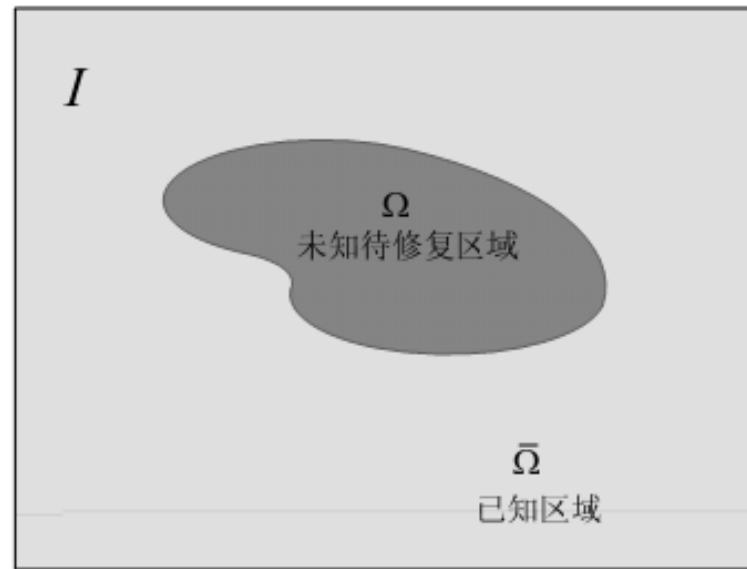




图像补全

陶煜波，鲍虎军，章国锋
浙江大学CAD&CG国家重点实验室

问题描述



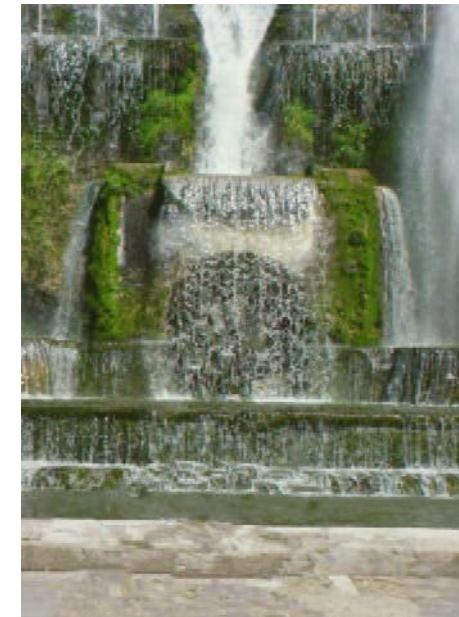
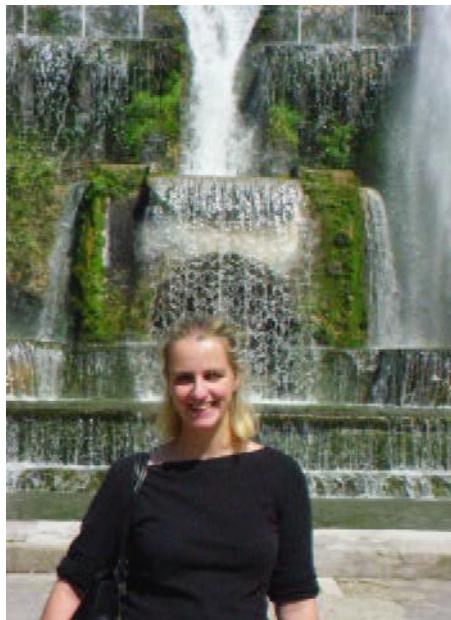
I 代表待修复图像， I 中深色区域 Ω 代表受损区域，也就是需要修补的区域，其余部分 $\bar{\Omega} = I - \Omega$ 为已知区域。Completion 即根据已知区域 $\bar{\Omega}$ 修复未知区域，得到重建区域 Ω' ，使得修复后的图像 $I' = \Omega' \cup \Omega$ 在视觉上自然。

图像补全的应用

图像修复



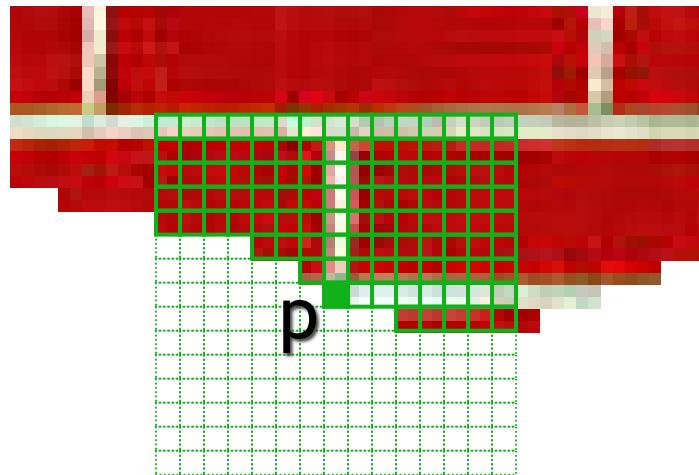
目标移除



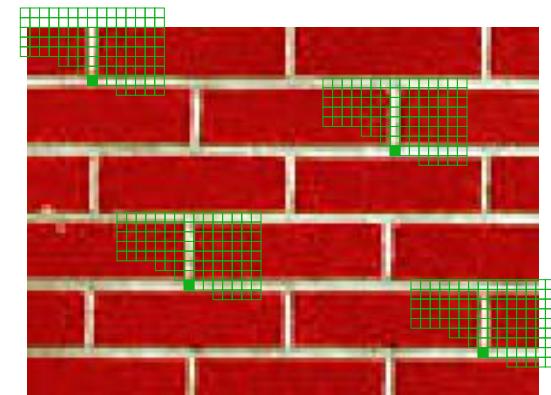
图像补全

- 思想：图像自身或图像库信息冗余性
- 关键：按照何种顺序进行填充
- 纹理
 - 基本的纹理合成算法
 - 经典的“剥洋葱”式填充顺序
 - 带有优先级的填充策略
- 结构
 - 带交互的补全算法
 - 局部优化算法

基本的纹理合成算法



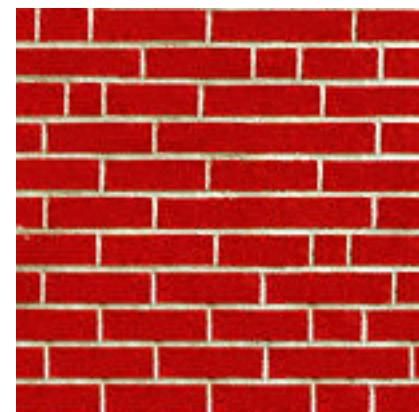
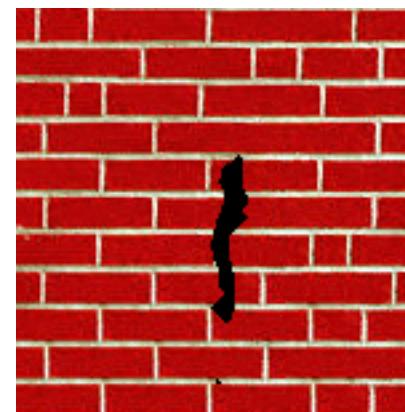
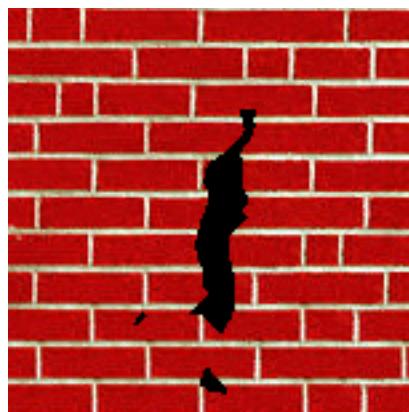
空洞的边界



已知的样本区域

按照何种顺序进行填充是十分重要的

经典的“剥洋葱”式填充顺序



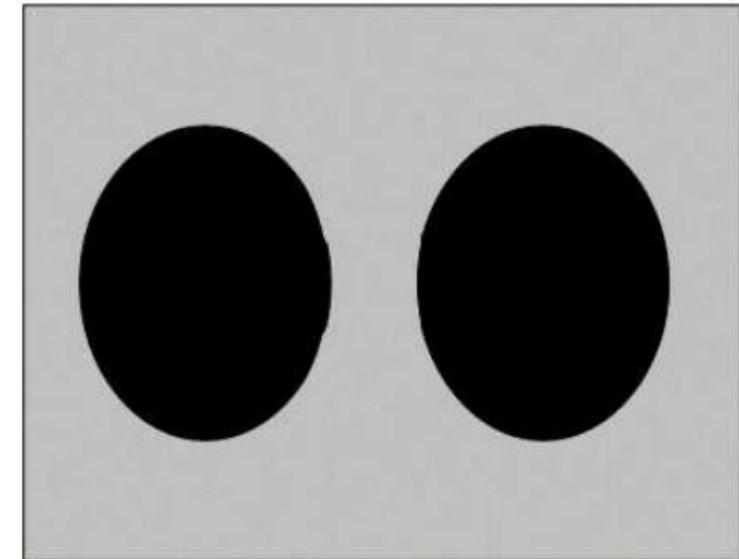
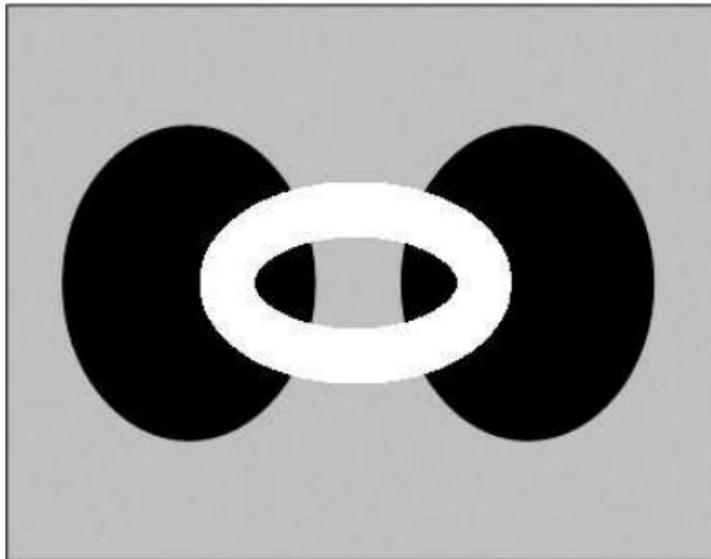
经典的“剥洋葱”式填充顺序



带有优先级的填充策略

Image Completion by Example-Based Inpainting

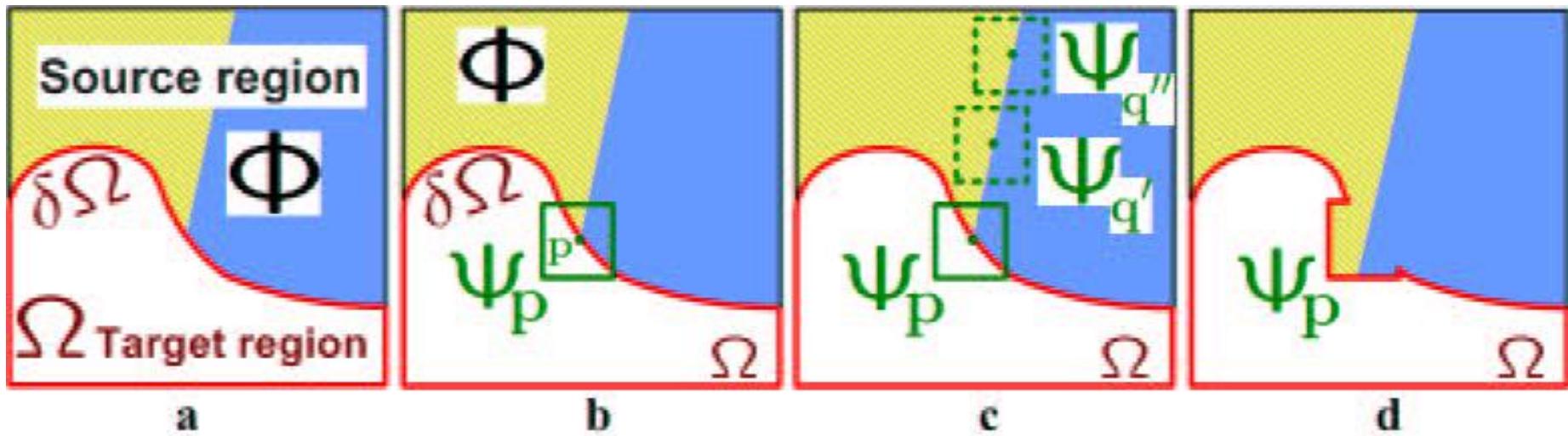
A. Criminisi, P. Perez, and K. Toyama,
CVPR 2003



带有优先级的填充策略

算法概览

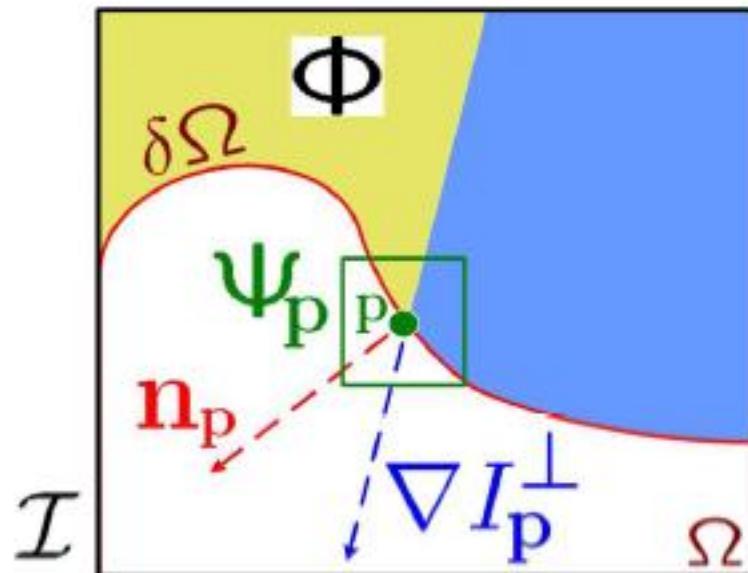
1. 用户选择需要补全的区域
2. 确定目前空洞边缘的像素位置
3. 为每一个像素计算优先级权重
4. 查找到优先级权重最大的像素位置 p , 并确定对应的块 P
5. 从图像已知区域匹配出最相似的块 S , 对 P 中不可见的像素进行补全
6. 更新优先级权重
7. 重复2-6步骤, 直到所有的像素被修复



带有优先级的填充策略

- 优先级度量:

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



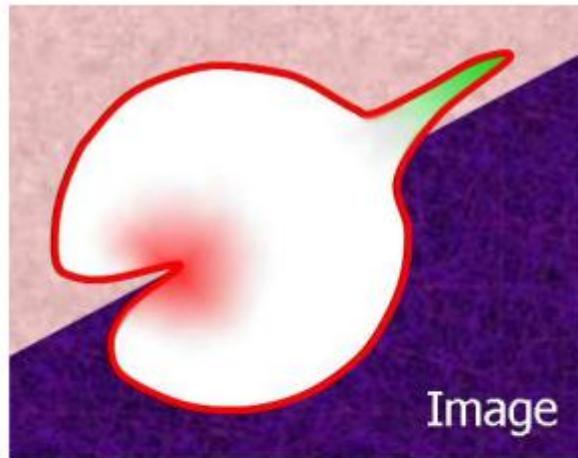
置信度项:

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

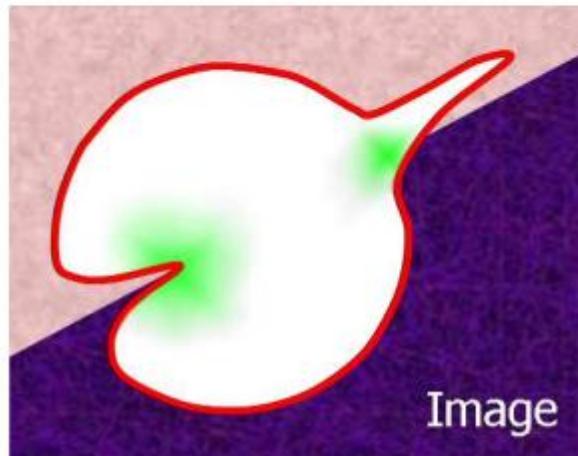
数据项:

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

带有优先级的填充策略



a



b

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区域

b图显示数据项的分布, 绿色表示数据项值高的区域

带有优先级的填充策略

- $C(p)$ 初始化值: 空洞区域像素为0, 已知区域部分的像素设置为1
- 相似度度量: CIE Lab空间上的Sum of Squared Difference (SSD)
- 优先级权重更新策略:
只更新 $C(p)$ 项, 更新后的值为块中心点 p_0 的置信度值

实验结果



a



b



c



d



e

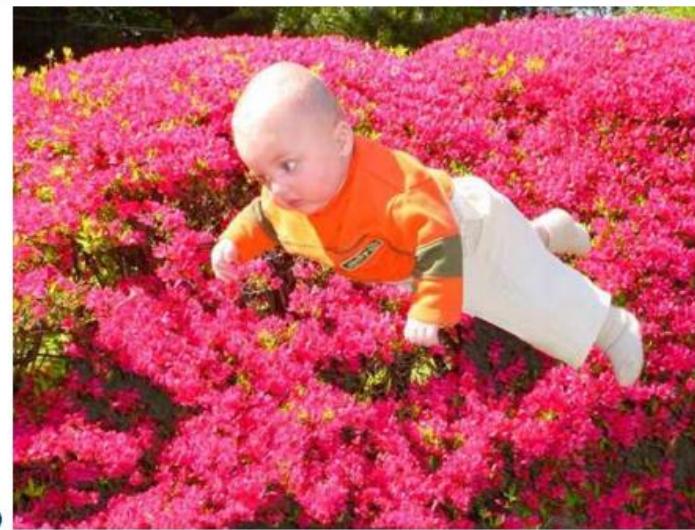


f

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？

带交互的补全算法

Image Completion with Structure Propagation

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

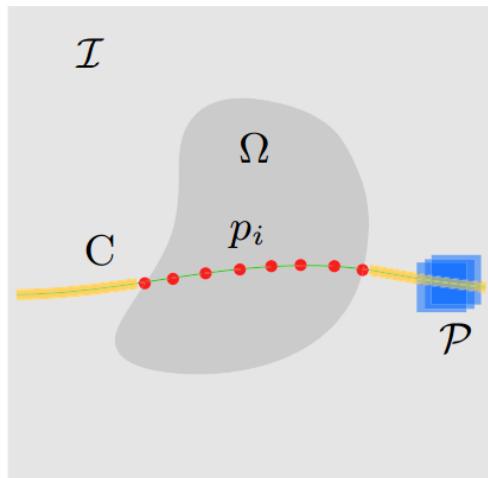
带有优先级的填充策略

算法概览



用户在空洞区域以及已知图像区域勾画结构线，
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

采样策略



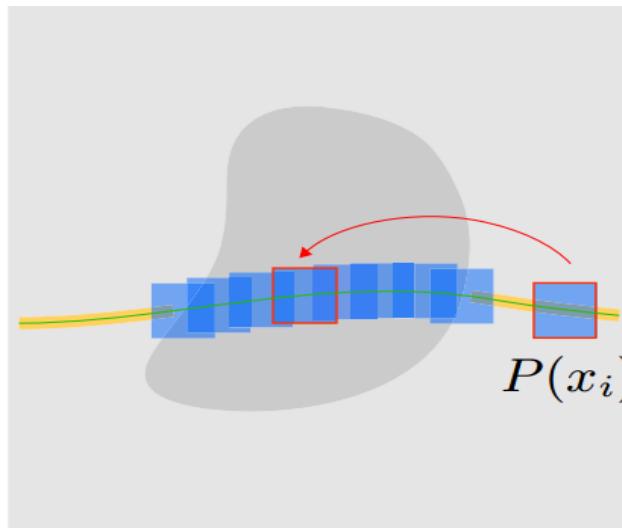
首先对处于 Ω 区域的 C 进行稀疏采样，依次得到 L 个锚点 $\{p_i\}_{i=1}^L$ 如左图所示，之后在 Ω 区域合成的块都是以这些锚点为中心的，这些锚点形成一条链或者可以认为是一个一维的图 $G = \{v, \varepsilon\}$ 。

