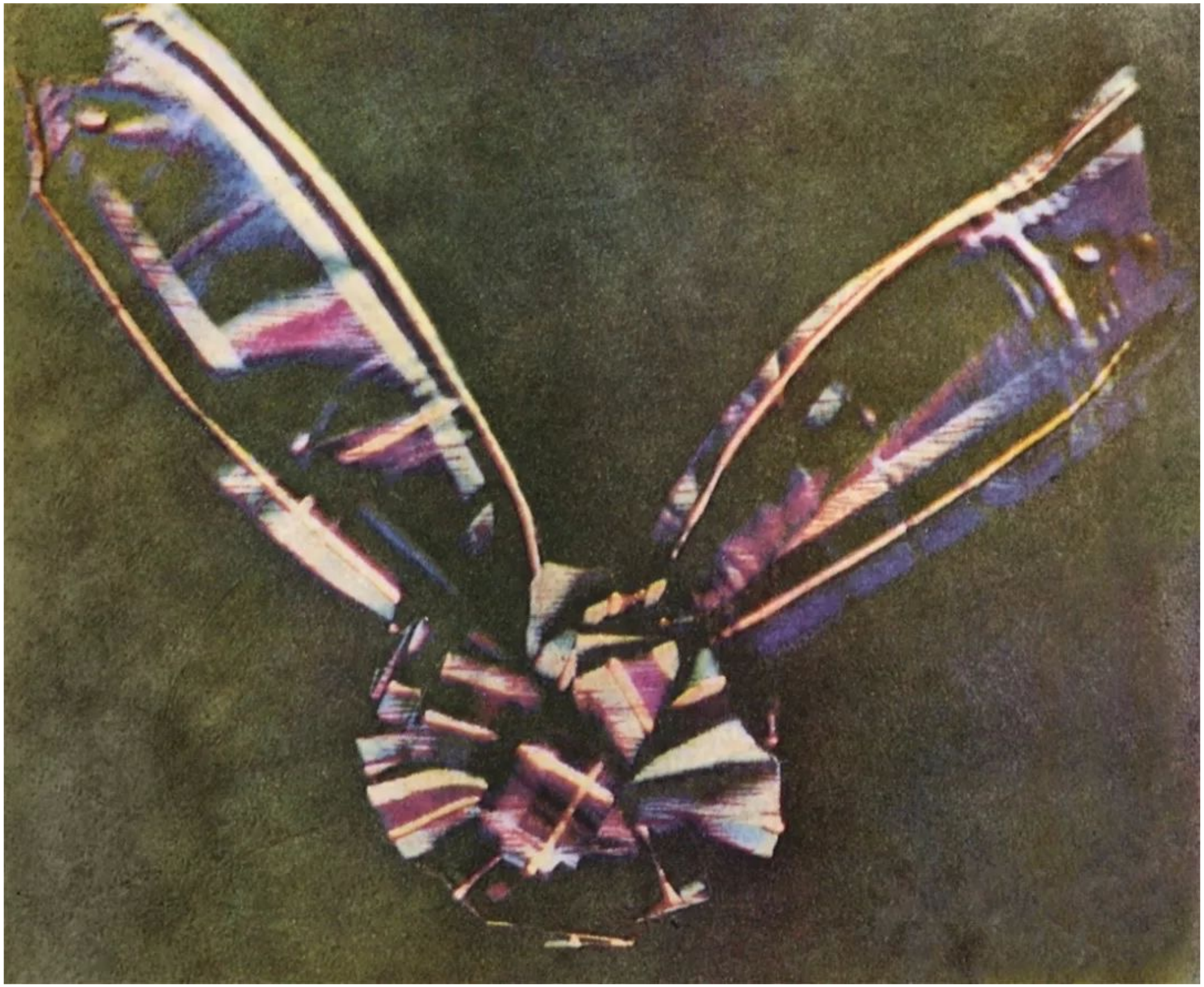
图像上色

计算摄影学

1861年，英国麦克斯维尔（Maxwell 1831–1879）拍摄了第一张彩色照片。



苏格兰花格呢缎带

彩色摄影历史

https://www.youtube.com/watch?v=eYjh4hVWYyc

直至1970年，彩色相片仍然十分稀有，人们只能透过黑白照片回顾历史。

如今，利用技术手段，黑白变彩色轻而易举！


copyright©徐宗懋圖文館


copyright©徐宗懋圖文館


NBC
copyright©徐宗懋圖文館


NBC
copyright©徐宗懋圖文館


Ps
File
Edit
Image
Layer
Type
Select
Filter
View
Window
Help

上色

- 上色是指在计算机辅助下对单色图片或视频添加颜色的过程

- 对灰度图像上色主要有两种方式：
  - 利用样本进行上色
  - 画笔交互式上色

样本上色

+

=

源图像：提供颜色信息

目标图像：待上色

