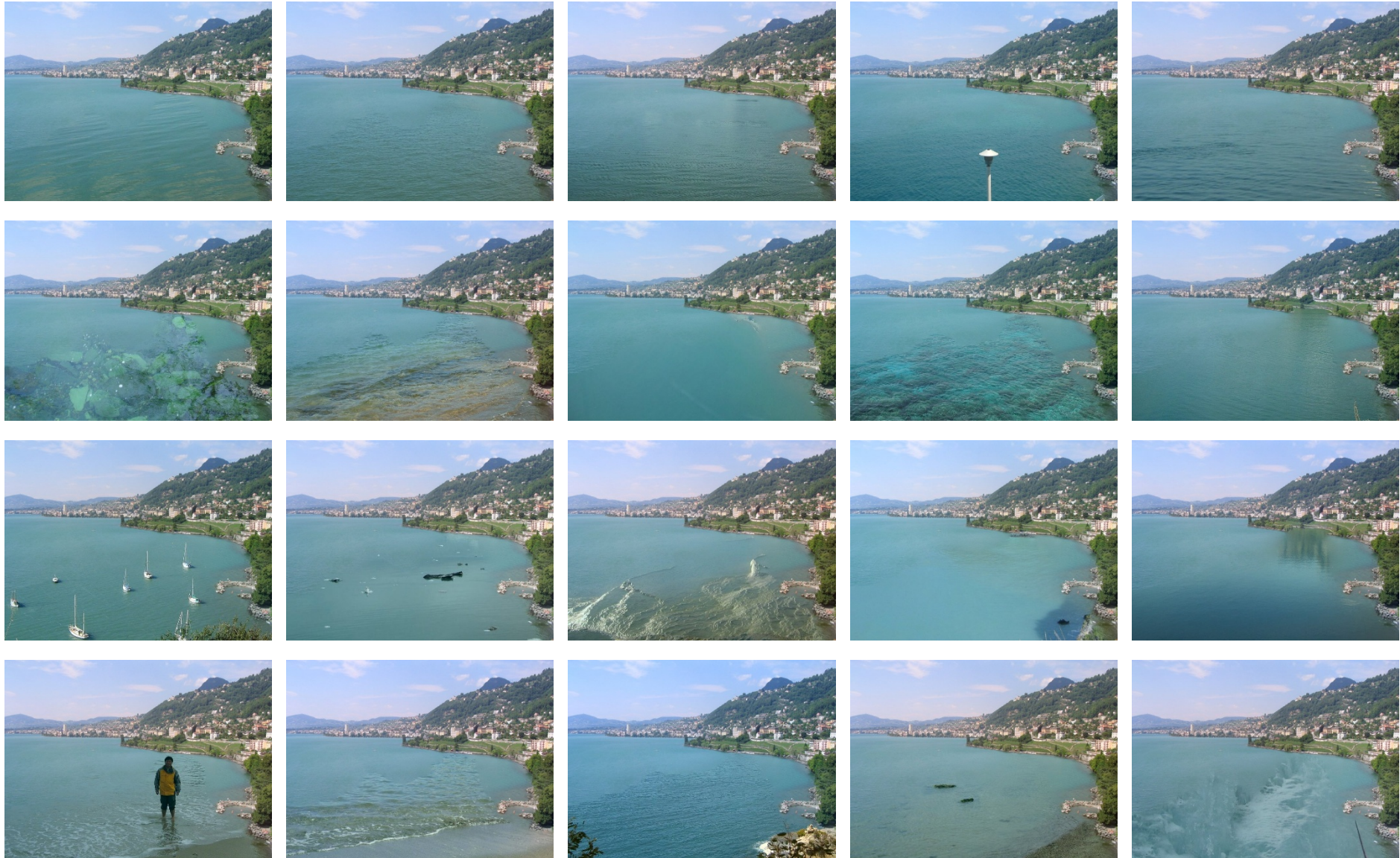


The context matching distance
(color + texture)