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，一般采样间隔为 1-5 个像素，得到样本块集

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

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：

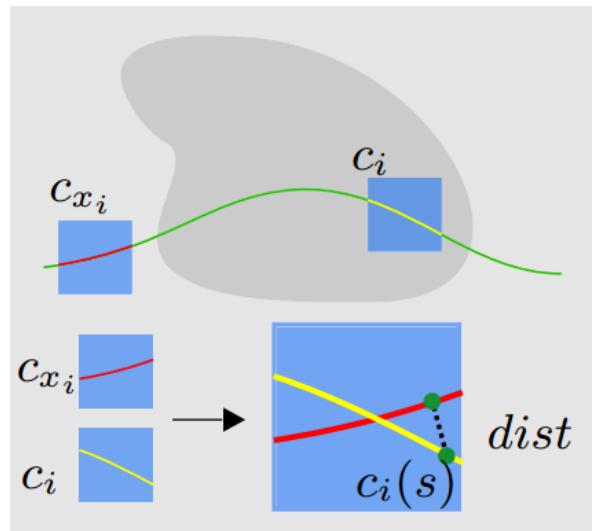
$$E(X) = \sum_{i \in v} E_1(x_i) + \sum_{(i, j) \in \varepsilon} E_2(x_i, x_j),$$

$$E_1(x_i) = k_s E_s(x_i) + k_I E_I(x_i).$$

$E_s(x_i)$, $E_I(x_i)$ 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

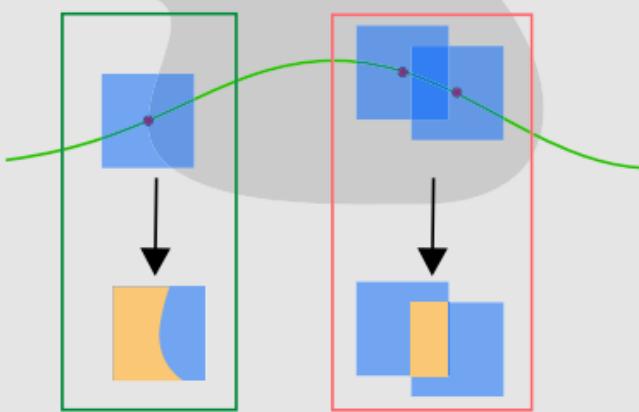


性。假设样本块 $P(x_i)$ 和定位于 p_i 的大小相同的矩形框中分别包含的曲线段 c_{x_i} 和 c_i ，如左图所示。

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

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。

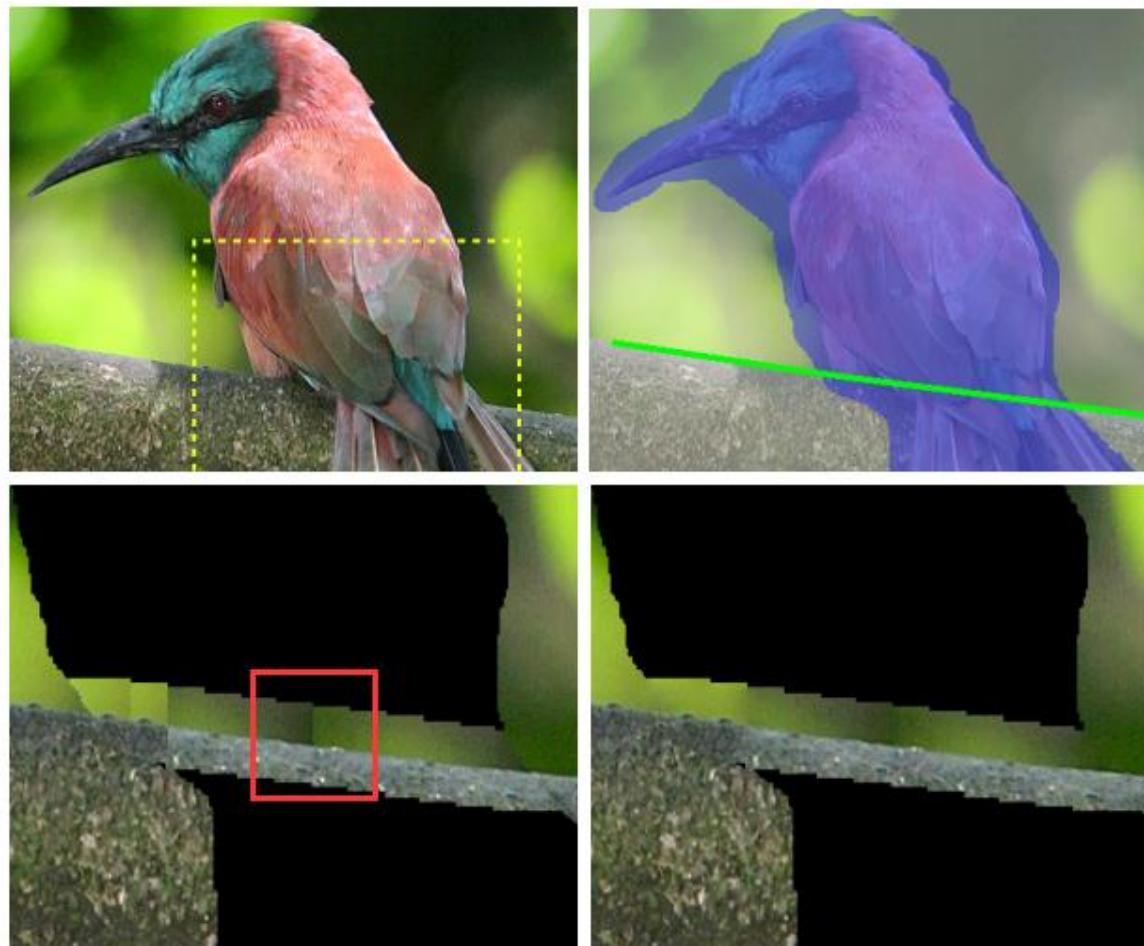
目标能量优化

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

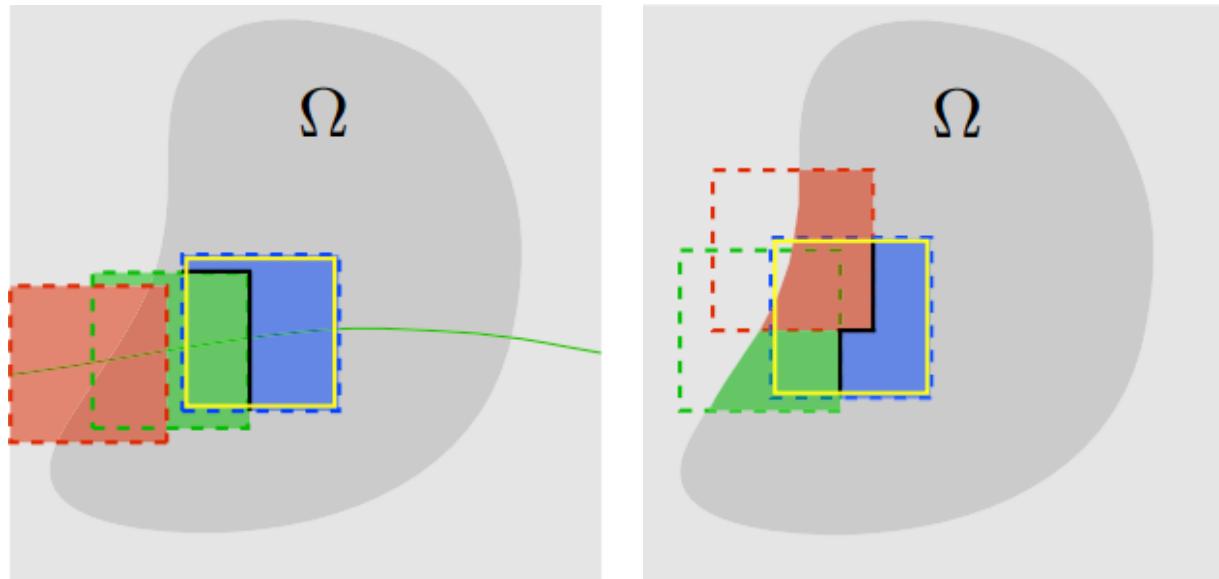
光测度修正

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，蓝色的块是将要合成的块。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。



修正前后对比

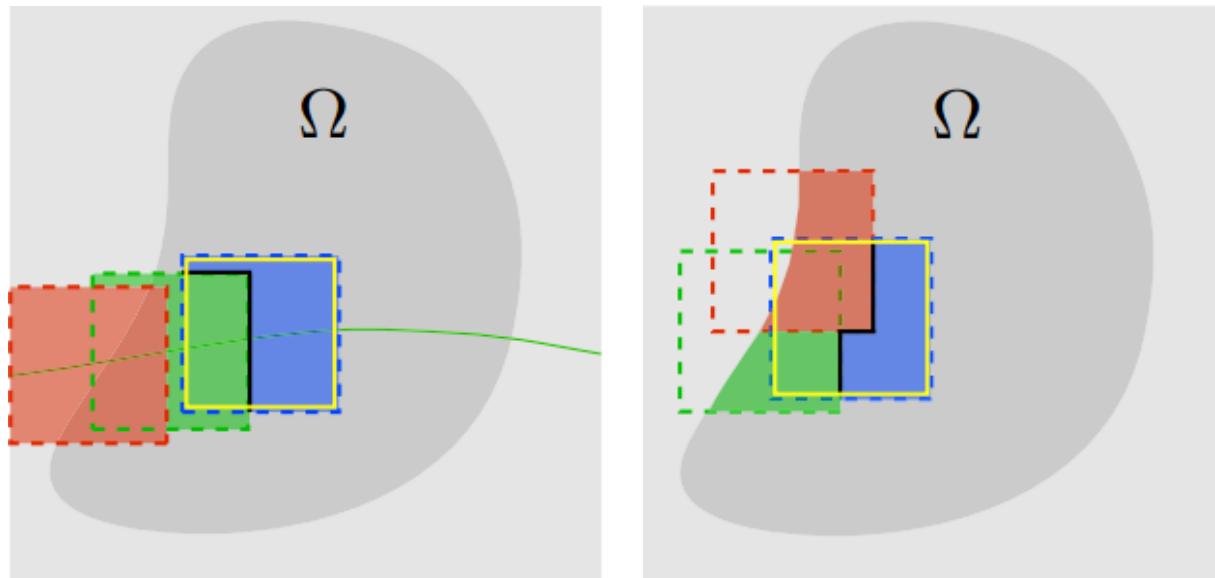


光测度修正

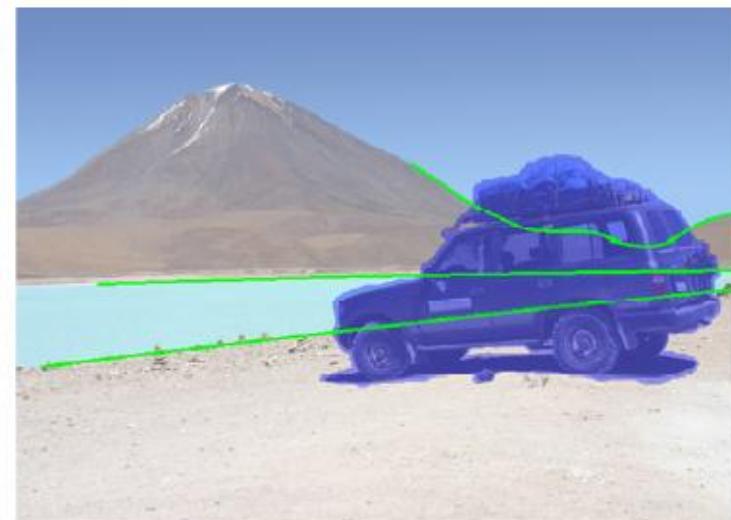
最后，我们通过修正 J 的梯度场 ∇J 重新构造一个新的块 J^* ，修正规则如下：

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

$\nabla J_y(x, y)^*$ 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

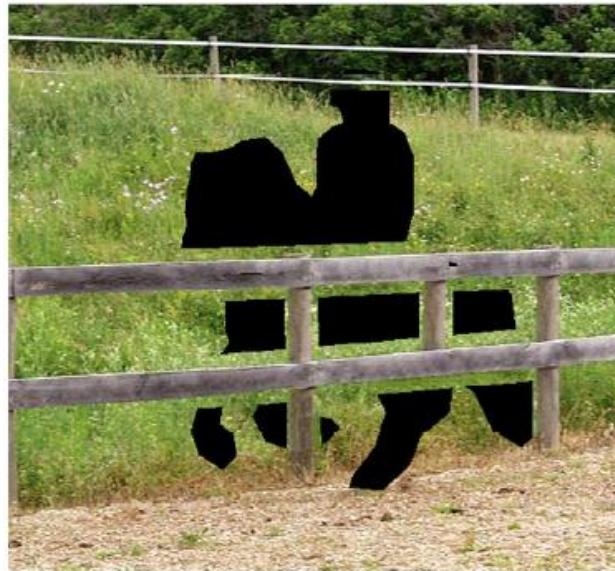
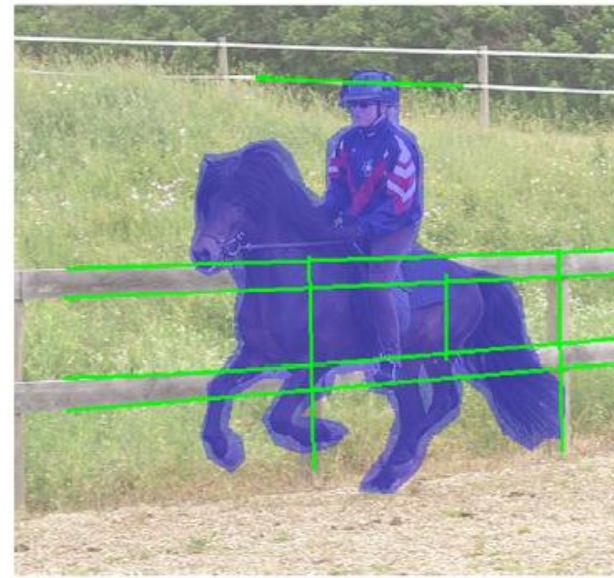


实验结果

Criminisi等人的方法



实验结果



实验结果

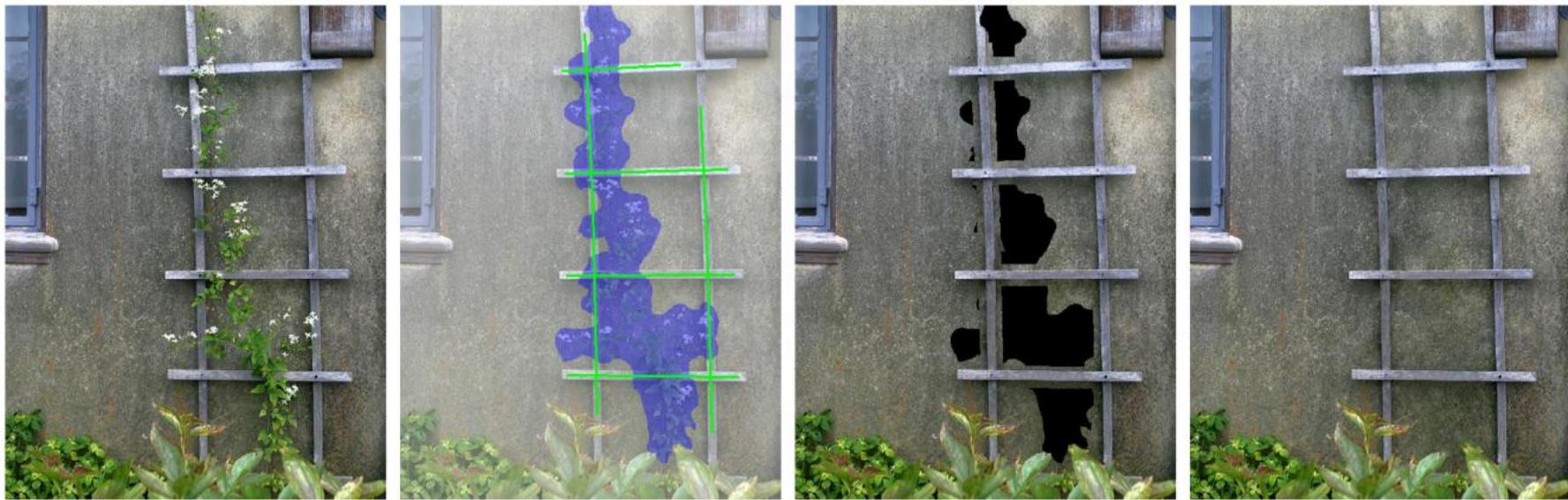
Criminisi等人的方法



实验结果



实验结果



局部优化算法

之前提到的自动方法都是按照某个顺序逐步的修复不可见区域的像素，因此都是一种局部方法，局部的方法容易产生一种结构性的后期修复的痕迹

局部优化算法



Space-time video completion

- Wexler, Y., Shechtman, E., Irani, M
- CVPR2004

一致性度量

Y.Wexler提出利用全局代价函数来求解图像补全问题，而这个代价函数在他的论文中被称之为一致性度量 (coherence measure)，假定S代表修复后的图像，T代表原始带有holes的图像

$$Coherence(S | T) = \sum_{p \in S} \min_{q \in T} s(W_p, W_q)$$

其中 p 为修复后图像中的像素点， q 为修复前原始图像中的像素点。W 代表包含像素的 Patch，s 代表两个 Patch 中的像素平方差和((sum of squared difference, SSD))。

能量优化

Y.Wexler采用EM(期望最大化)来求解该目标函数:

- a)对图像进行初始化修补(可以采用其他的修补算法)
- b)对修补区域的所有像素 p 的邻域块在已知区域中求得最相似的块
- c)根据块的相似性以及空间位置权重(即越靠近已知区域的patch获得更多的权重)对所有像素进行加权平均
- d)重复步骤2)和3)直到收敛

能量优化

在该算法中寻找最相似块是一个十分耗时的过程，但是Barnes 等人提出的PatchMatch算法极大的改善了这一情况

实验结果



实验结果



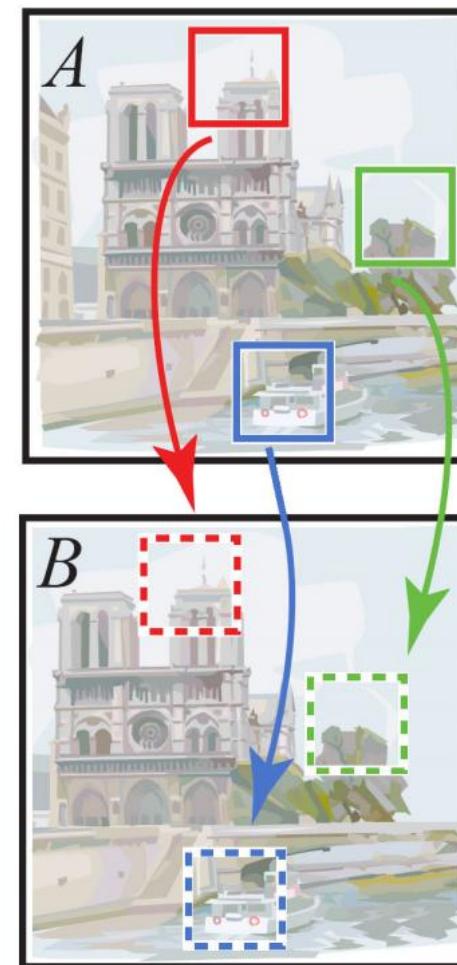
实验结果



PatchMatch: A randomized correspondence algorithm for structural image editing

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

最近邻域(Nearest-Neighbor Field)

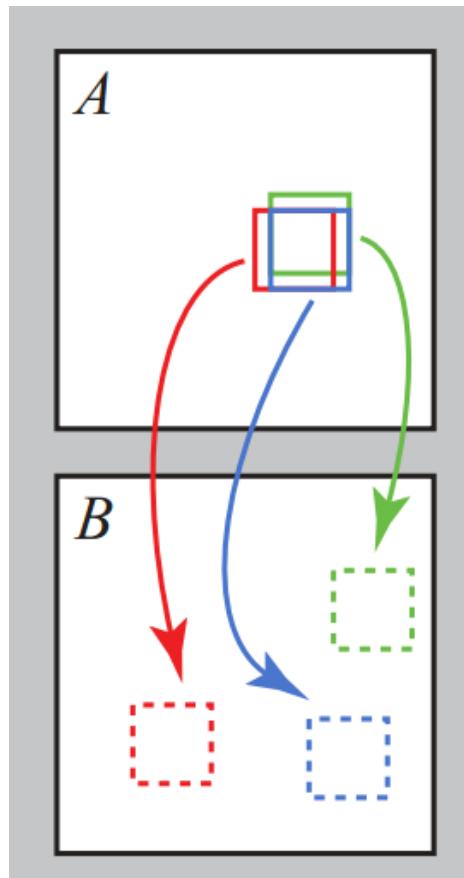


基本思想

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

1. 在图像范围内均匀采样



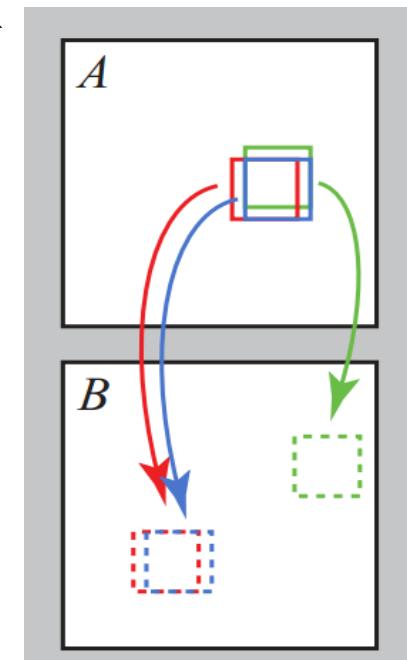
算法流程

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

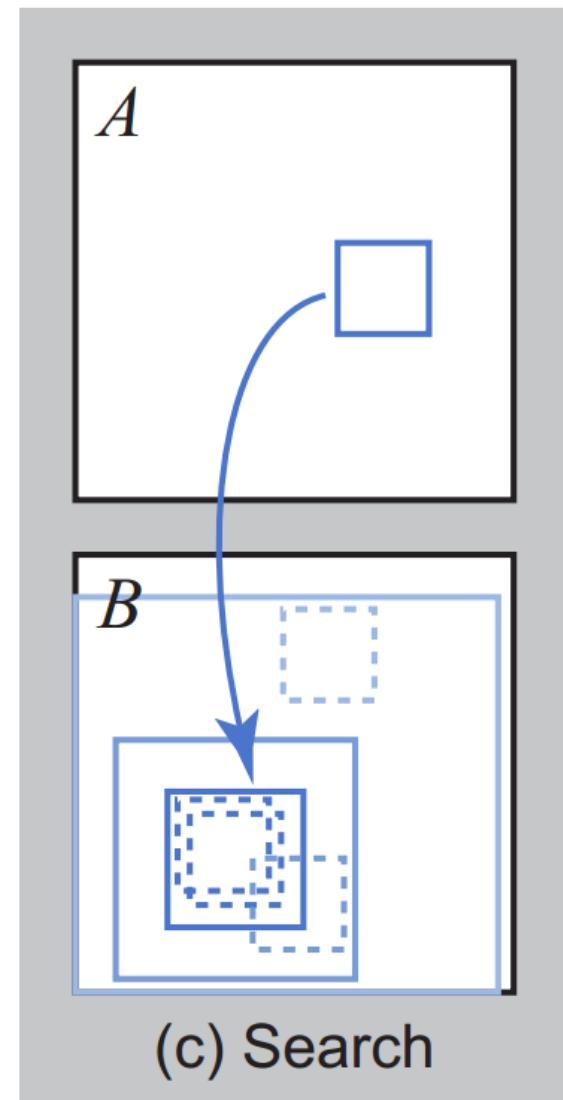


算法流程

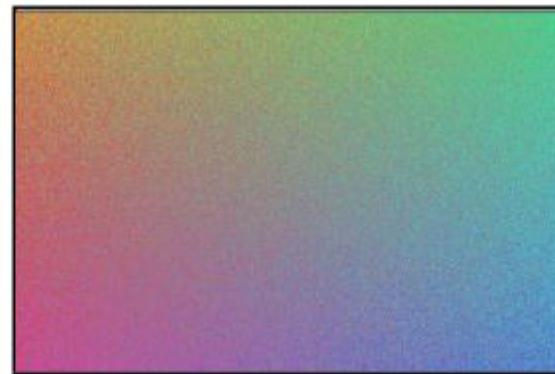
随机搜索过程Sn:

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

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, α 为收缩系数



实验结果



(a) originals

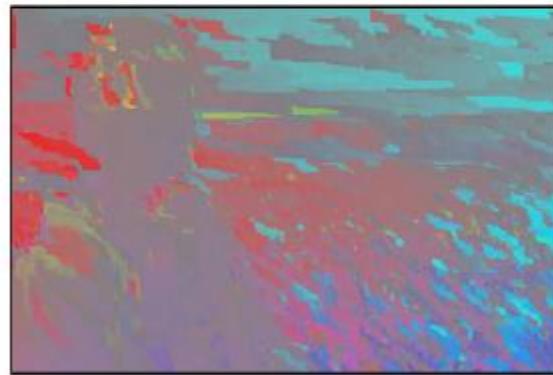
(b) random

(c) $\frac{1}{4}$ iteration

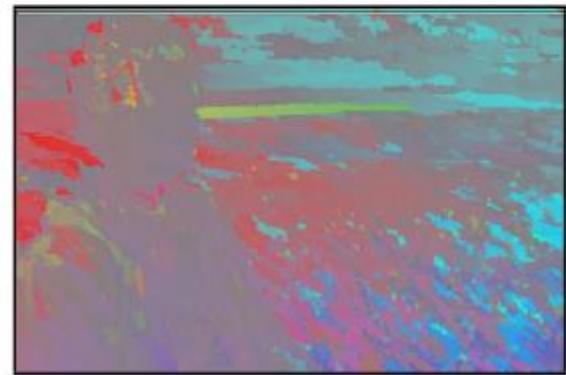
实验结果



(d) $\frac{3}{4}$ iteration



(e) 1 iteration



(f) 2 iterations



Image Melding: Combining Inconsistent Images using Patch- based Synthesis

- Darabi, S., Shechtman, E.,
Barnes, C., Goldman, D.B., Sen,
P
- SIGGRAPH2012

算法概览

ImageMelding 主要从三个方面对Y.Wexler等人的算法进行了优化：

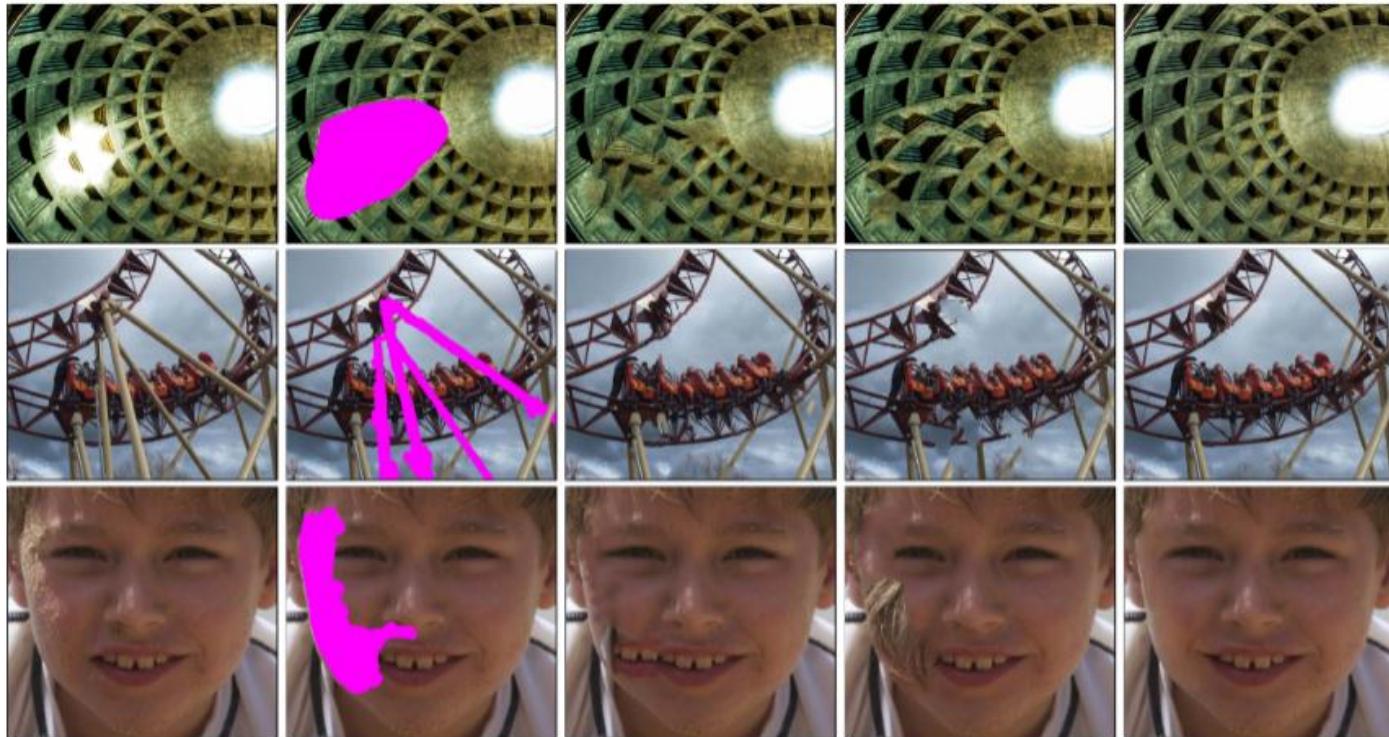
1. 在一致性度量中添加梯度相似性度量
2. 扩充样本块的搜索空间，包括平移，旋转，非均匀缩放，镜面反射，为每个通道的颜色添加一个偏移和增幅
3. 利用梯度和颜色信息重新生成新的图像

目标能量

$$E(T, S) = \sum_{q \in T} \min_{p \in S} (D(Q, P) + \lambda D(\nabla Q, \nabla P))$$

其中 T 代表目标区域即待修复区域, S 为原图像中已知区域, $Q = N(q)$, $N(q)$ 代表以 q 为左上角的大小为 w 的 patch, $P = f(N(p))$, f 代表一种几何和光测度转换, 最后将 $N(p)$ 变换为一个大小为 w 的块 P 。 $\nabla Q = (\nabla_x L, \nabla_y L)$, 代表图像在 Lab 颜色空间中 L 通道在 x , y 方向的梯度。 λ 为一个常量, 代表一个权重。由于该算法扩充了样本块的搜索空间, 相对 Y.Wexler 等人的算法能够得到更好的补全质量, 但是它需要更多的时间来求解目标函数。

实验结果



从左到右依次为 原图像, mask 图像, Y.Wexler 等人算法修复结果, shift-map 结果 Soheil Darabi 等人算法结果

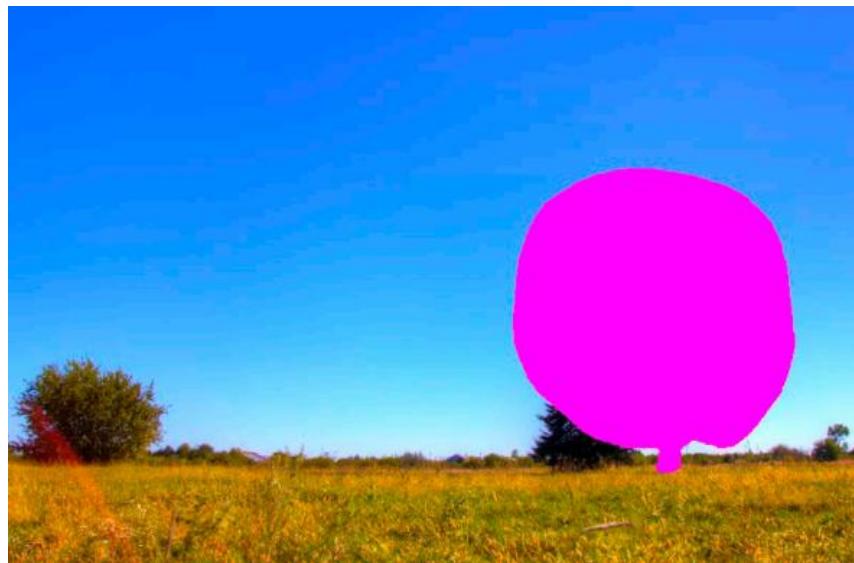
实验结果



实验结果



实验结果



Shift-map image editing

- Pritch, Y., Kav-Venaki, E., Peleg, S
- ICCV2009

算法概览

输入图像 $I(x, y)$ 和输出图像 $R(u, v)$ 的关系是由一个偏移图 $M(u, v) = (t_x, t_y)$ 来决定的，输出像素 $R(u, v)$ 由输入像素 $I(u + t_x, v + t_y)$ 来得到。最优的偏移图问题被构造图标记问题，其中节点(site)为输出图像的像素，每一个输出像素被一个偏移量 (t_x, t_y) 标记。最优的偏移图 M 将最小化下面的能量函数：

$$E(M) = \alpha \sum_{p \in R} E_d(M(p)) + \sum_{(p,q) \in N} E_s(M(p), M(q)),$$

Shift-map算法应用于图像补全

我们定义需要修复的图像为 I ，待修复的区域为 Ω ，注意其中 Ω 是带边界的 $\partial\Omega$ ，

$$E_d(M(p)) = \begin{cases} 0, & \text{if } p \in \Omega \text{ and } p + M(p) \in I - \Omega \\ 0, & \text{if } p \in \partial\Omega \text{ and } M(p) = (0, 0) \\ +\infty, & \text{if } p \in \Omega \text{ and } p + M(p) \notin I - \Omega \end{cases}$$

E_d 项约束了修补的像素一定要来自于 I 中的已知区域，且边界上的像素一定是不能移动的。 ↵

Shift-map算法应用于图像补全

E_s 项同时考虑了输出图像与输入图像对于空间邻域的色差和梯度差特征：

$$E_s(M) = \sum_{(u,v) \in R} \sum_i (R((u,v) + e_i) - I((u,v) + M(u,v) + e_i))^2 + \\ \beta \sum_{(u,v) \in R} \sum_i (\nabla R((u,v) + e_i) - \nabla I((u,v) + M(u,v) + e_i))^2$$

其中 e_i 是四个方向的单位向量代表像素的四个邻域，由于同时考虑了梯度差，所以结构特征能够比较好的保留。

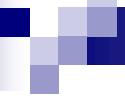
实验结果



本文结果



Criminisi等人
算法的结果



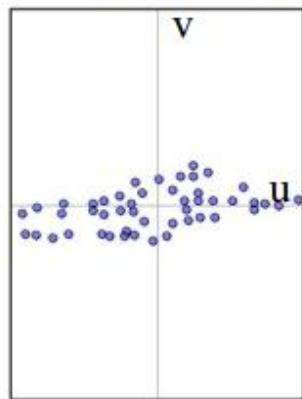
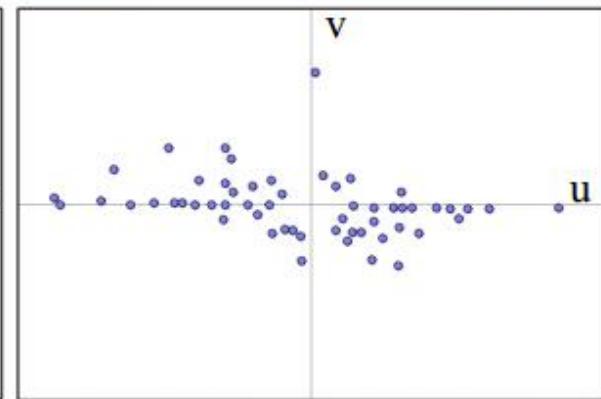
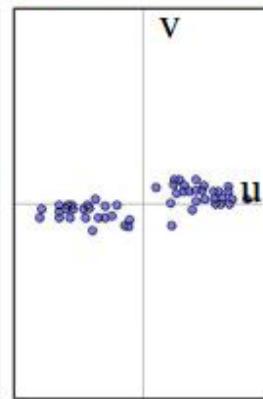
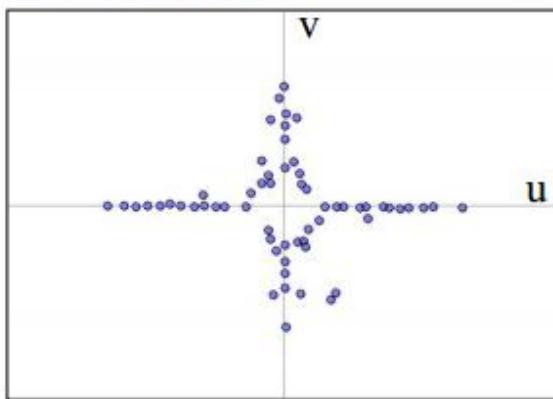
Statistics of patch offsets for image completion

- Kaiming He , Jian Sun
- ECCV2012

算法概览

何恺明等人对朴素的**shift-map**进行了改进，他们认为偏移量表达的是原始图像中的一种结构重复信息，可以利用事先统计的一些代表原图结构重复信息的主偏移量(**dominant offset**)作为多标签问题的候选解

主偏移量



实验结果

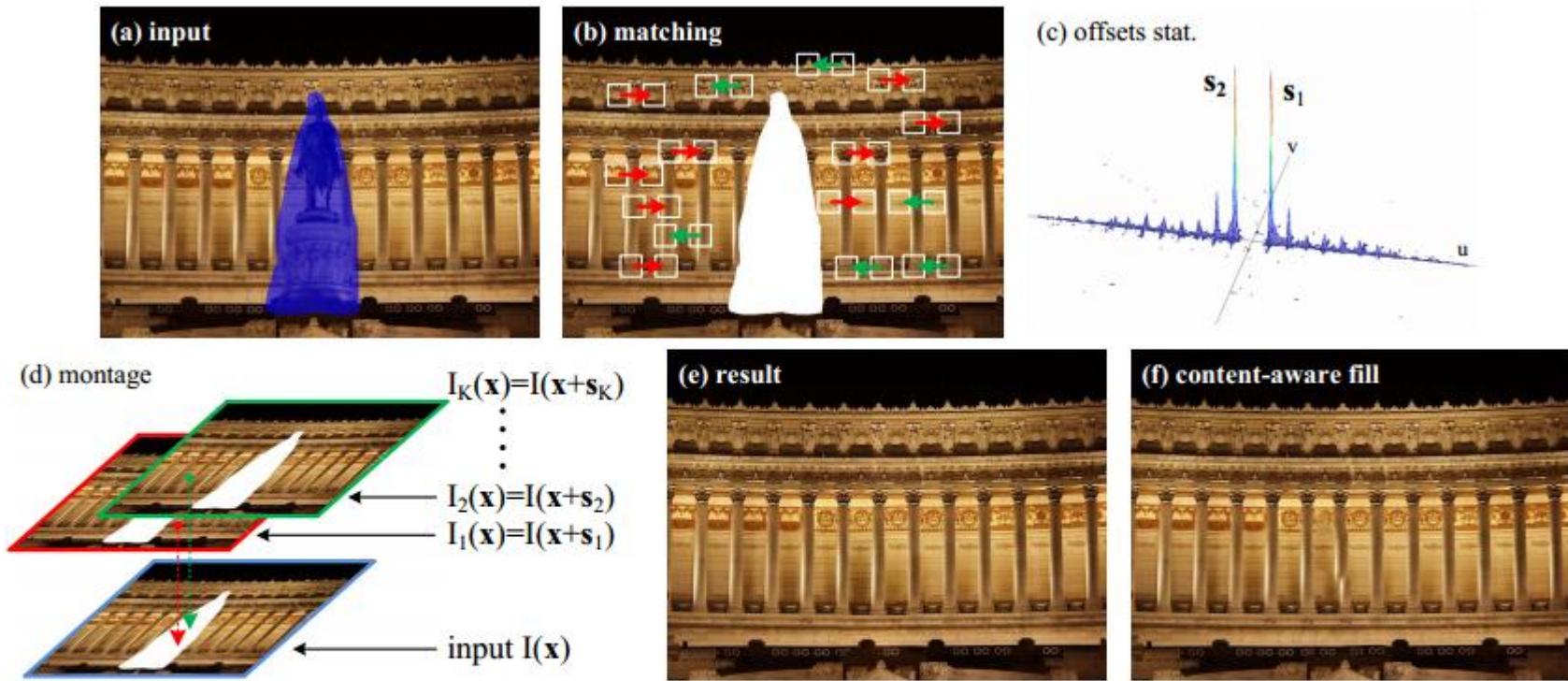
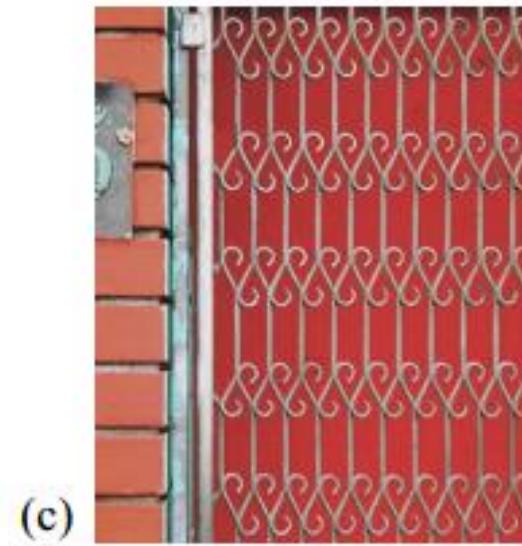
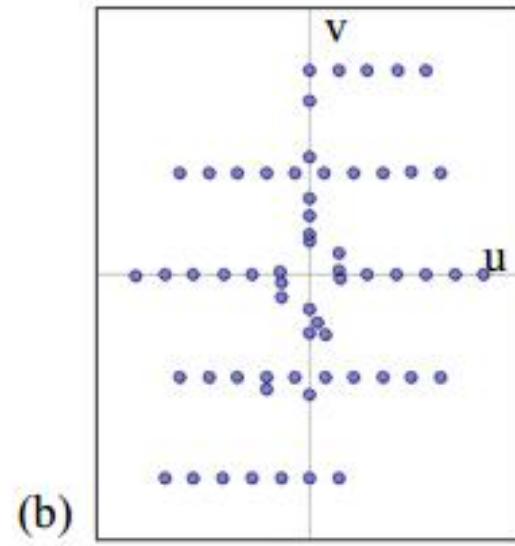
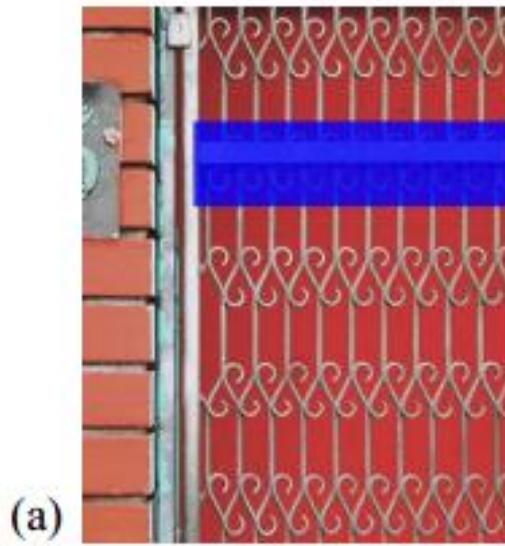
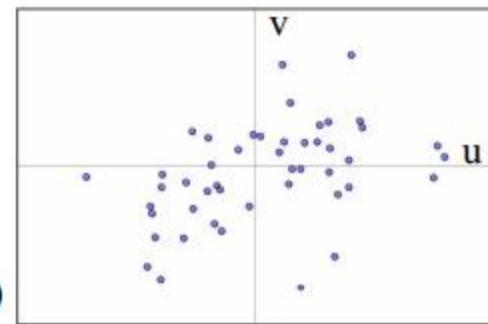


Fig. 1. Algorithm outline. (a) Input image with a mask overlayed. (b) Matching similar patches in the known region. (c) The statistics of the offsets of the similar patches. The offsets corresponding to the highest peaks are picked out. (d) Combining a set of shifted images with the given offsets. (e) Our result. (f) Result of Content-Aware Fill.