上色结果

Transferring Color to Greyscale Images

T. Welsh, M. Ashikhmin, and K. Mueller

SIGGRAPH 2002

基本方法

源图像

+

目标图像

=

结果

- 扫描目标图像，对于每个像素：
  - 在样本中找到最佳匹配点（综合考虑亮度以及与邻域像素的亮度标准差）
  - 将匹配点的颜色赋予该像素

最终算法

源图像

+

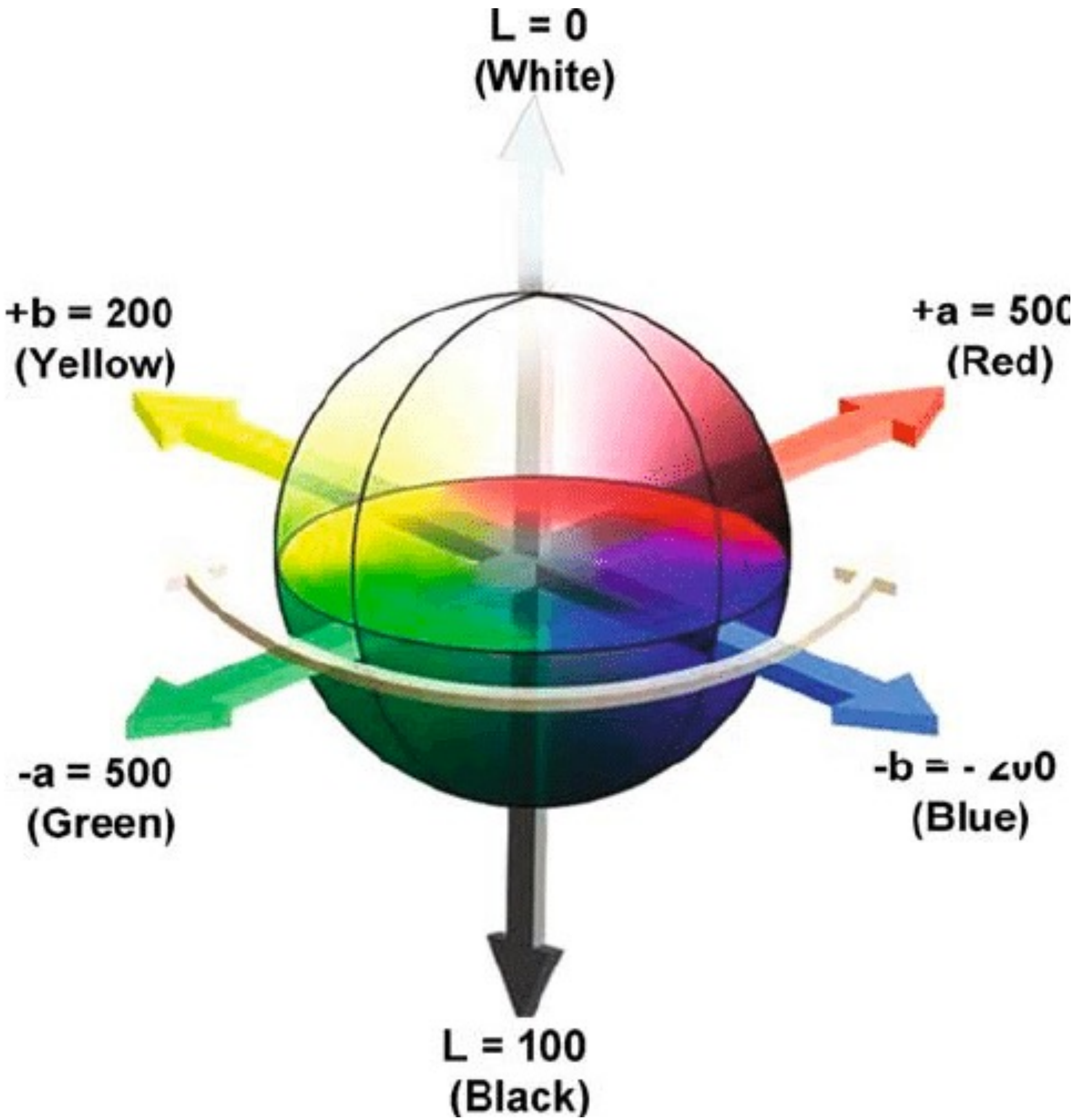
目标图像

=

结果

- 将源图像、目标图像从RGB空间转换到Lab空间
- 对转换后的源图像做 luminance remapping
- 在源图像中做 jittered sampling，得到样本（约200个）
- 扫描目标图像，对于每个像素：
  - 在样本中找到最佳匹配点（综合考虑亮度以及与邻域像素的亮度标准差）
  - 将匹配点的 $\alpha\beta$ 赋予该像素，保留目标图像的L通道不变
  - 将目标图像从Lab空间转回RGB空间