The graph cut cost

Top 20 Results











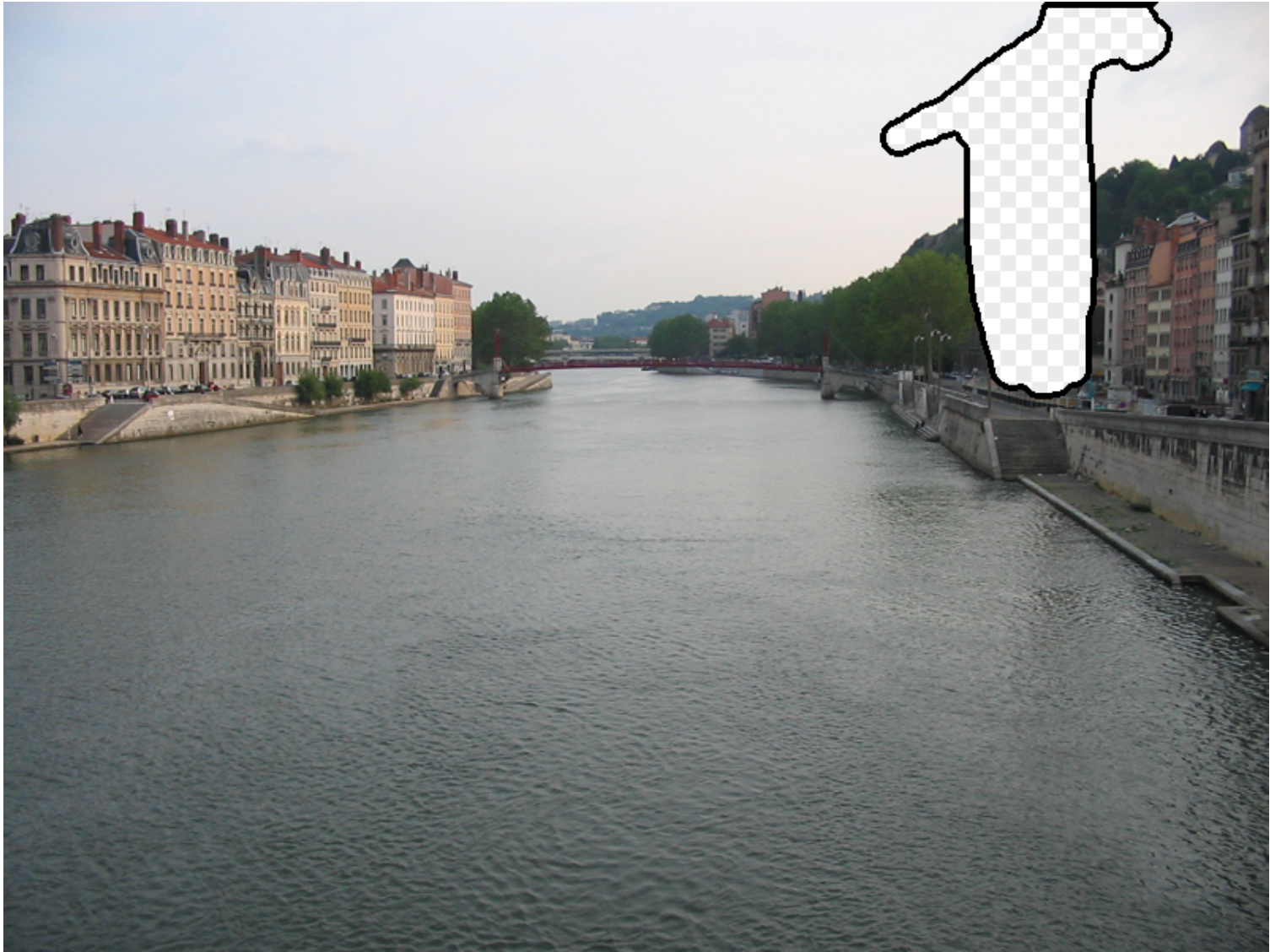