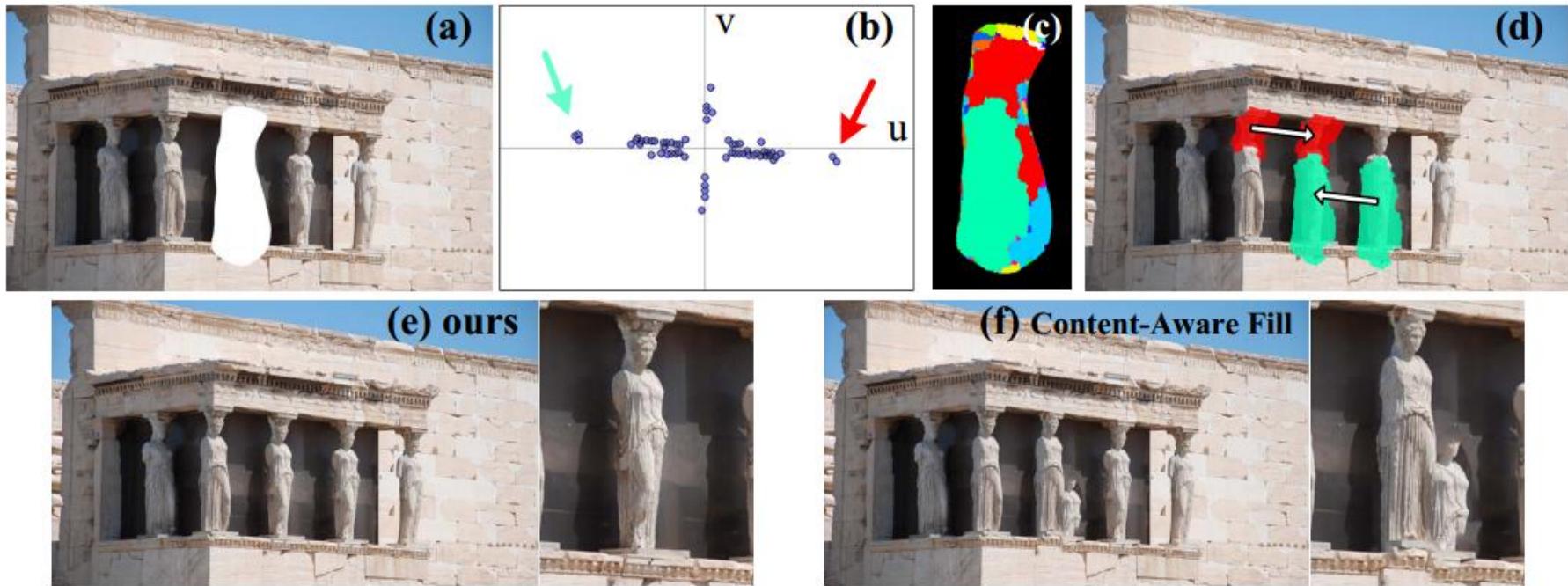
实验结果

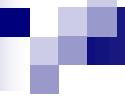


实验结果



实验结果





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



Images

Showing: All image sizes ▾

Results 1 - 20 of about 908,000 for **alley** [definition] with Safesearch on. (0.07 seconds)

Change Alley Aerial Plaza with its The Printer's Alley sign looking ... Looking west past Printers Alley.

679 x 450 - 469k - jpg

franklin.thefuntimesguide.com

679 x 450 - 464k - jpg

franklin.thefuntimesguide.com

More Bubble Gum Alley photos can be ...

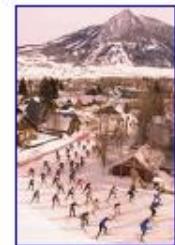
764 x 591 - 33k - gif

www.locallinks.com

Gasoline Alley gang

692 x 430 - 177k - jpg

newcritics.com



2007 Alley Loop Sponsors

300 x 453 - 51k - jpg

www.cbnordic.org



Change Alley : interior

550 x 413 - 98k

infopedia.nlb.gov.sg



Earl G. Alley ...

321 x 383 - 19k - jpg

www.msstate.edu



Gun Alley 8.5x11 Full Color Ink

Wash ...

390 x 301 - 14k - jpg

www.rorschachentertainment.com



Grace Court Alley

732 x 549 - 98k - jpg

www.bridgeandtunnelclub.com



Grace Court Alley

732 x 549 - 80k - jpg

www.bridgeandtunnelclub.com

panoramic photo of Alligator Alley
4902 x 460 - 1048k - jpg
sflwww.er.usgs.gov



Richard B. Alley

450 x 361 - 29k - gif

www.ncdc.noaa.gov



Also, Chicken Alley is reported to ...
450 x 337 - 82k
phidoux.typepad.com



Ego Alley

500 x 375 - 48k - jpg

dc.about.com

Data

2.3 Million unique images from Flickr groups and keyword searches.

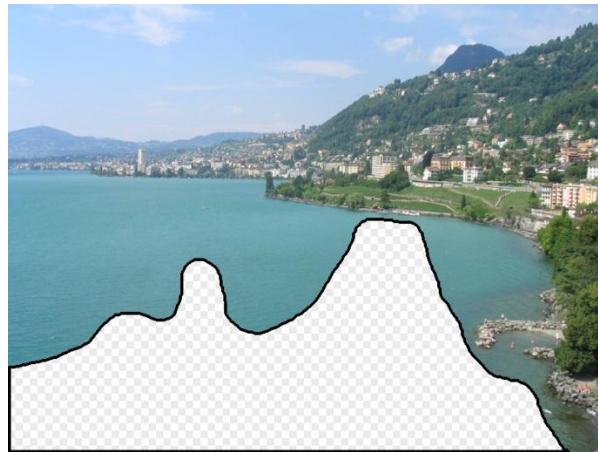




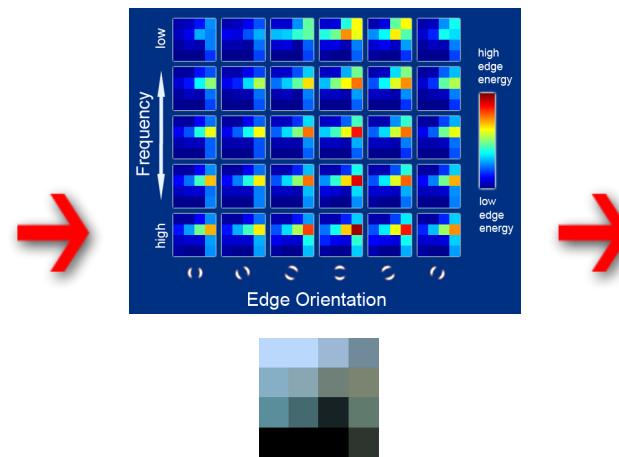


Scene Completion Result

The Algorithm



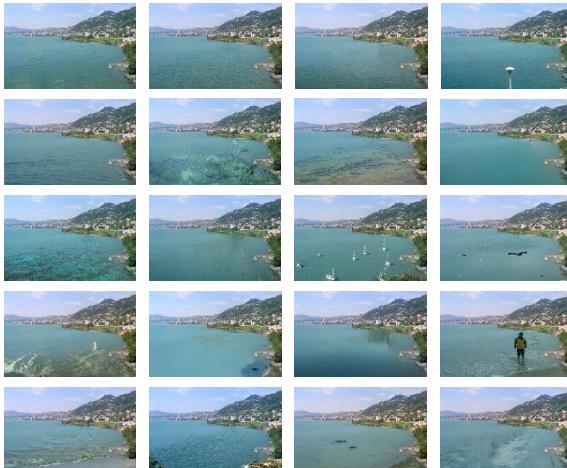
Input image



Scene Descriptor



Image Collection



20 completions

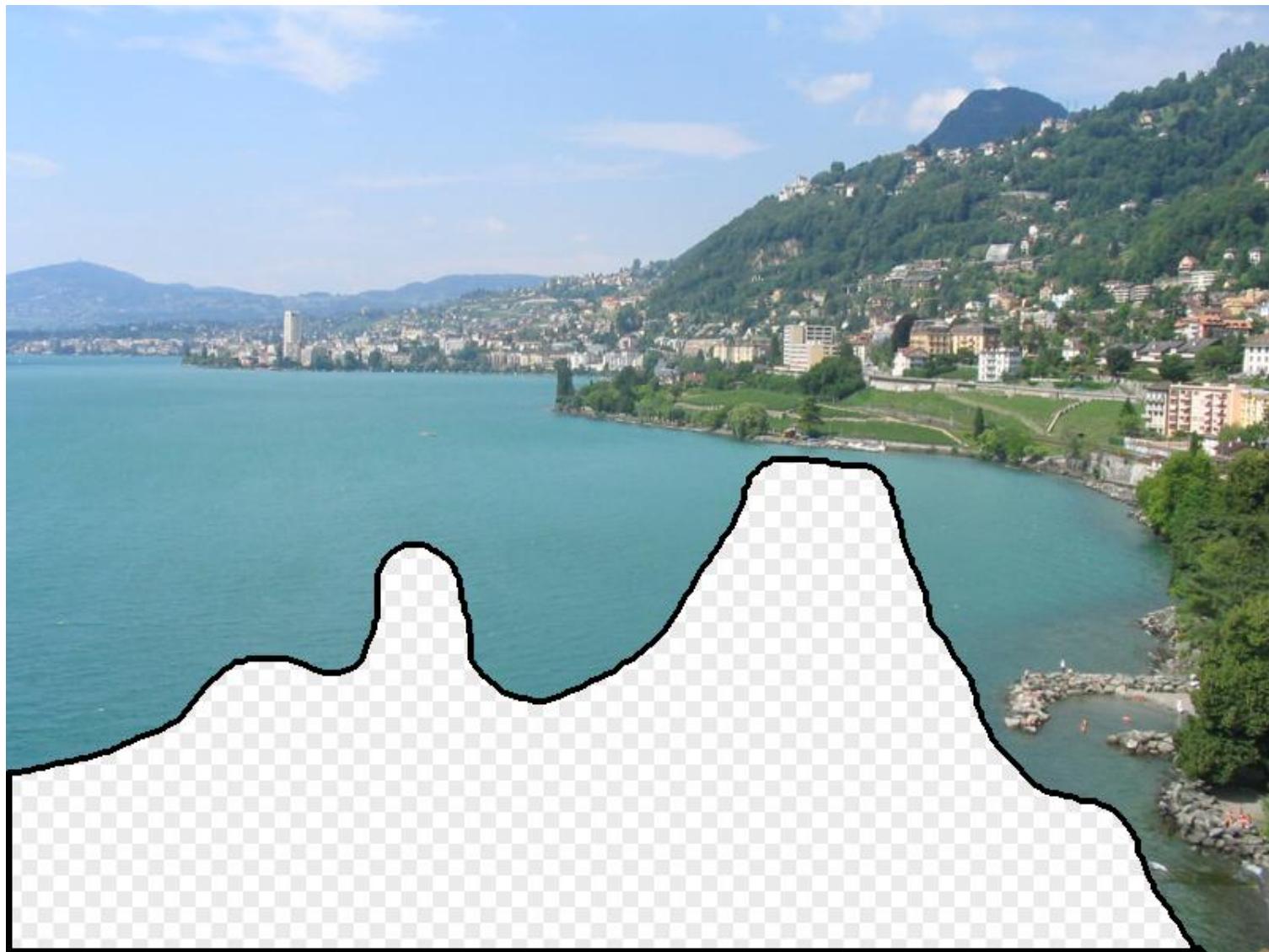


Context matching
+ blending

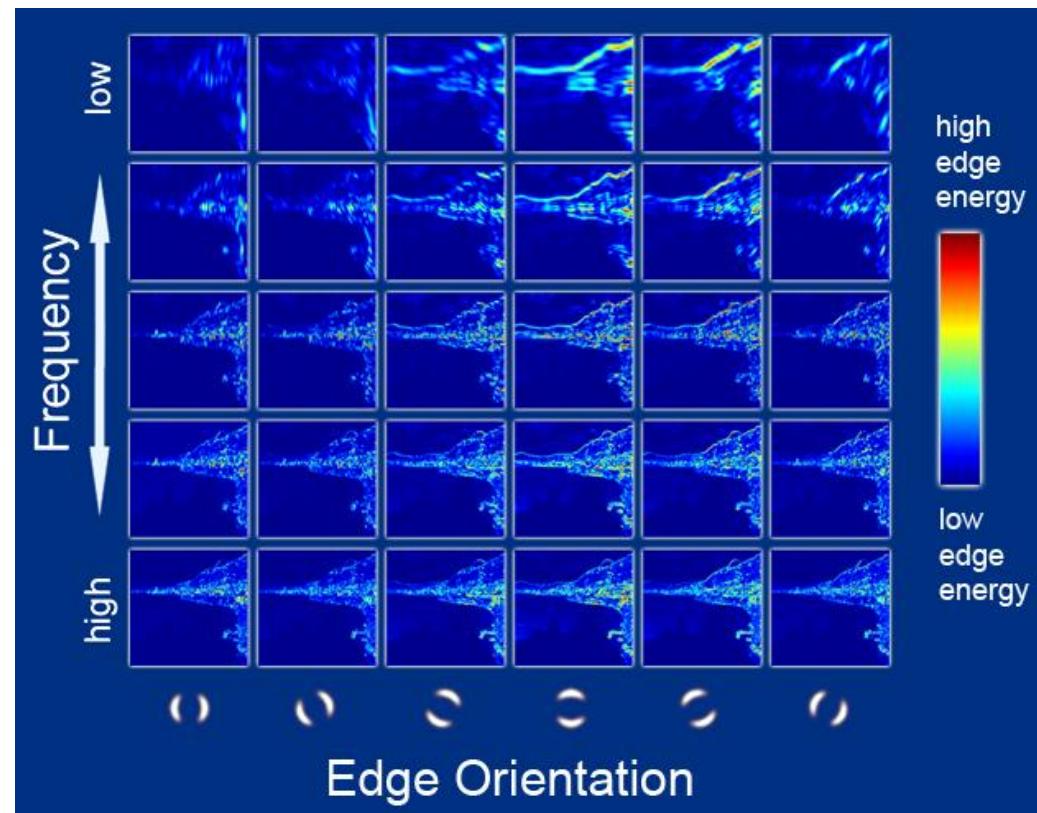
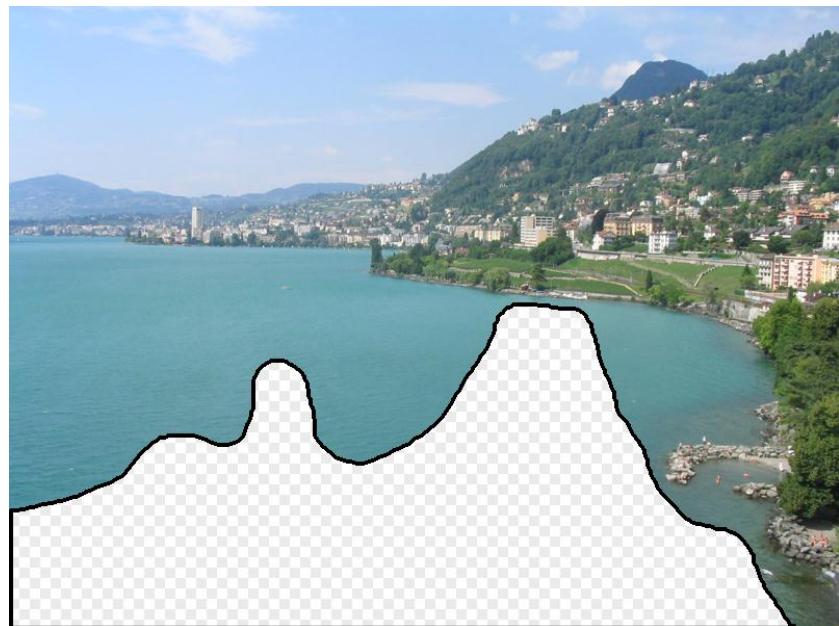


200 matches

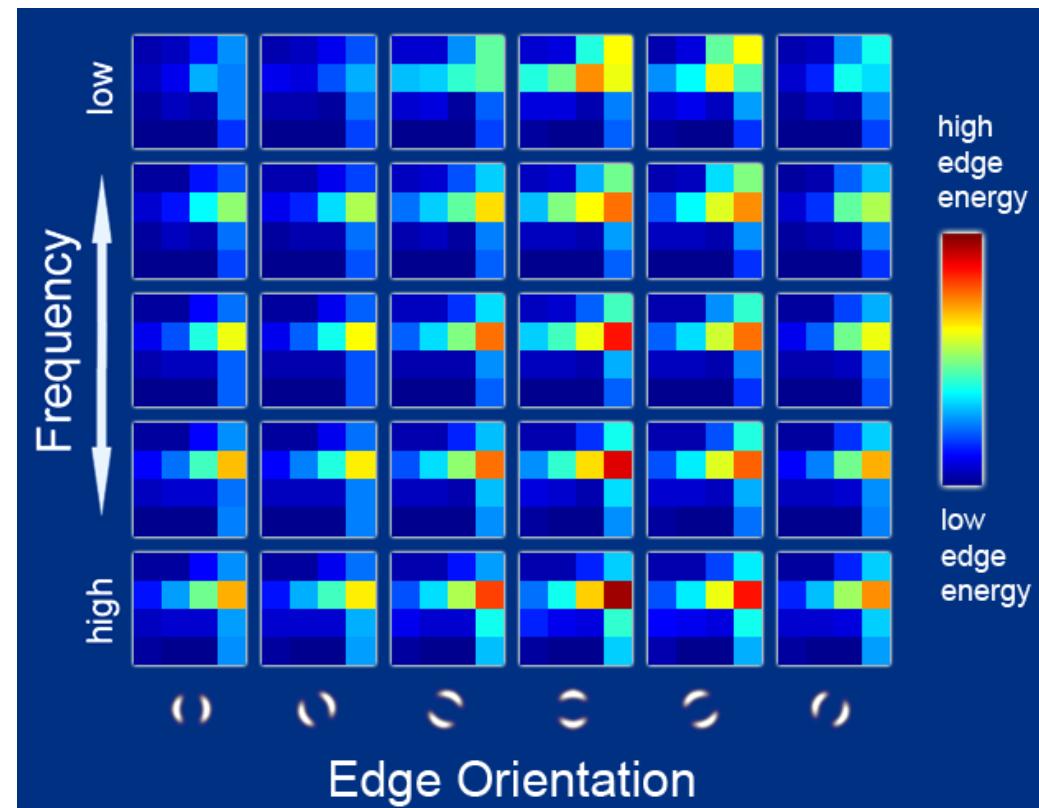
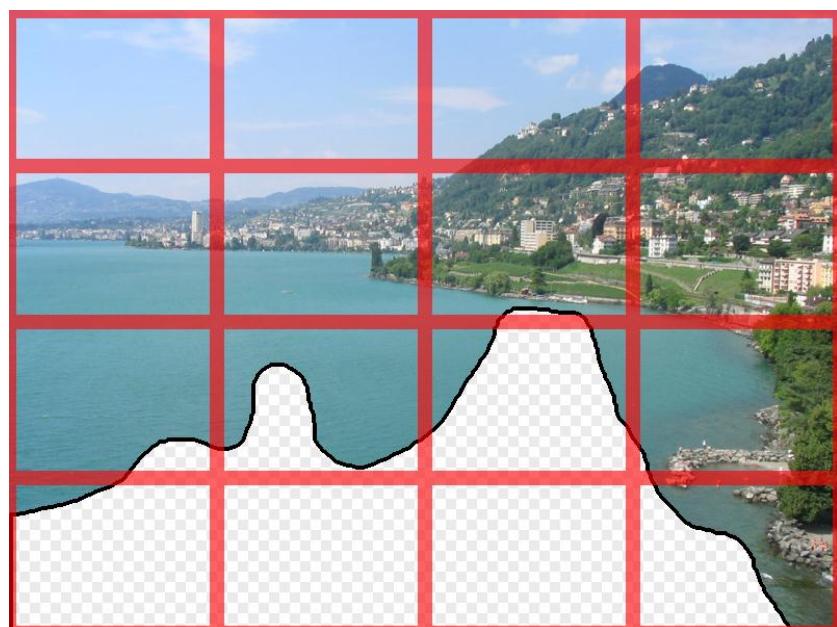
Scene Matching