转换到LAB空间



L = 0
(White)
+b = 200
(Yellow)
+a = 500
(Red)
-a = 500
(Green)
-b = - 200
(Blue)
L = 100
(Black)

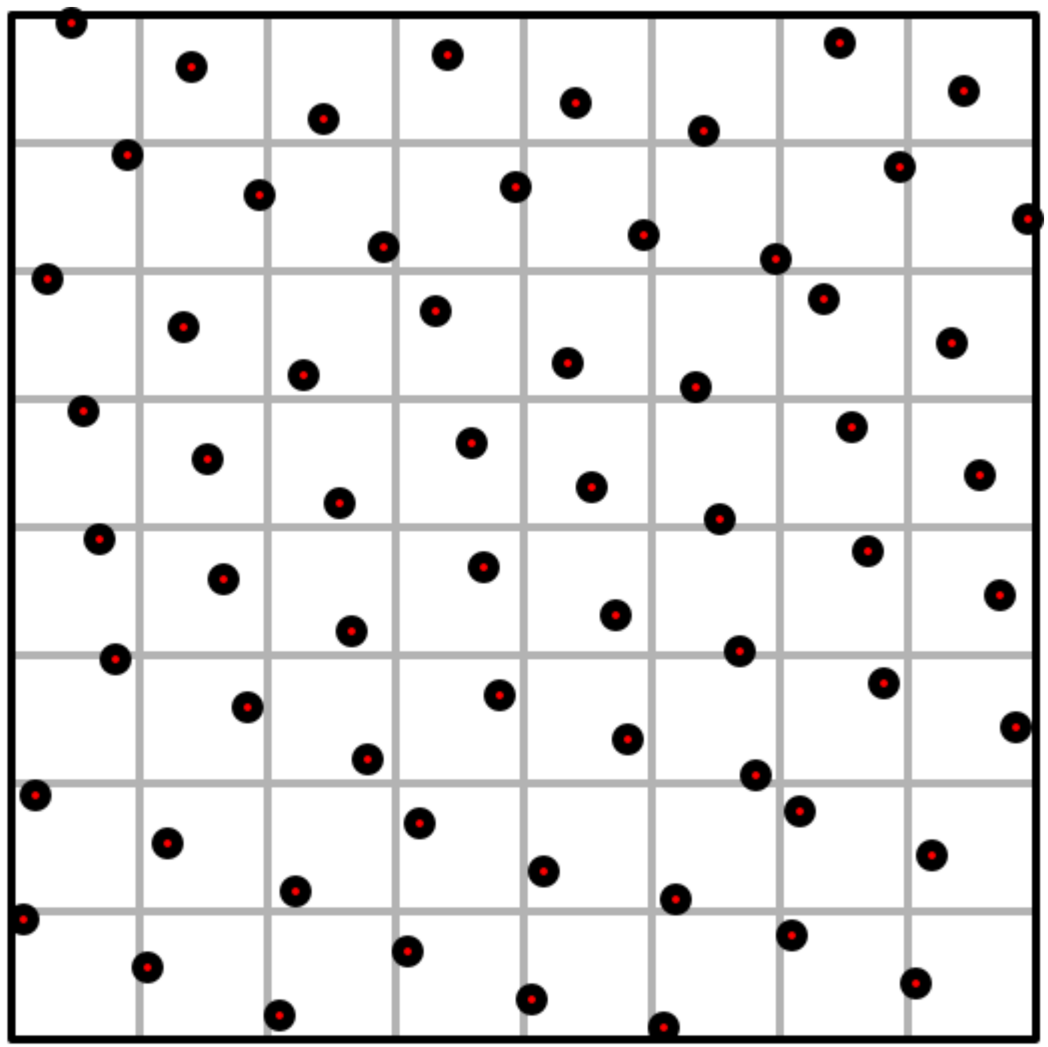
Luminance remapping