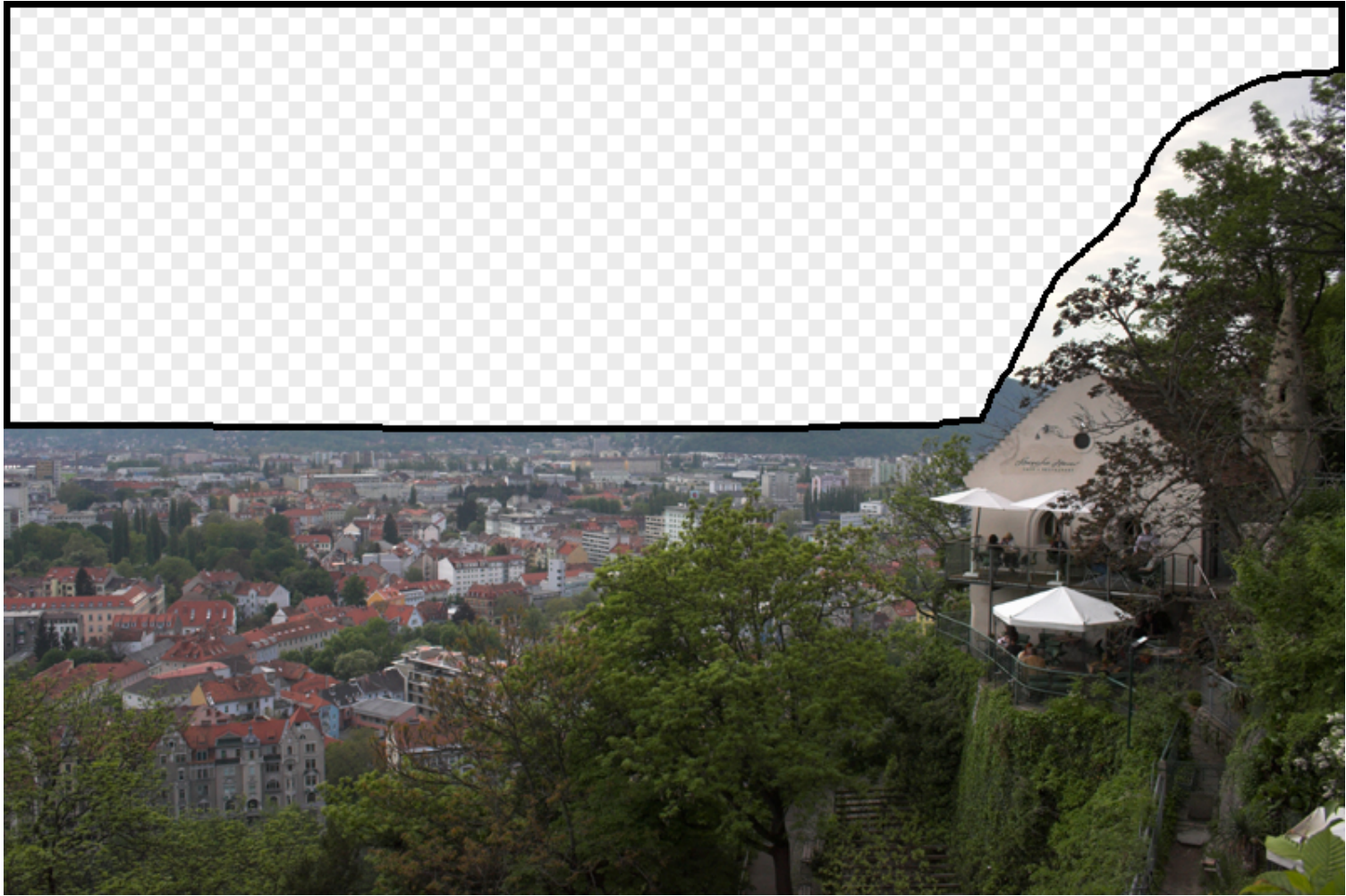




















Using CNNs?