Scene Descriptor

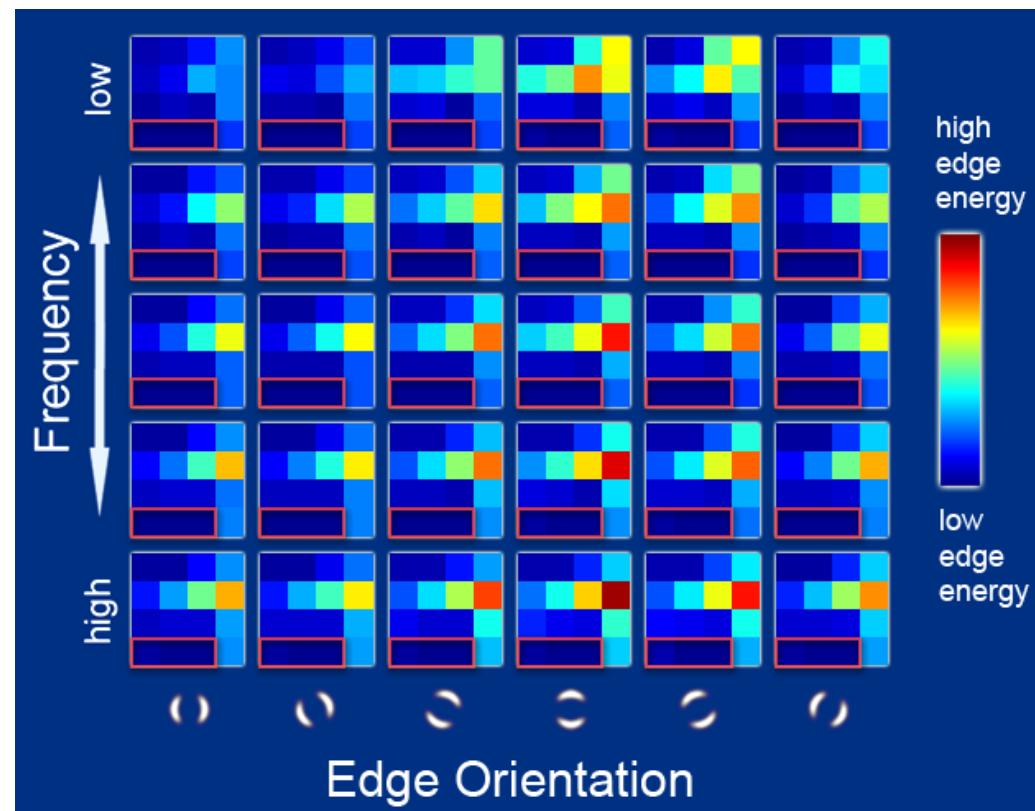
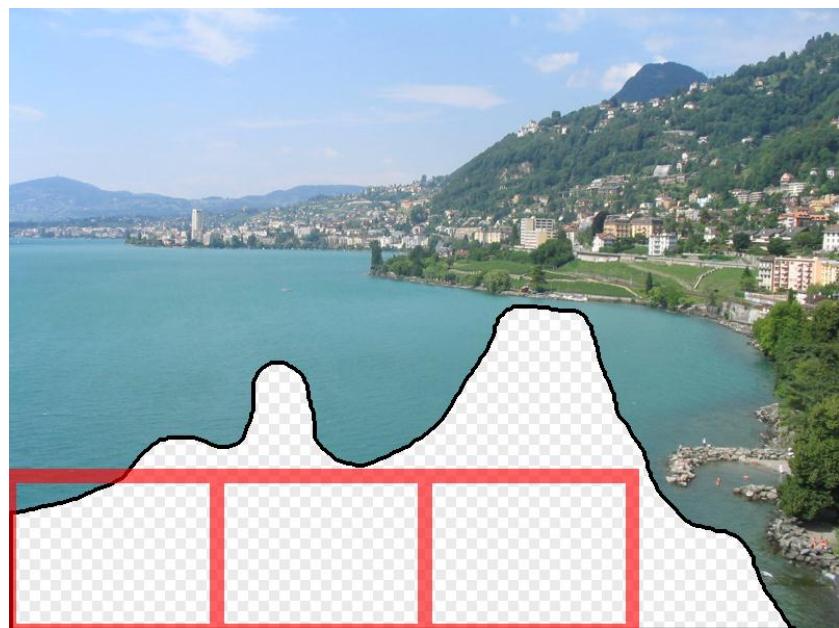


Scene Descriptor



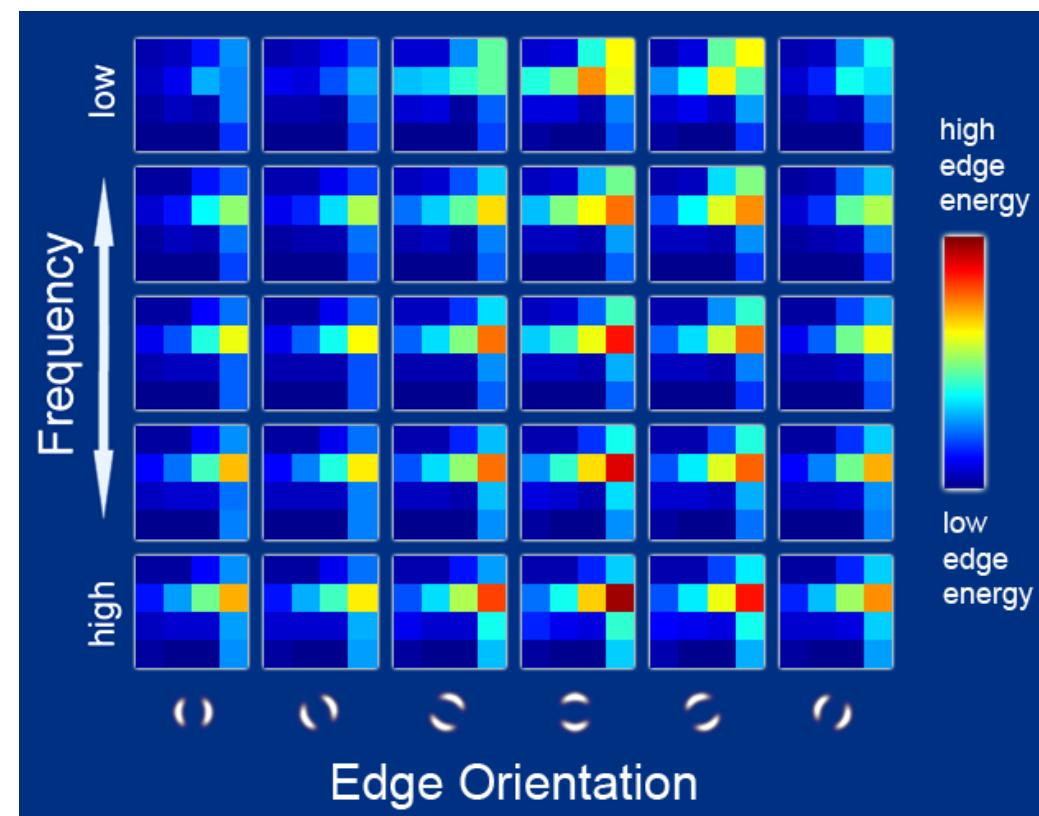
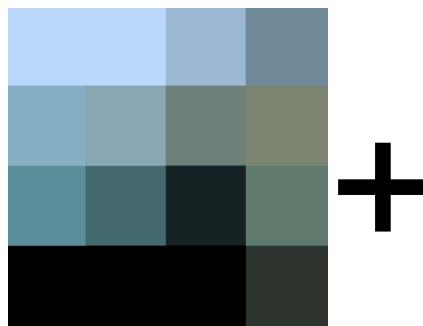
Gist scene descriptor
(Oliva and Torralba 2001)

Scene Descriptor



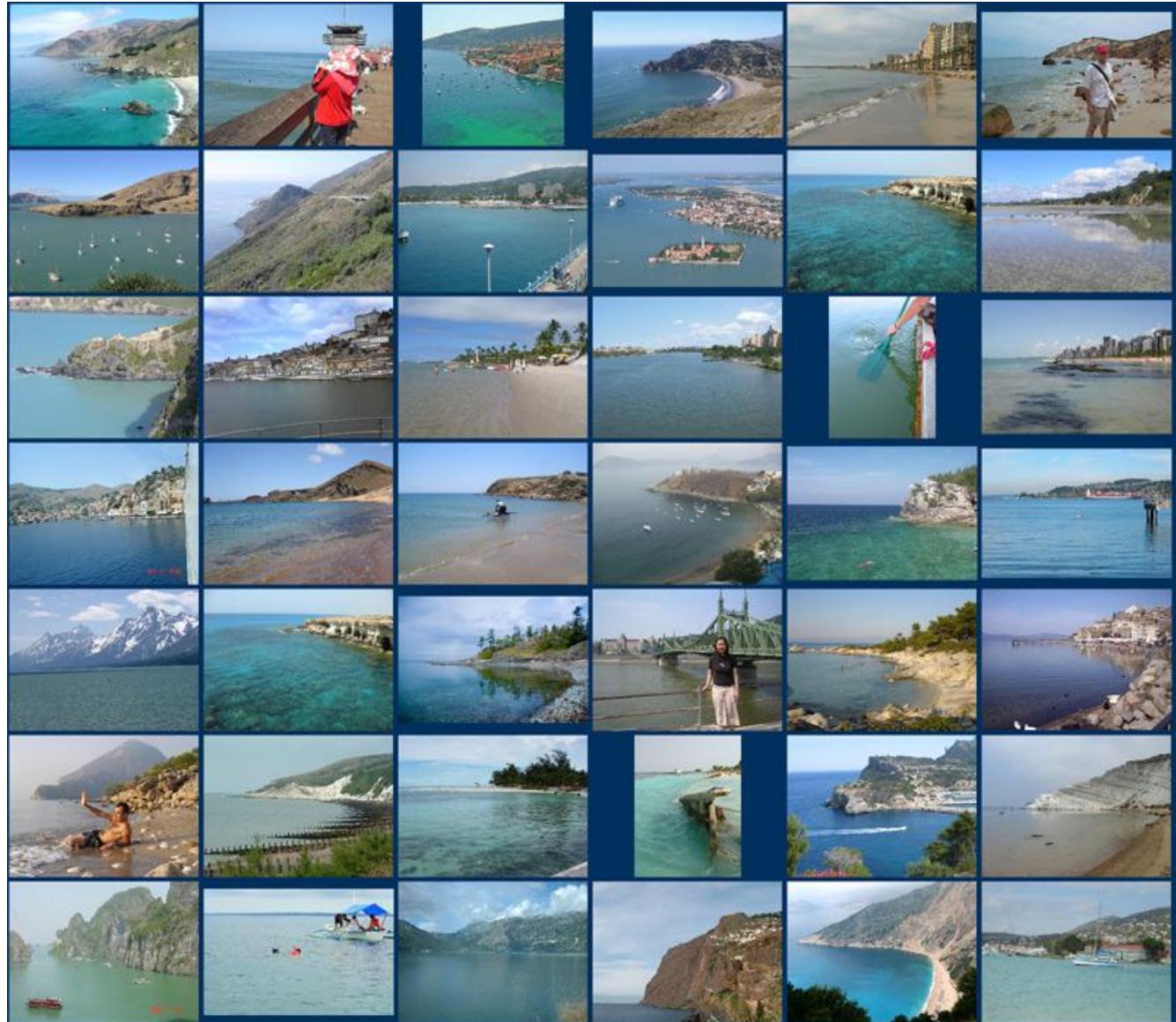
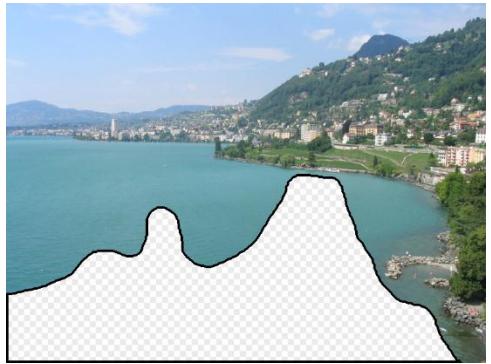
Gist scene descriptor
(Oliva and Torralba 2001)

Scene Descriptor



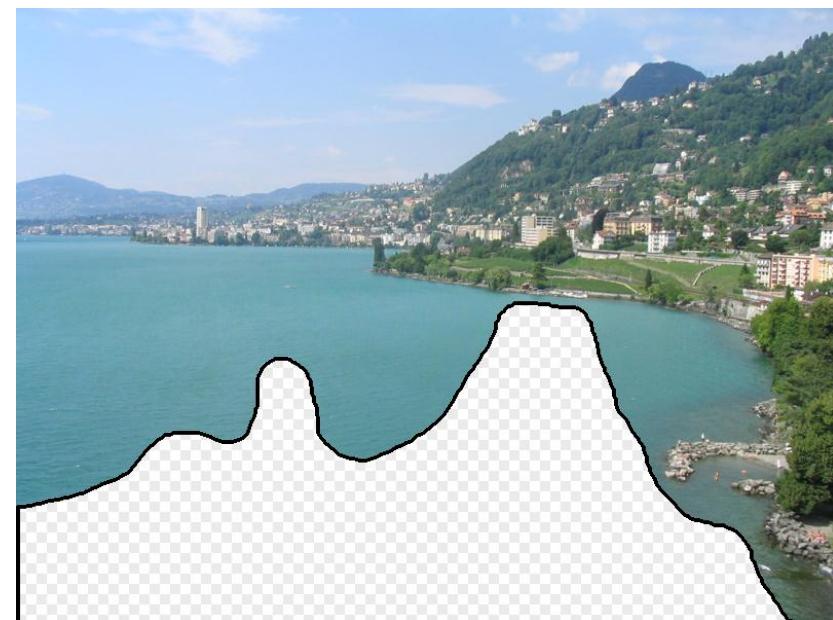
Gist scene descriptor
(Oliva and Torralba 2001)

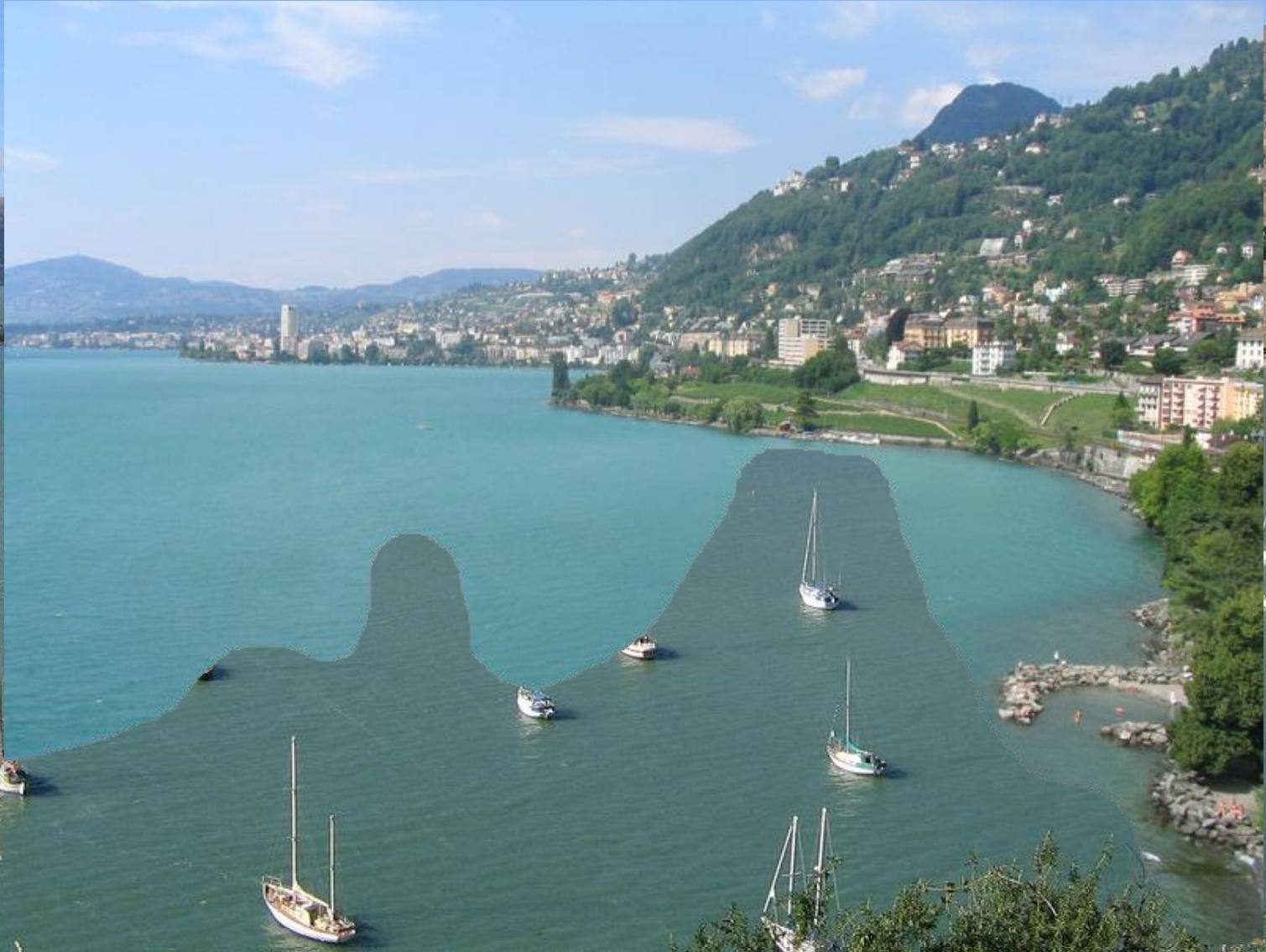




... 200 total

Context Matching

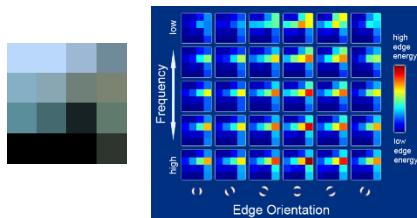






Result Ranking

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



The scene matching distance

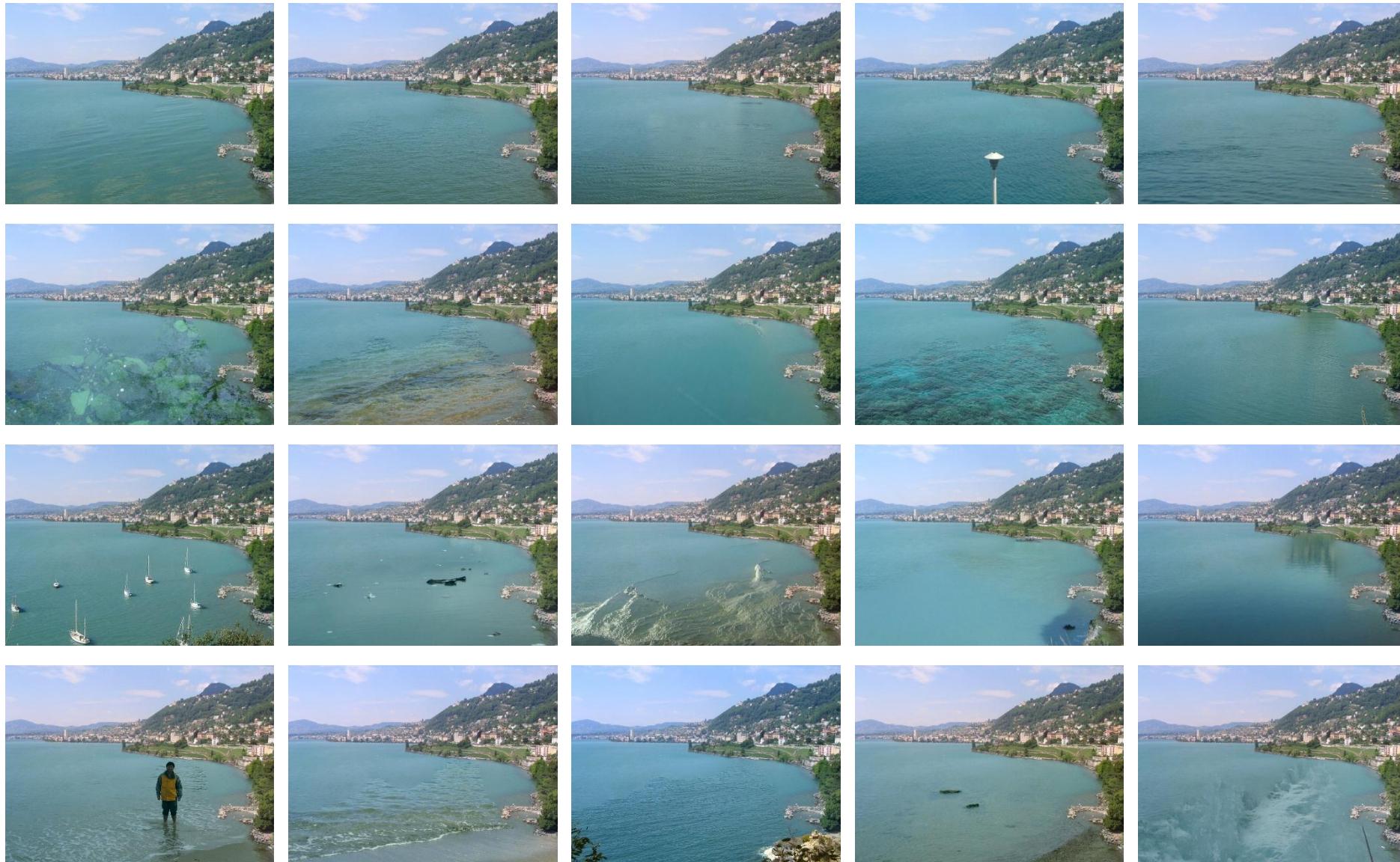


The context matching distance
(color + texture)