- 对转换后的源图像做 luminance remapping

$$L(p) = \frac{\sigma_B}{\sigma_A}\left(L(p) - \mu_A\right) + \mu_B$$

$L(p)$: 源图像某像素的亮度值　σ_A, σ_B :　亮度标准差　μ_A, μ_B :源图像与目标图像的亮度均值

Jittered sampling

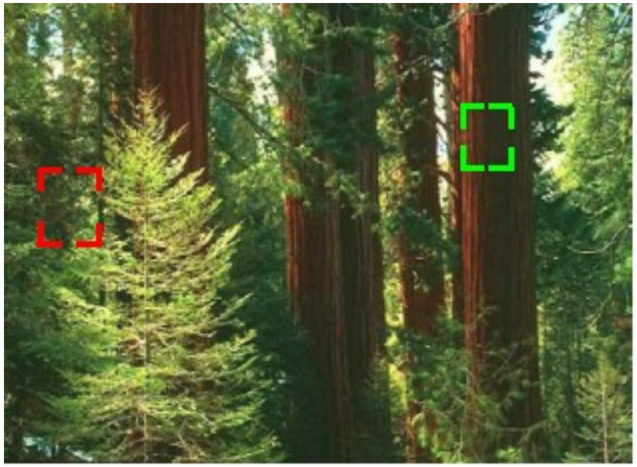


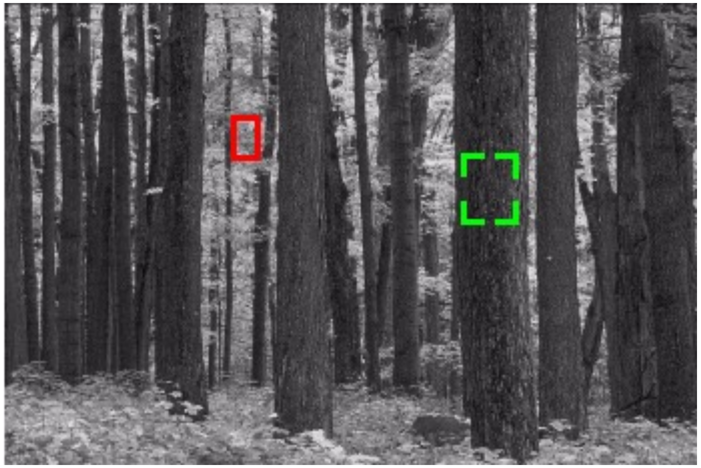
问题

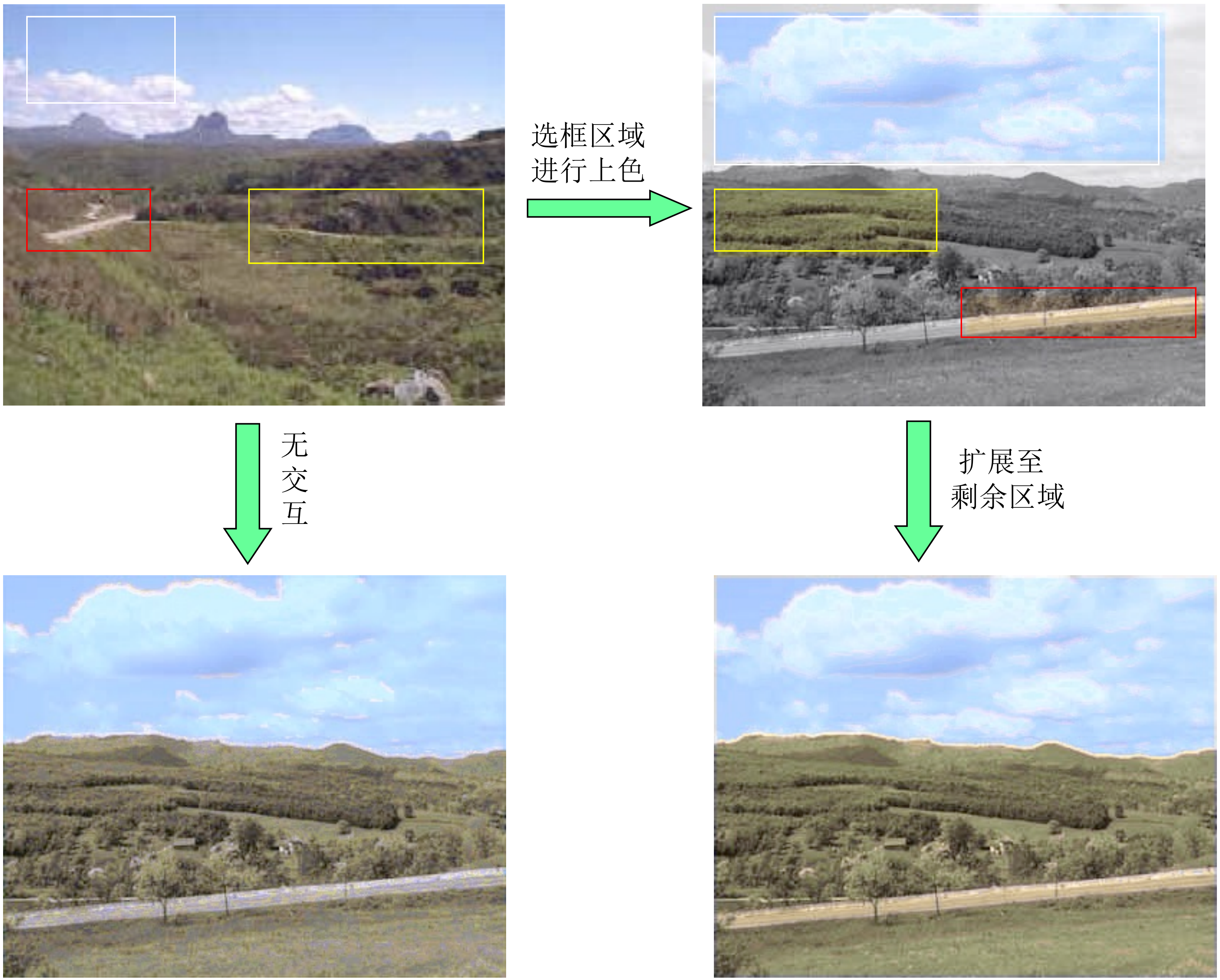
当源图像与目标图像颜色相一致的区域亮度值却不一致，导致上色效果不理想

解决方案

加入交互，在源图像与目标图像中指定相对应的区域