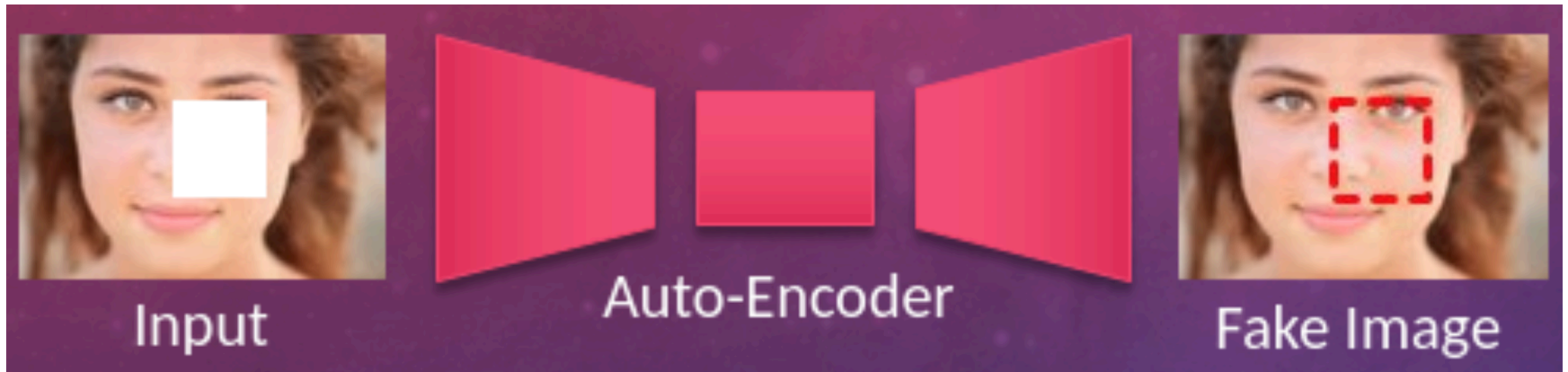
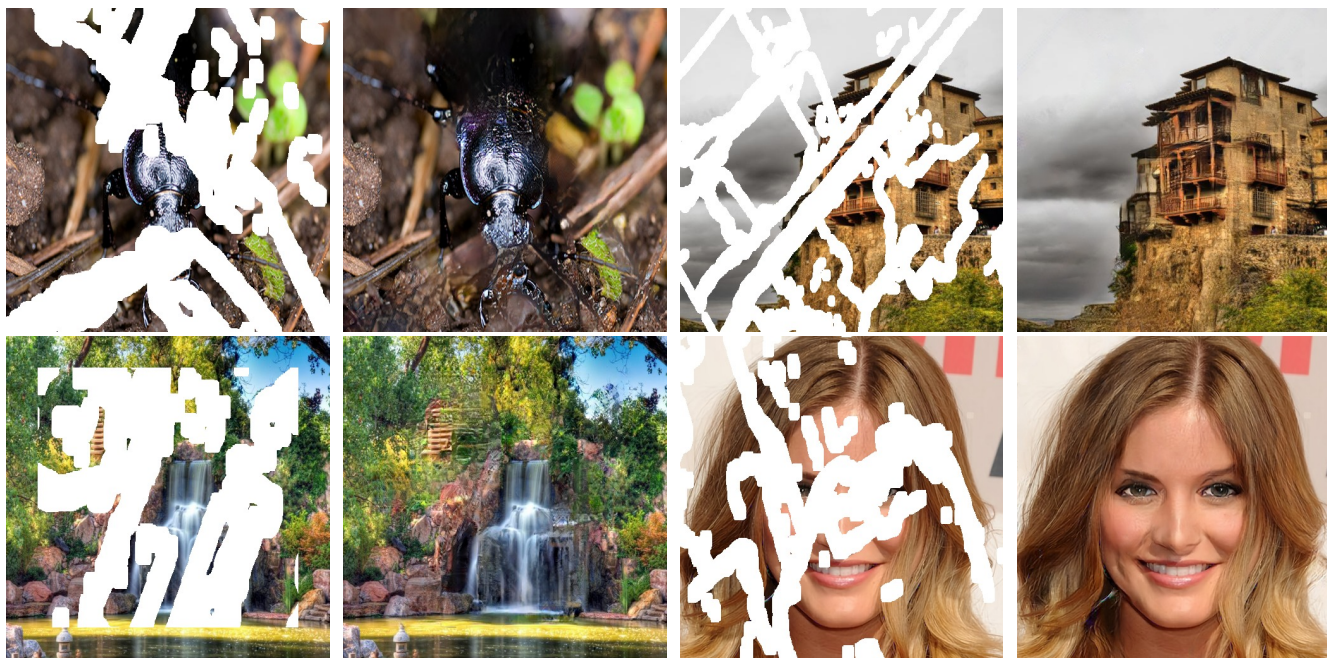


Image Inpainting for Irregular Holes Using Partial Convolutions

Guilin Liu Fitsum A. Reda Kevin J. Shih Ting-Chun Wang
Andrew Tao Bryan Catanzaro

NVIDIA Corporation



Free-Form Image Inpainting with Gated Convolution

JIAHUI YU, University of Illinois at Urbana-Champaign

ZHE LIN, Adobe Research

JIMEI YANG, Adobe Research

XIAOHUI SHEN, Adobe Research

XIN LU, Adobe Research

THOMAS HUANG, University of Illinois at Urbana-Champaign