The graph cut cost

Top 20 Results







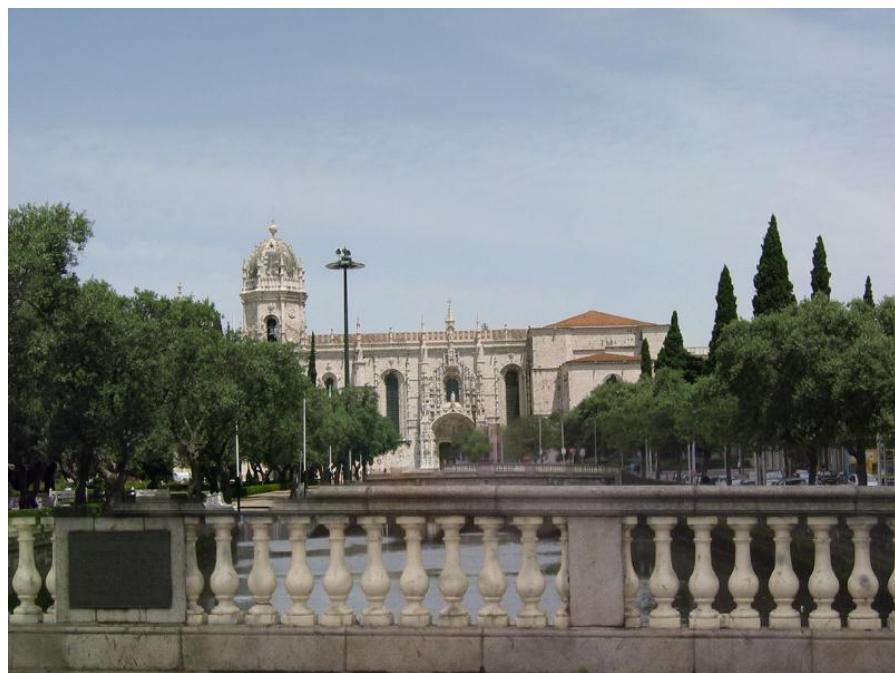








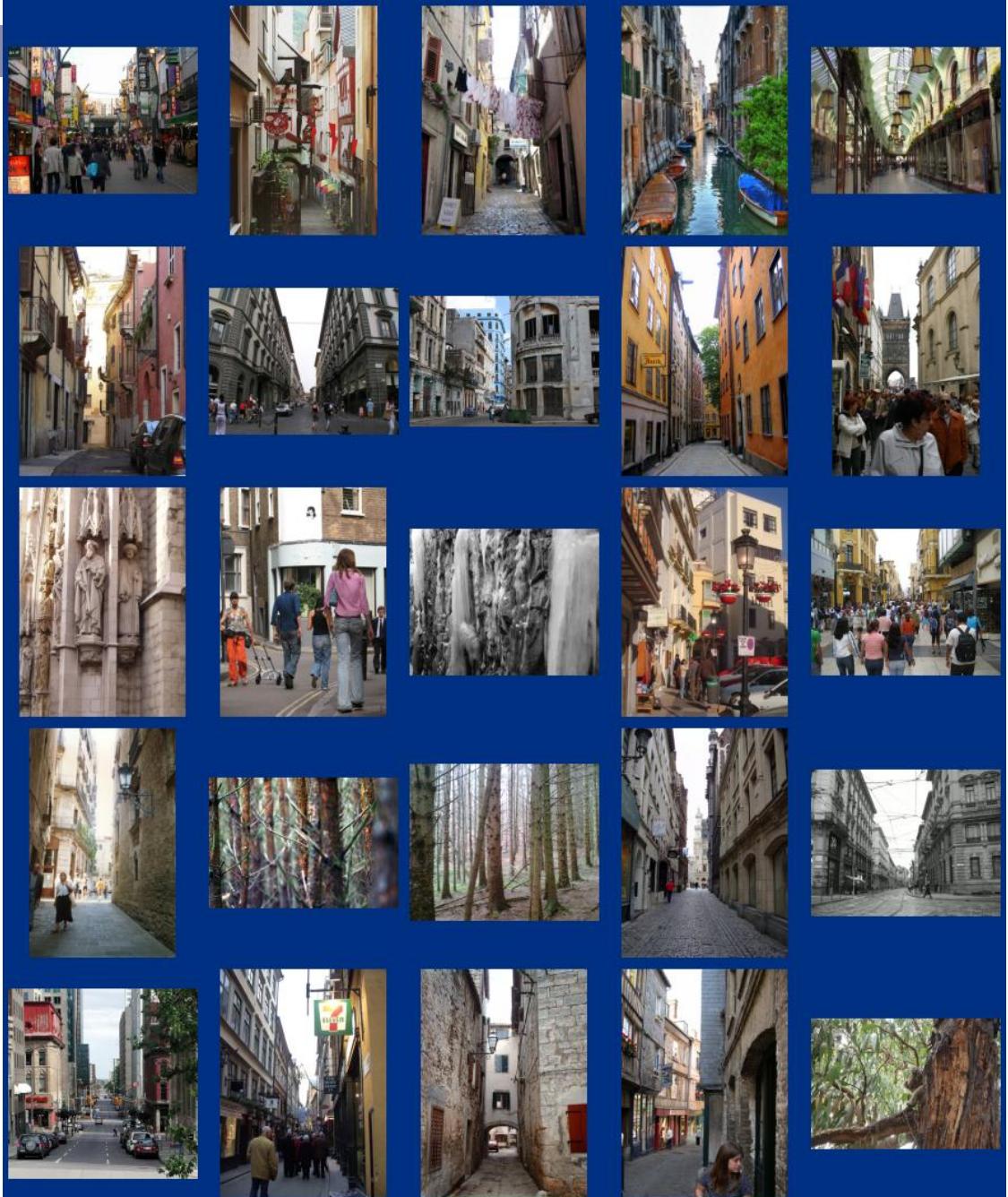




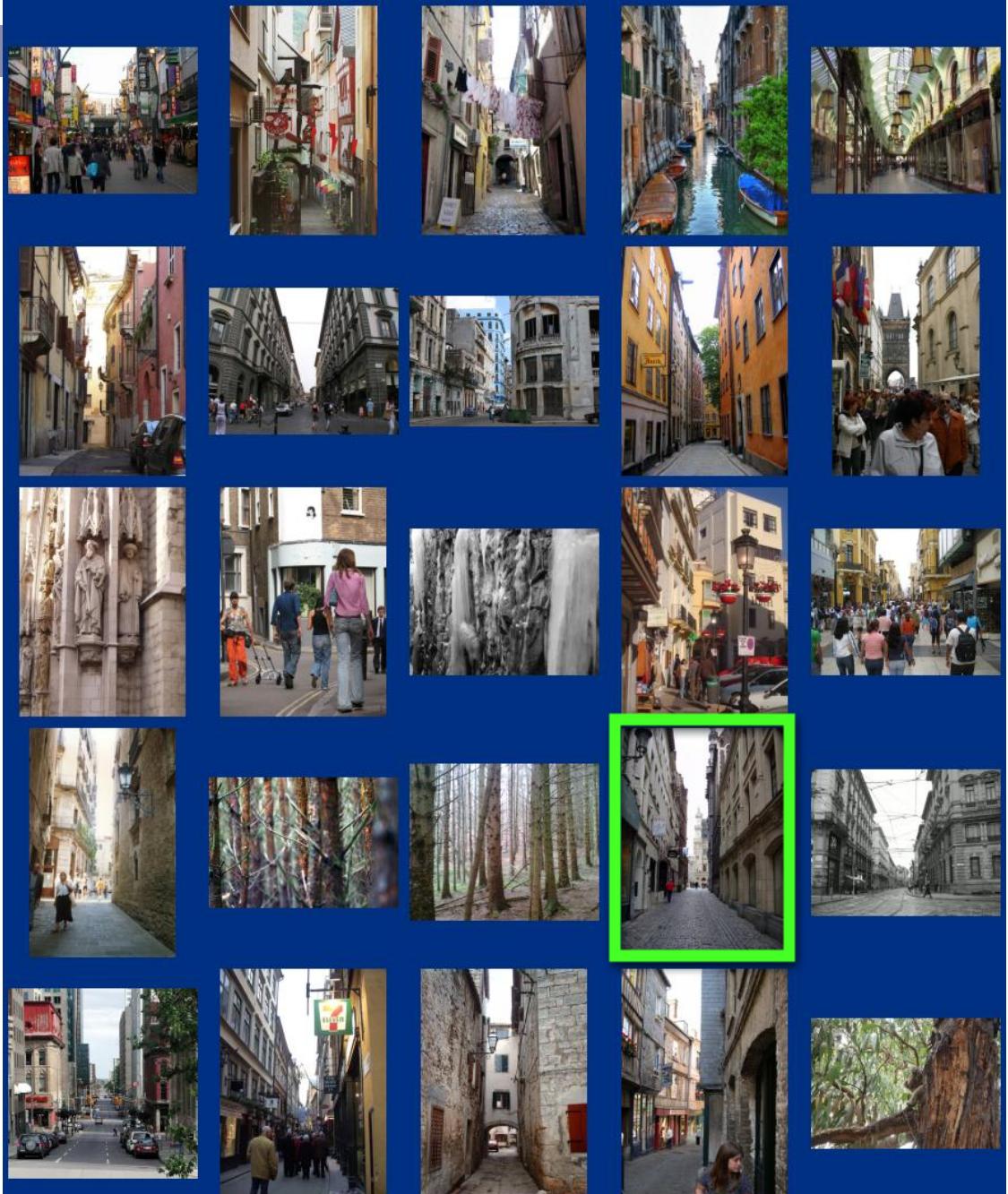
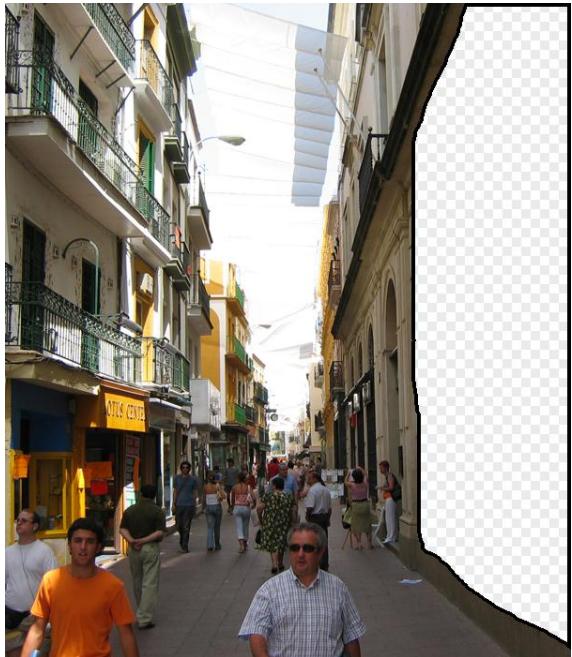




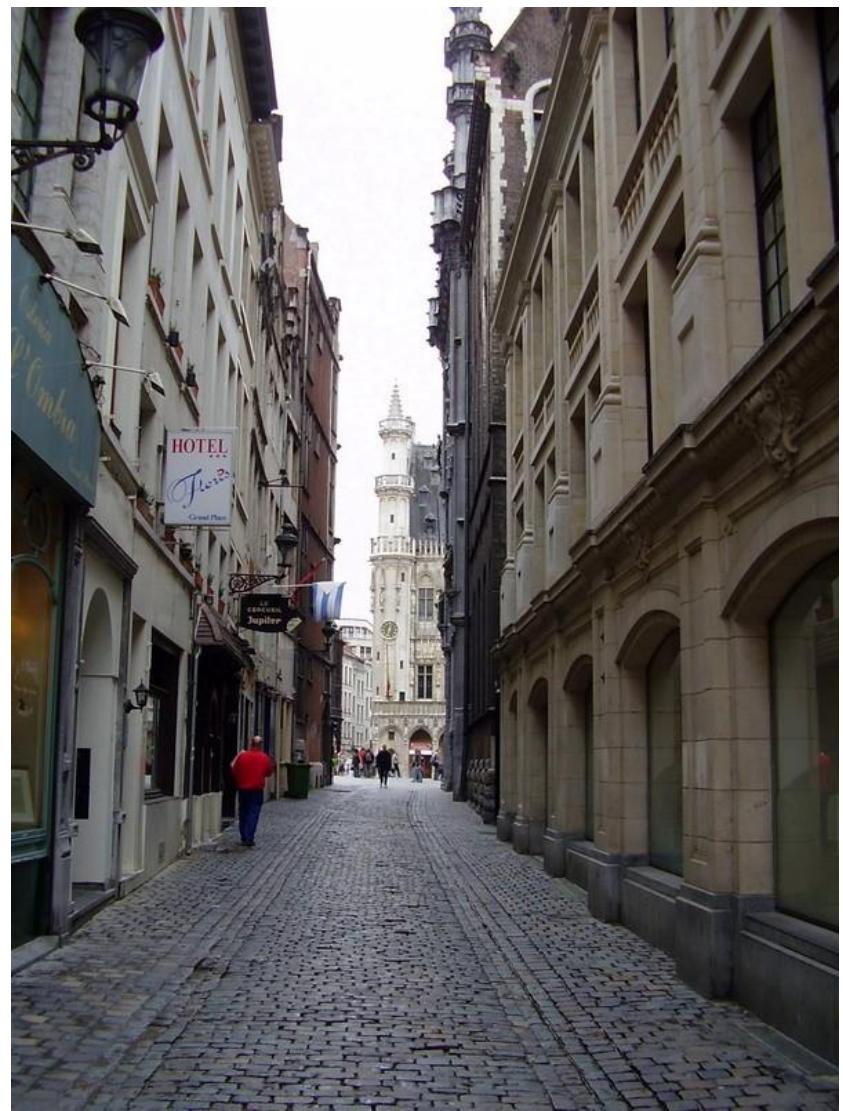




... 200 scene matches

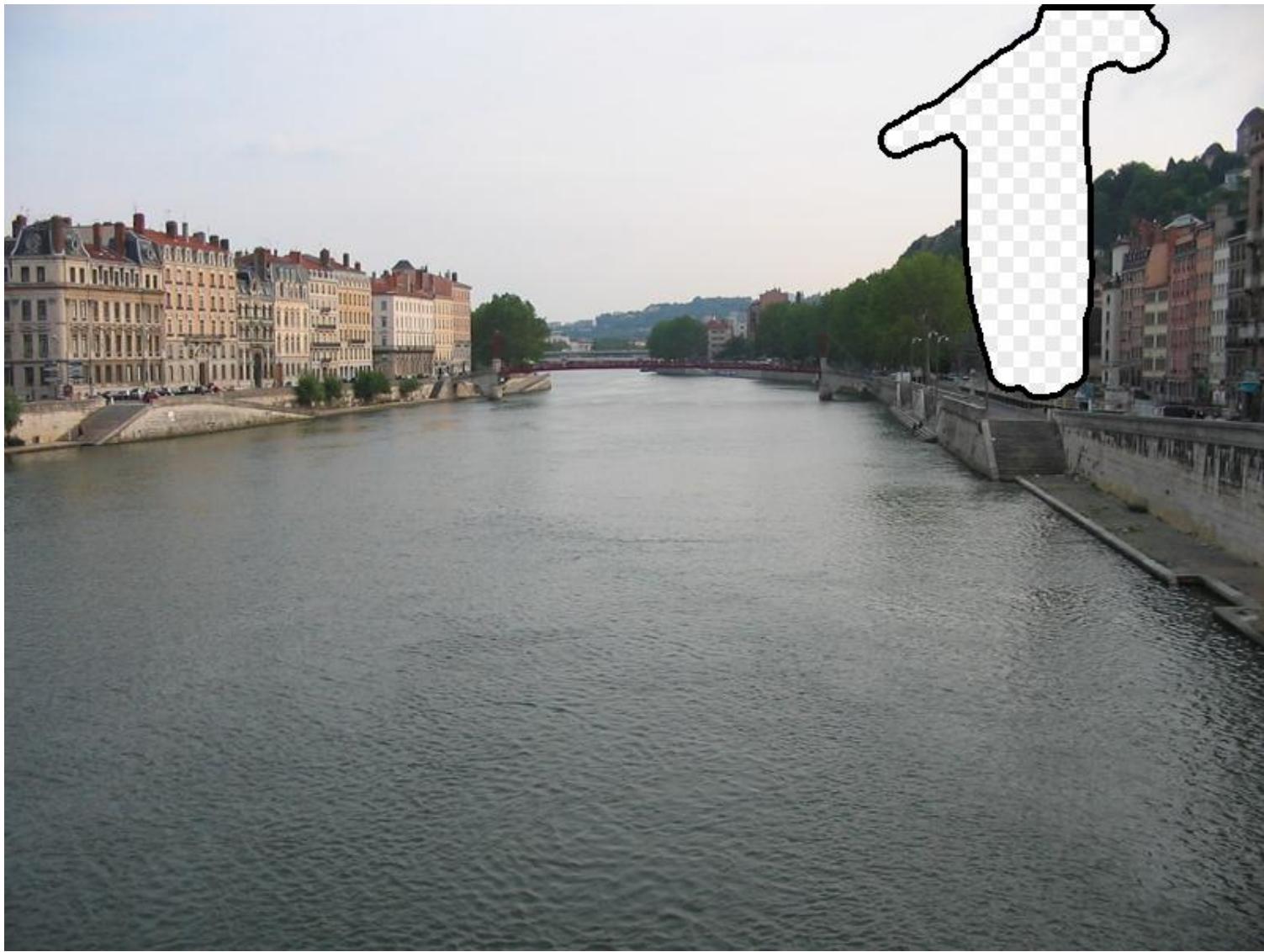


... 200 scene matches

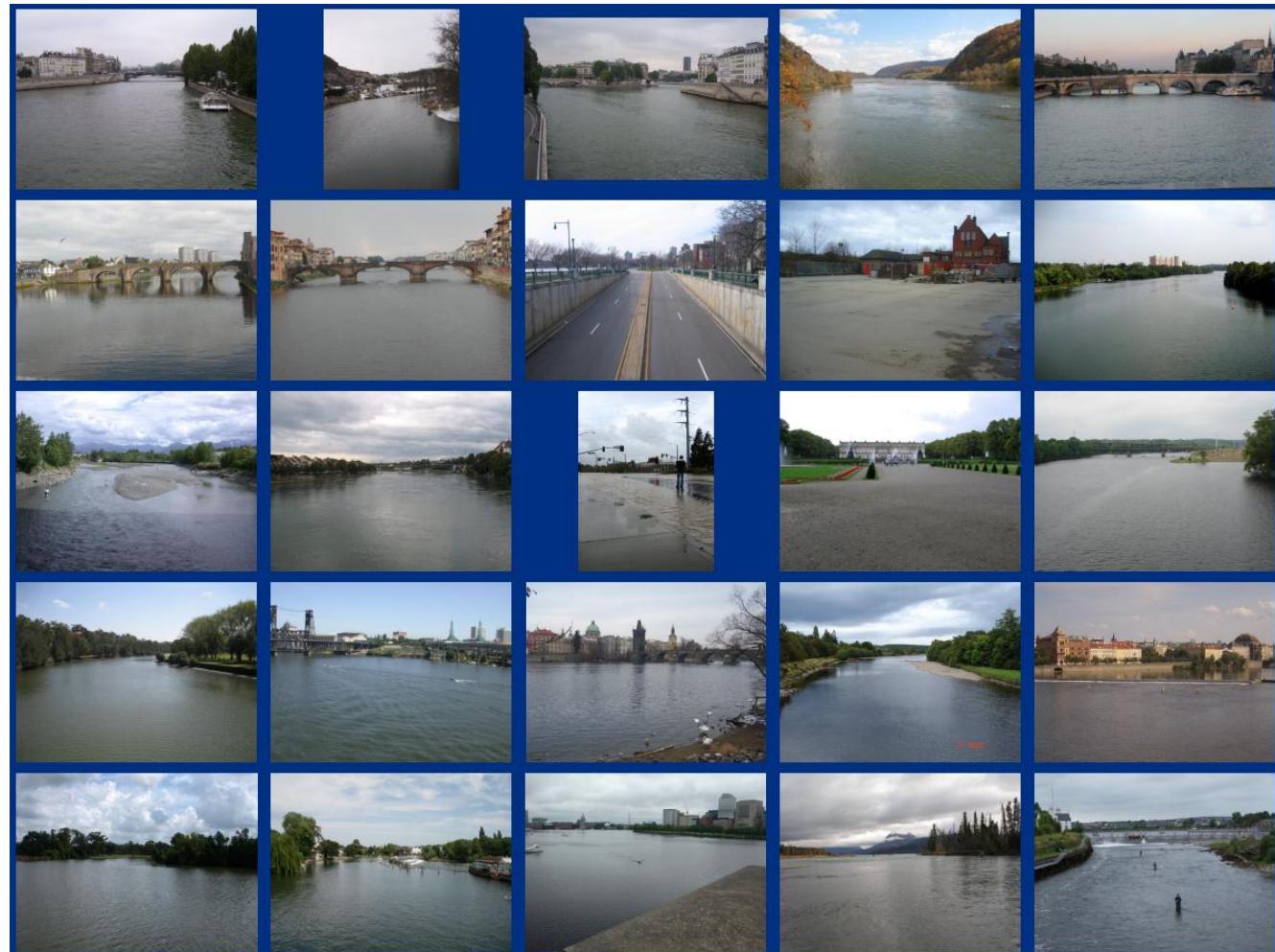
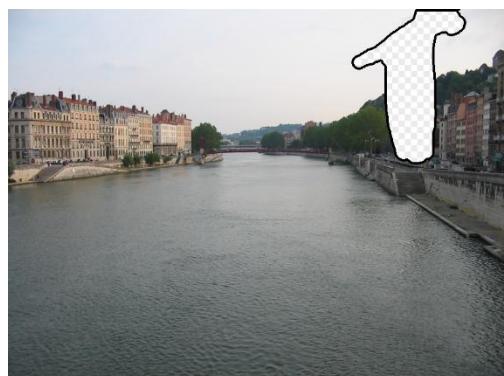