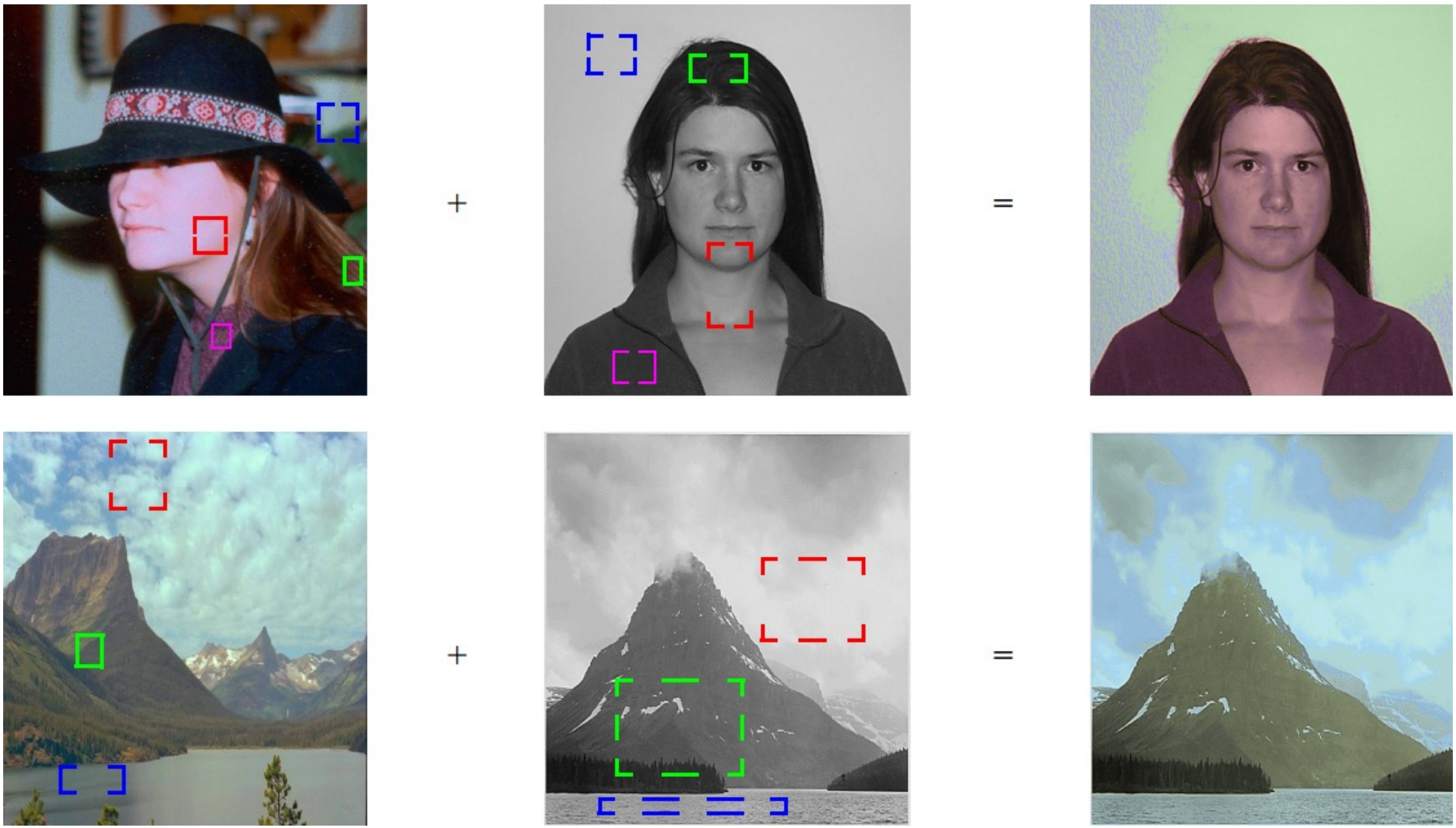

选框区域
进行上色
无
交
互
扩展至
剩余区域

细节

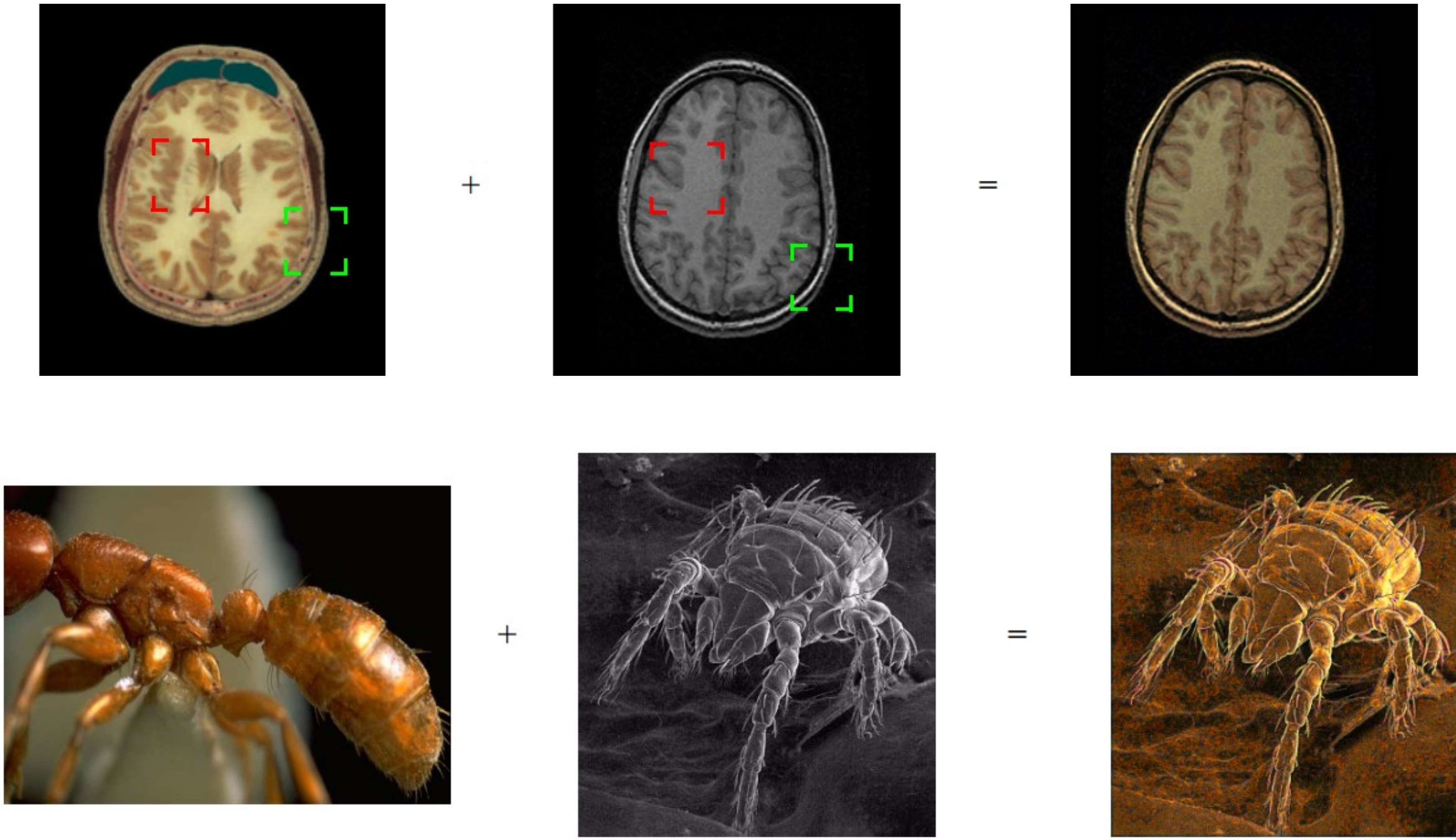
- 对选框部分进行上色
  - 对选框部分做 lumiance remapping
  - 源图像中，每块区域做 jittered sampling (~50)