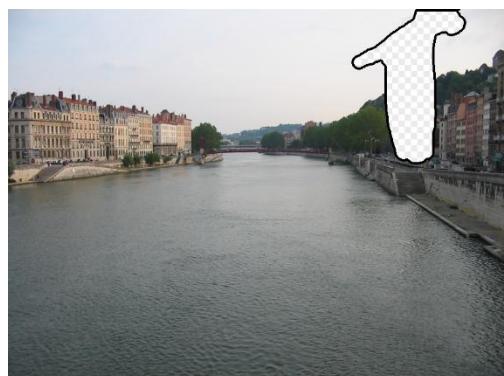




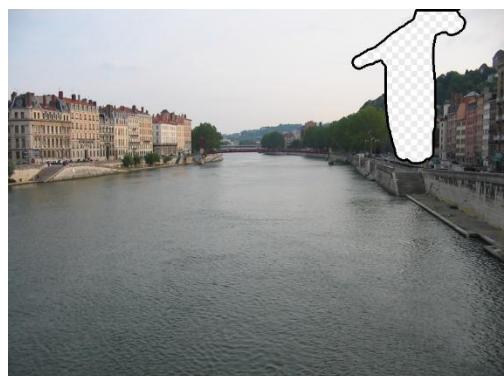




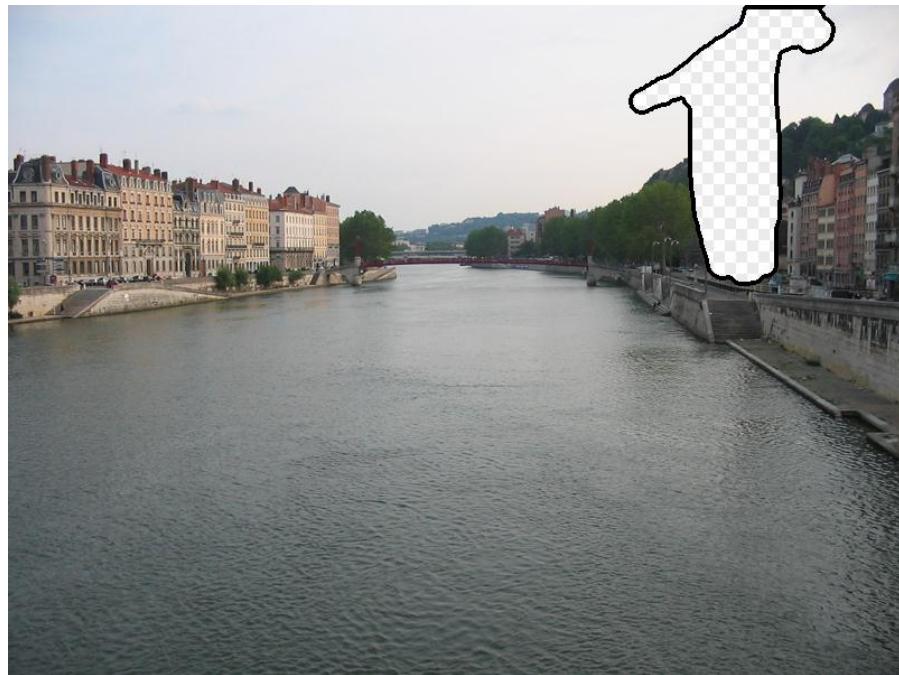
... 200 scene matches



... 200 scene matches



... 200 scene matches

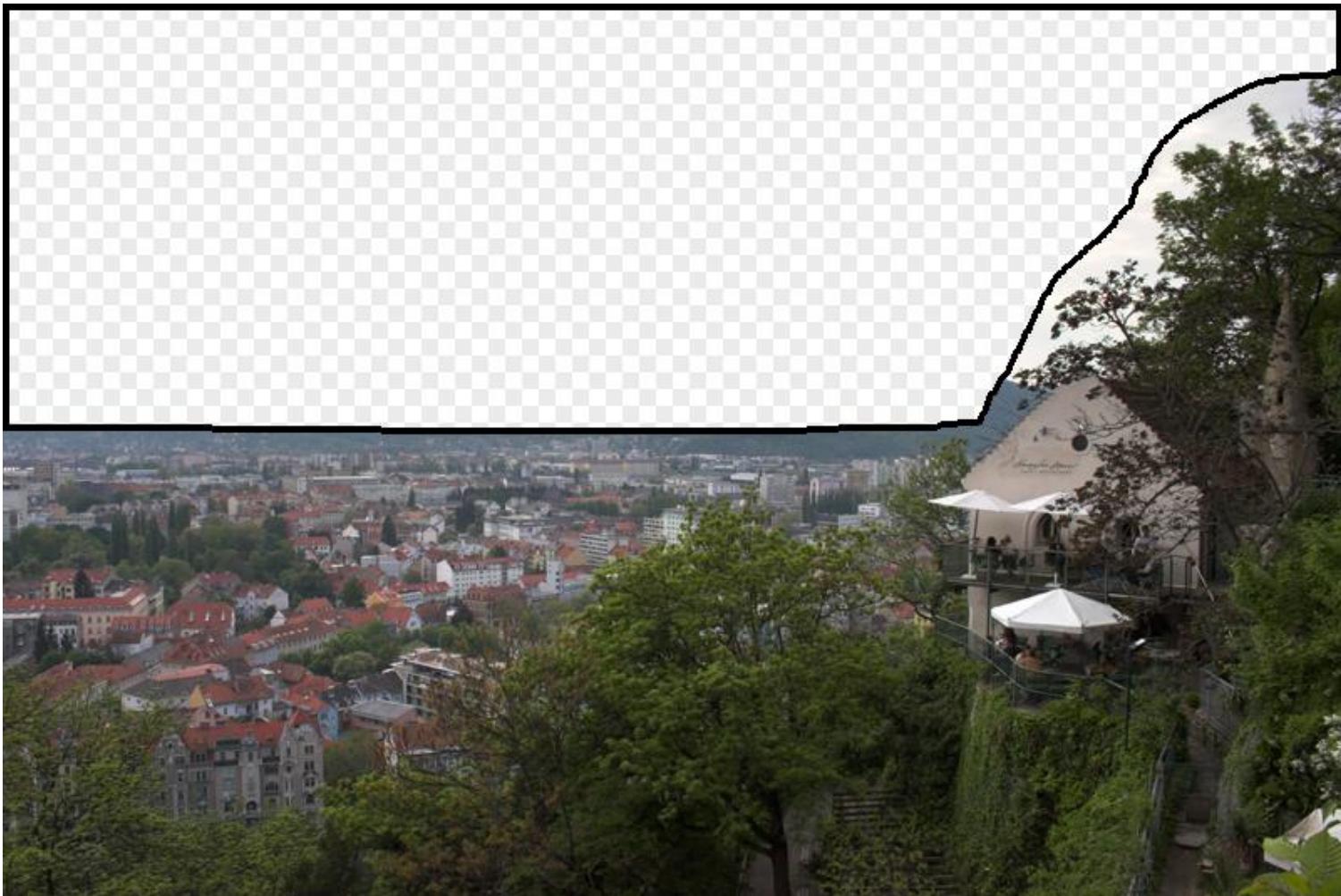




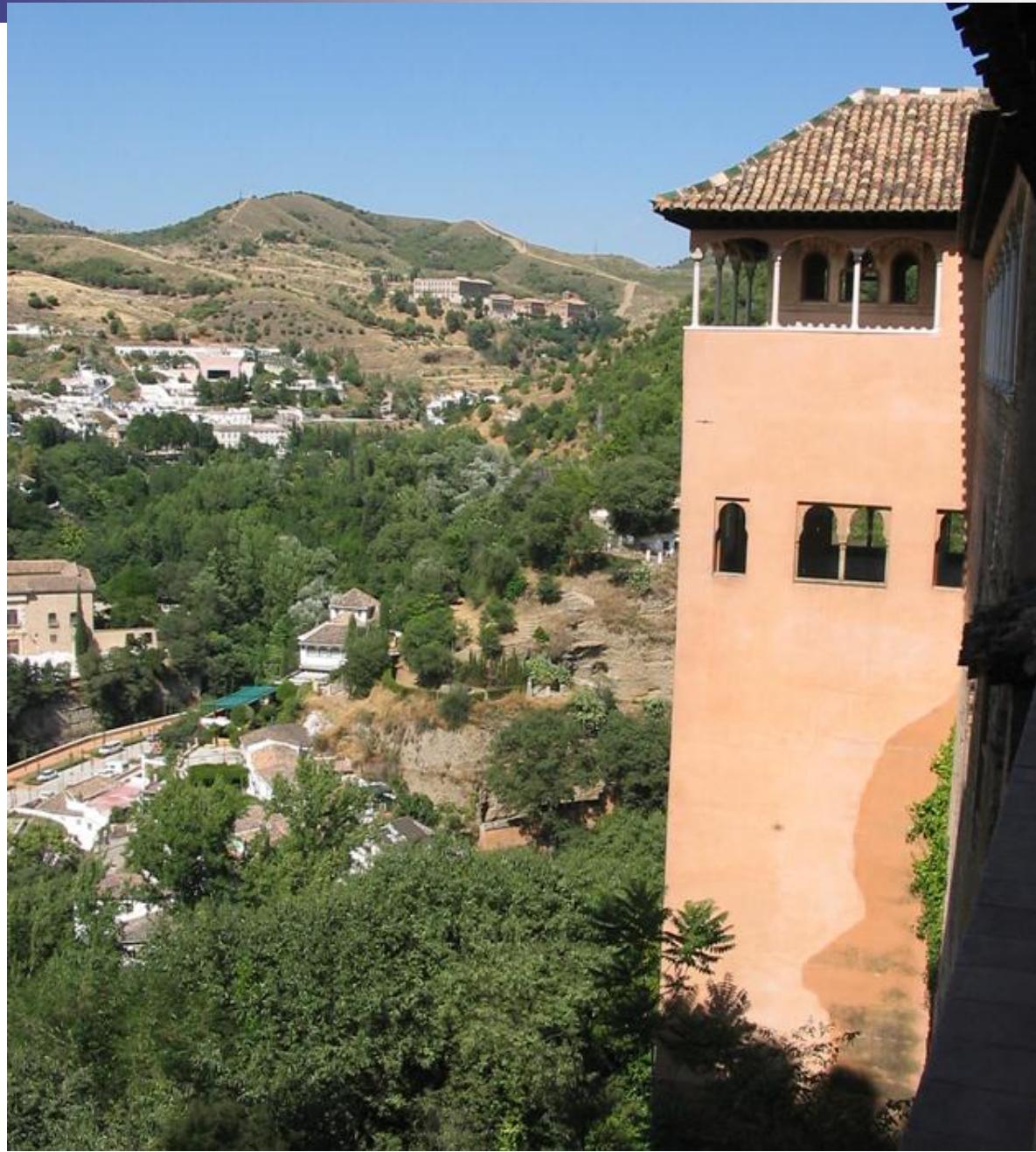


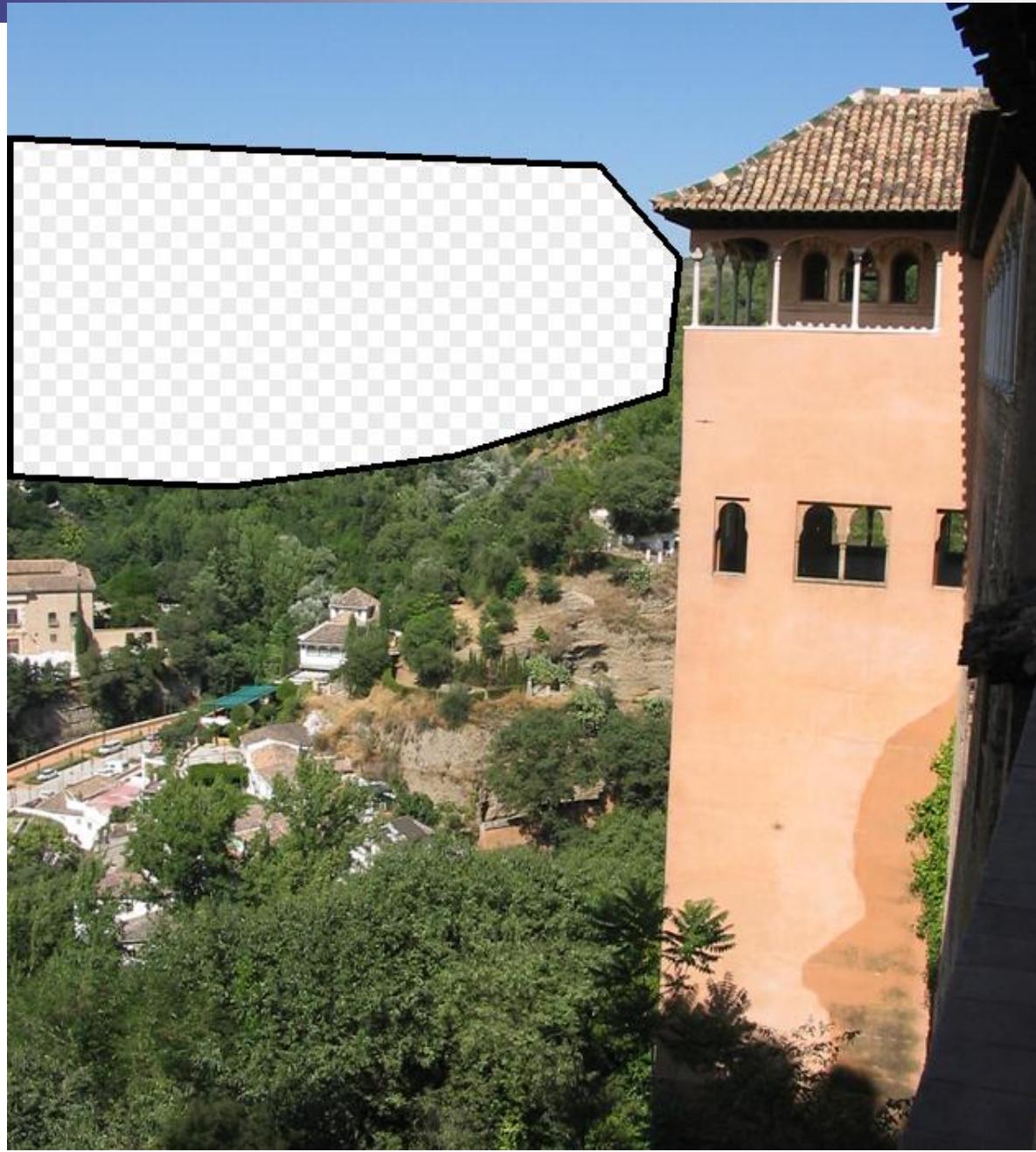










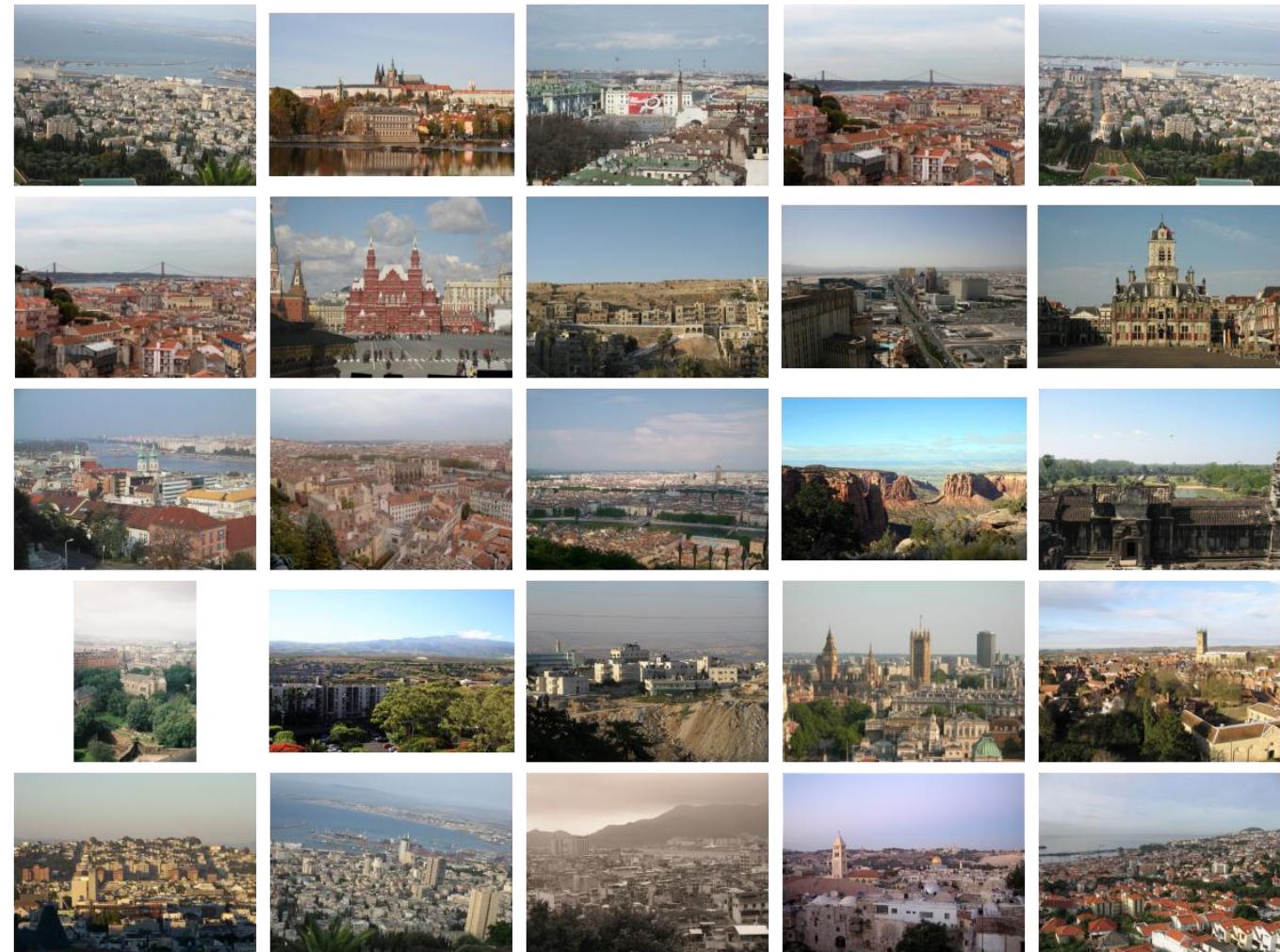
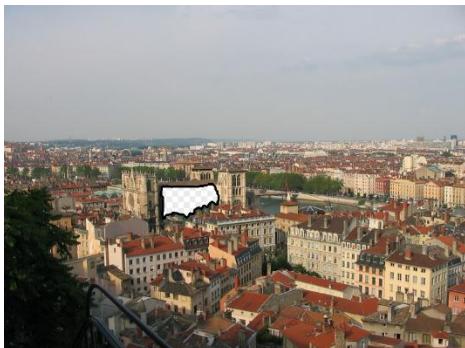
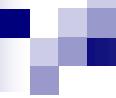










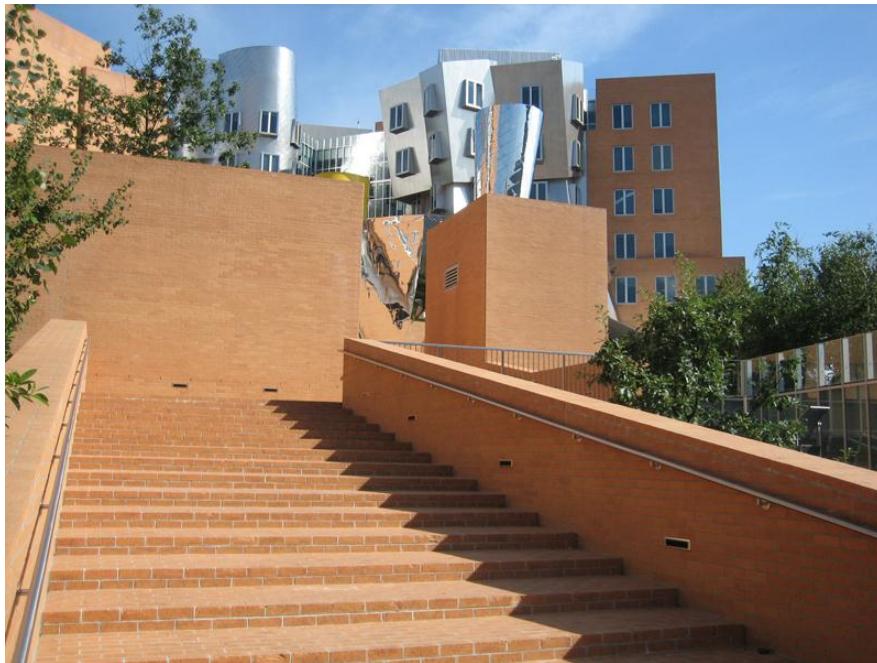


... 200 scene matches





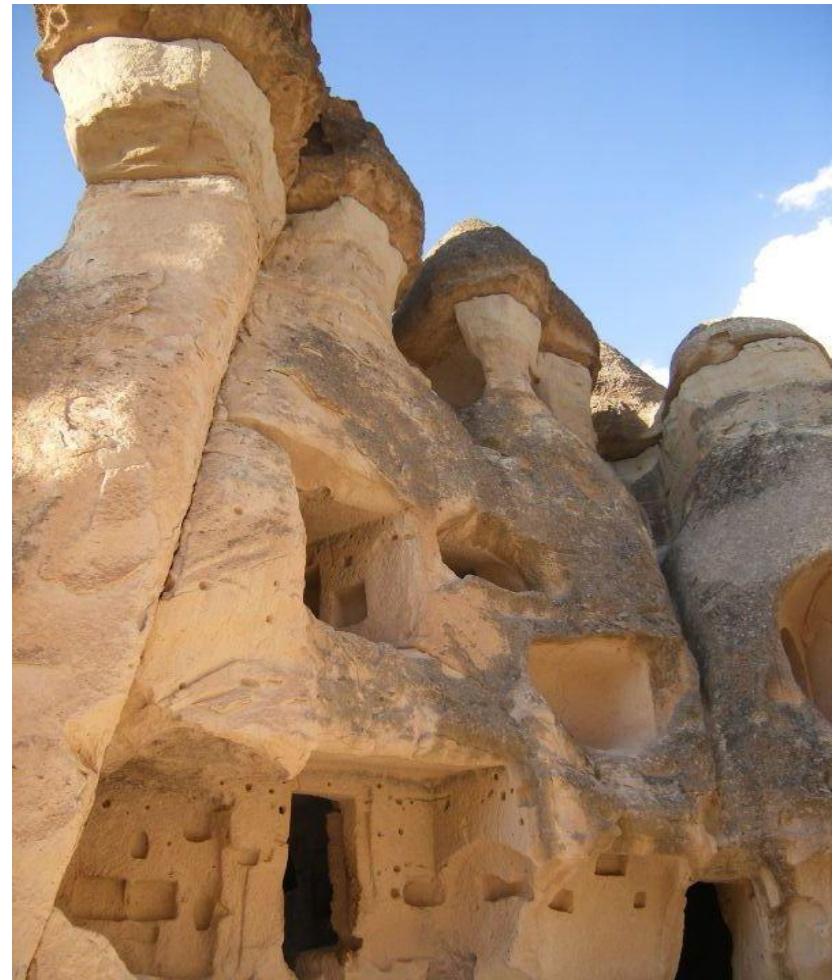
Failures



Failures



Failures



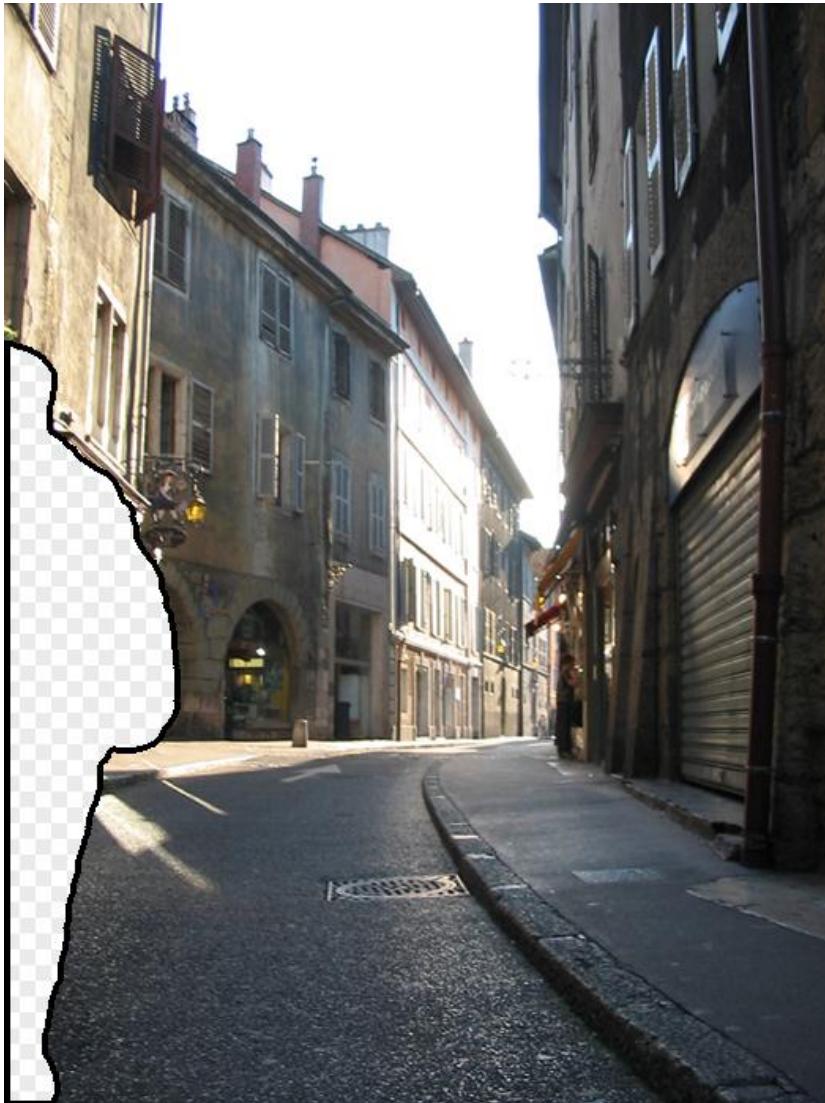
Failures



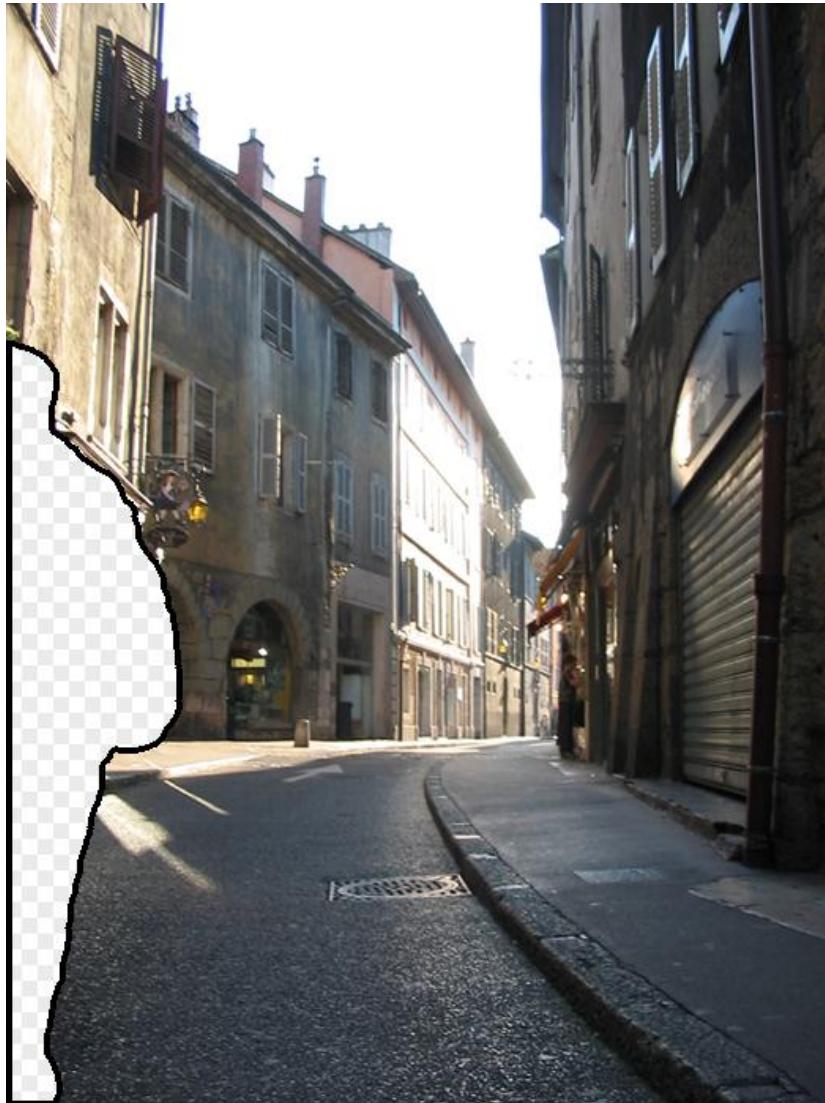
Failures



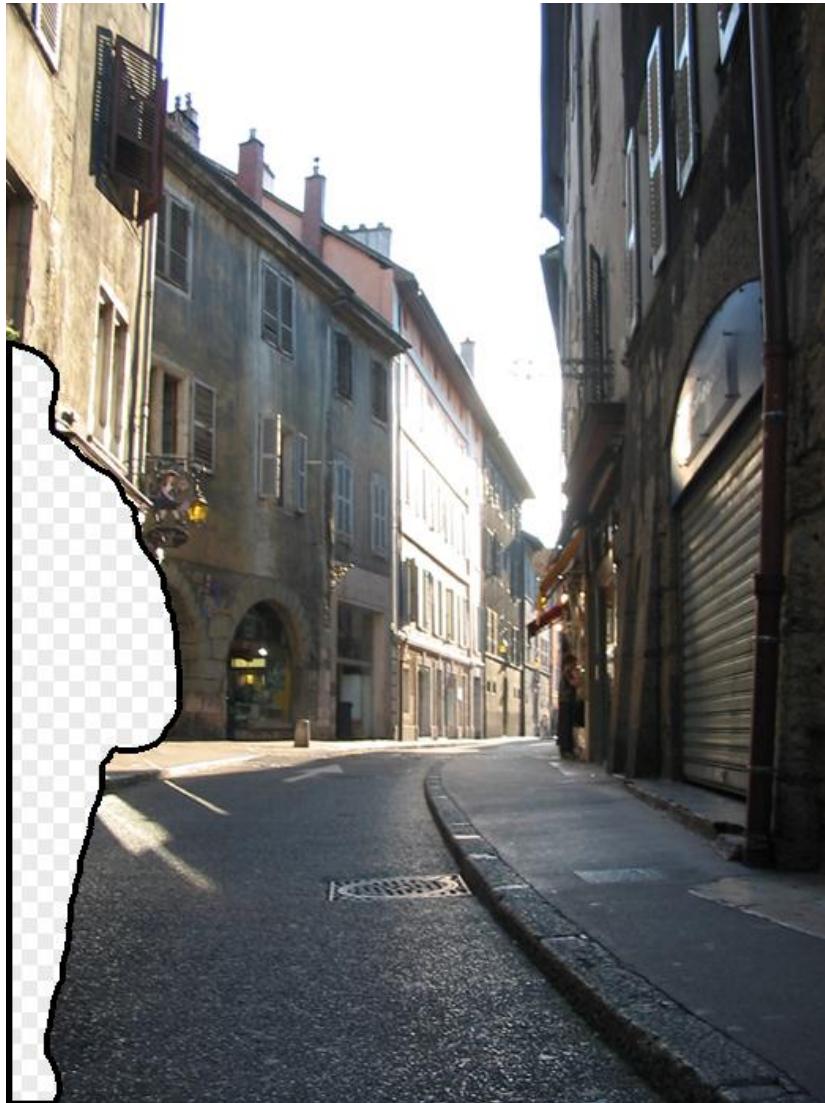
Failures



Failures



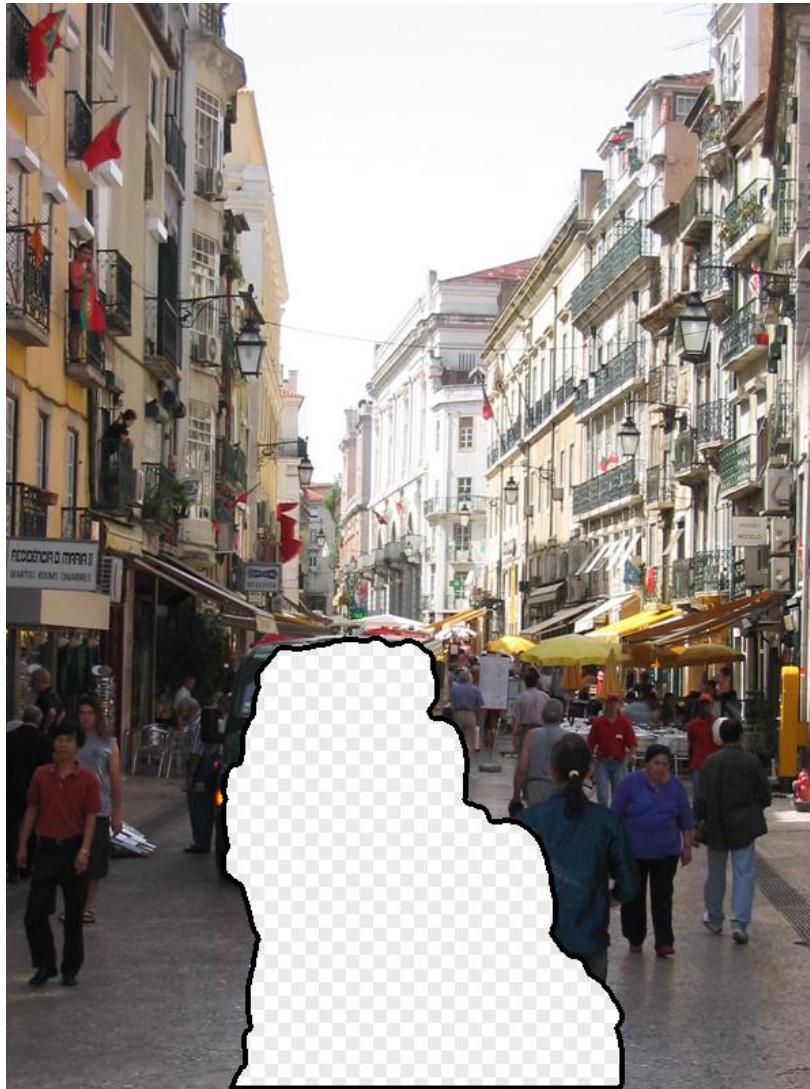
Failures



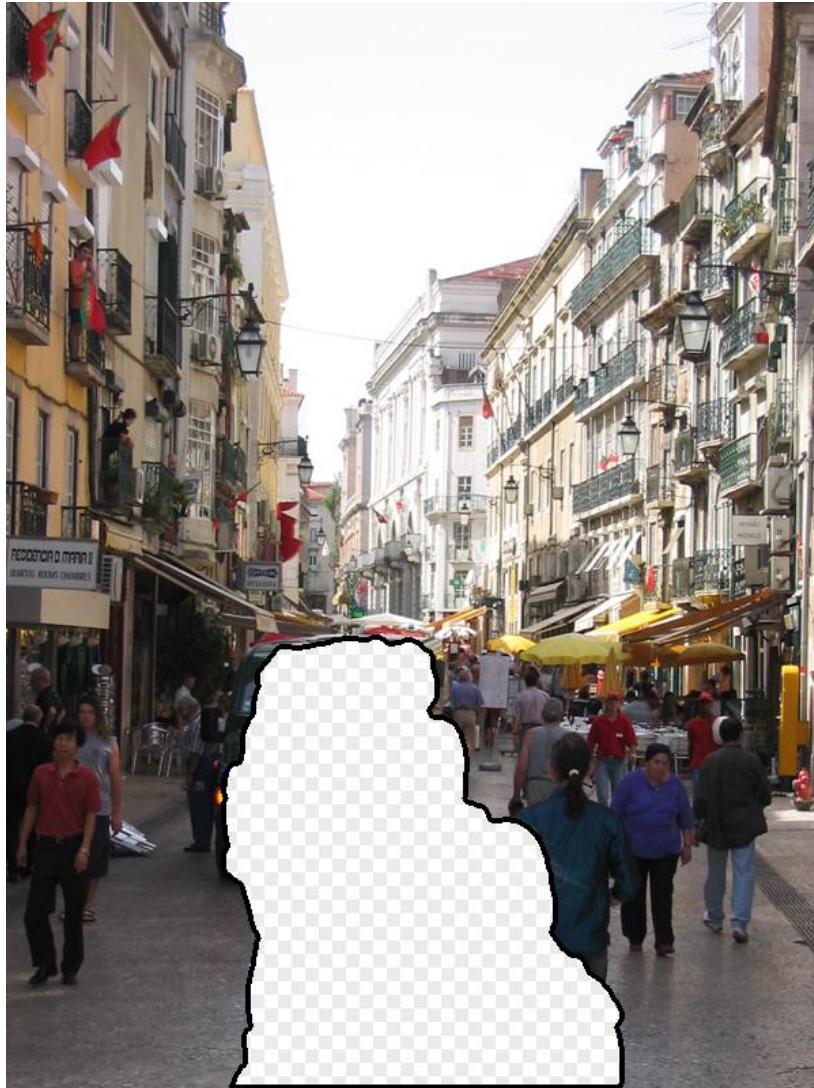
Failures



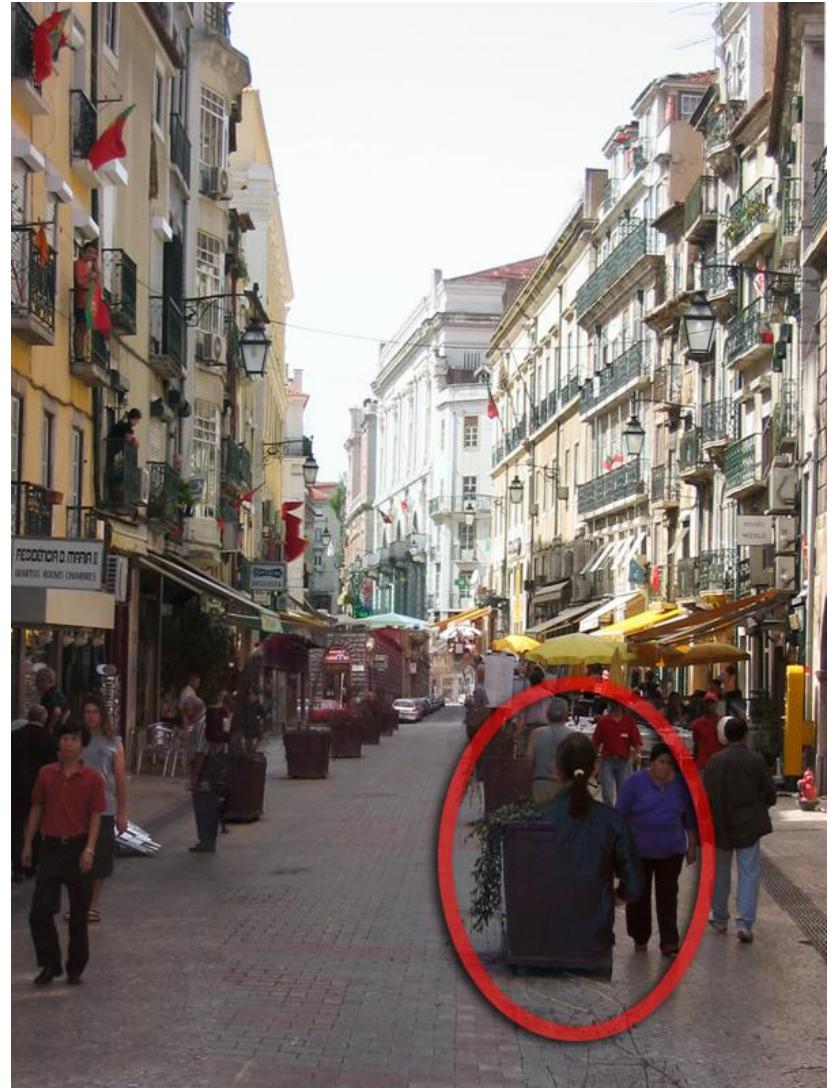
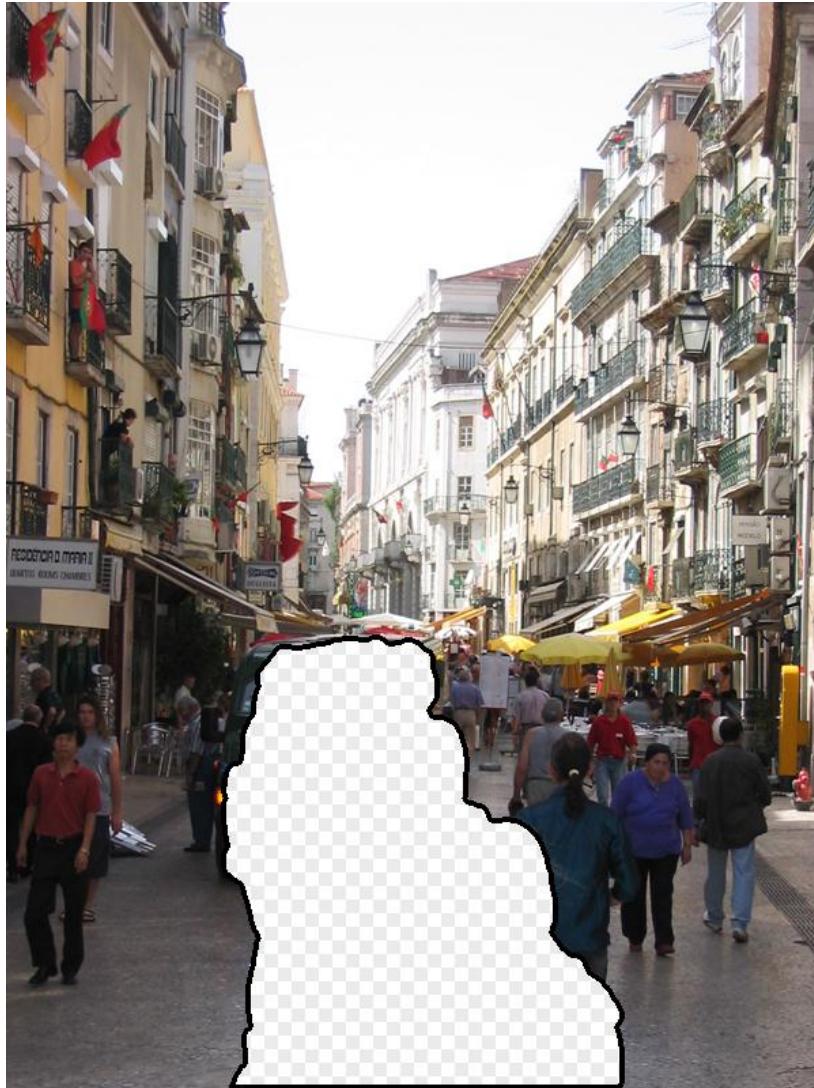
Failures



Failures



Failures



图像补全总结

- 思想：图像自身或图像库信息冗余性
- 关键：按照何种顺序进行填充
- 纹理
 - 基本的纹理合成算法
 - 经典的“剥洋葱”式填充顺序
 - 带有优先级的填充策略
 - Image Completion by Example-Based Inpainting
- 结构

图像补全总结

■ 结构

- 带交互的补全算法

- Image Completion with Structure Propagation

- 局部优化算法

- Space-time video completion
 - PatchMatch: A randomized correspondence algorithm for structural image editing
 - Image Melding: Combining Inconsistent Images using Patch-based Synthesis
 - Shift-map image editing
 - Statistics of patch offsets for image completion
 - Scene Completion Using Millions of Photographs



Thank you!`