- 扩展剩余区域
  - 对目标图像上的每一个像素，在目标图像已上色的区域中找最佳匹配（根据亮度）
  - 对剩余部分上色

结果展示



结果展示



扩展

- 选择更为有效的匹配函数
- 参考图像的选择
- 添加约束条件，例如空间一致性

Colorization by Example

R. Irony, D. Cohen-Or, and D. Lischinski

Eurographics Symposium on Rendering, 2005

优点

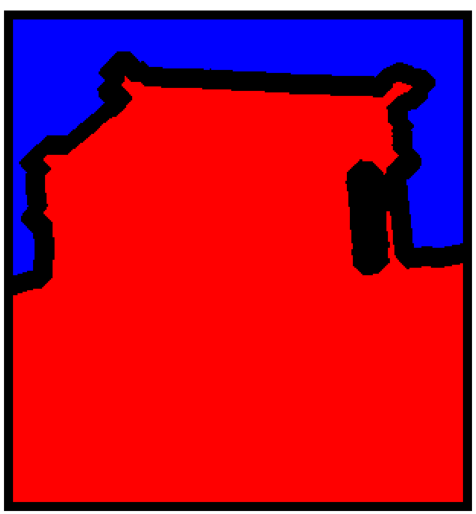
·提升空间一致性

Welsh et al.

优点

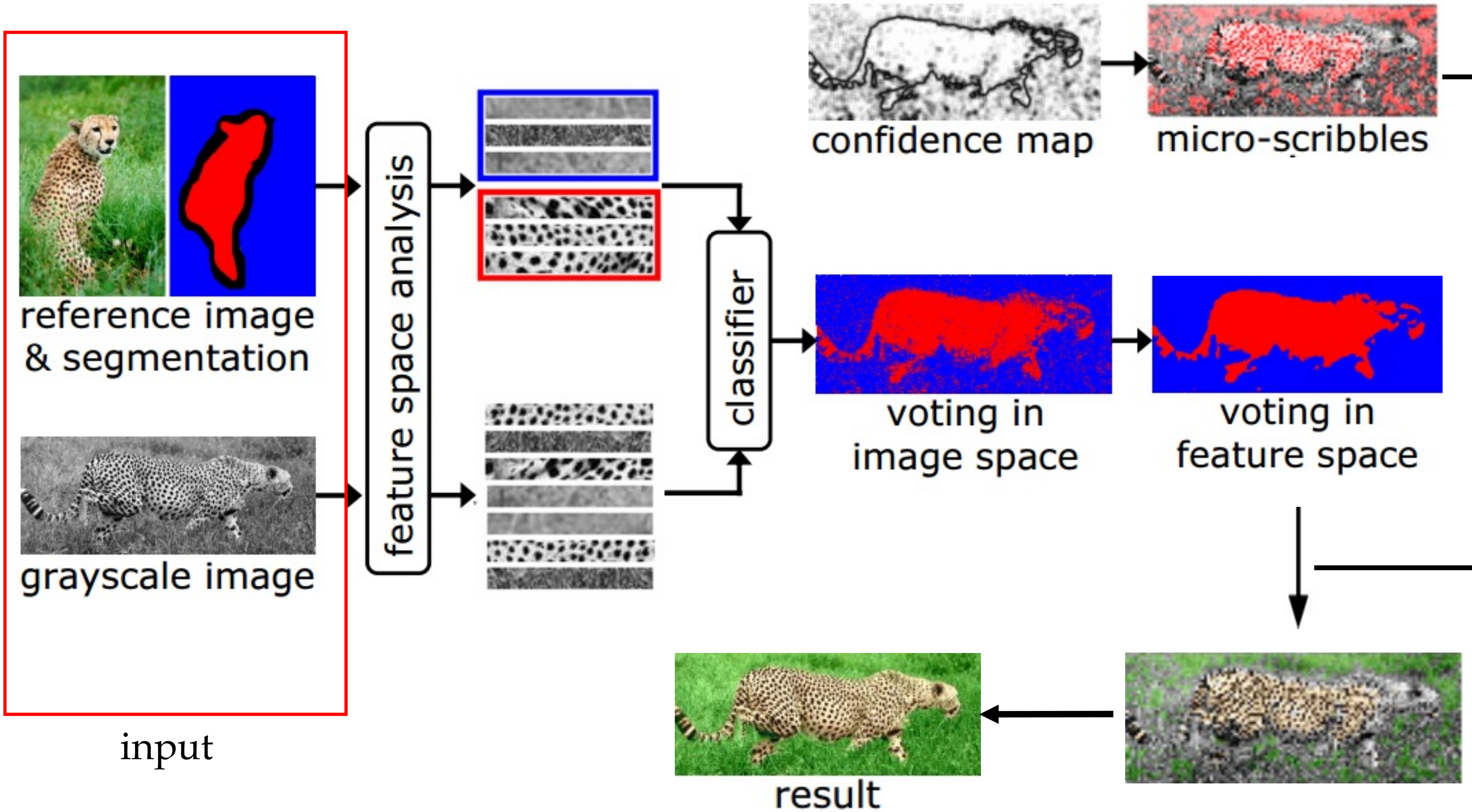
·提升空间一致性

新方法



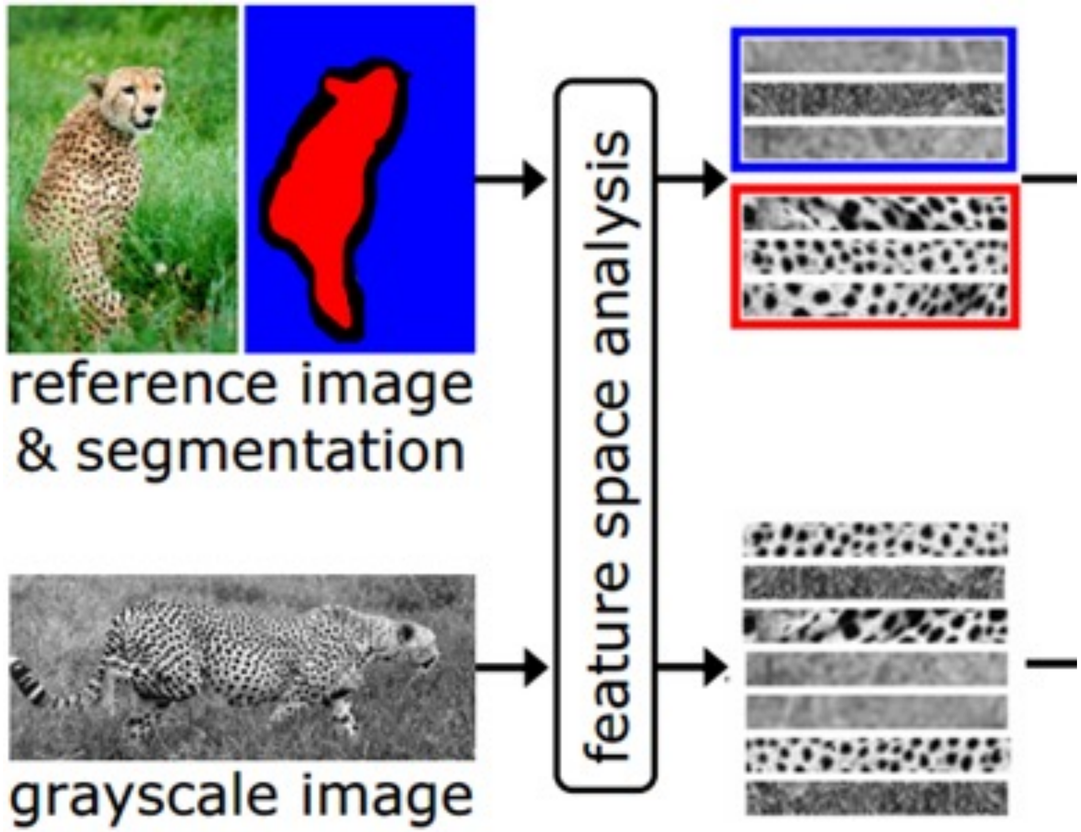
概述

Four steps: (1) training, (2) classification, (3) color transfer, (4) optimization



input

概述



根据参考图像的亮度通道以及分割结果，进行监督学习，构建一个低维的特征向量空间以及分类器。使得某一像素，依据其少量邻域像素，能够判断该像素属于哪块分割区域

分类器

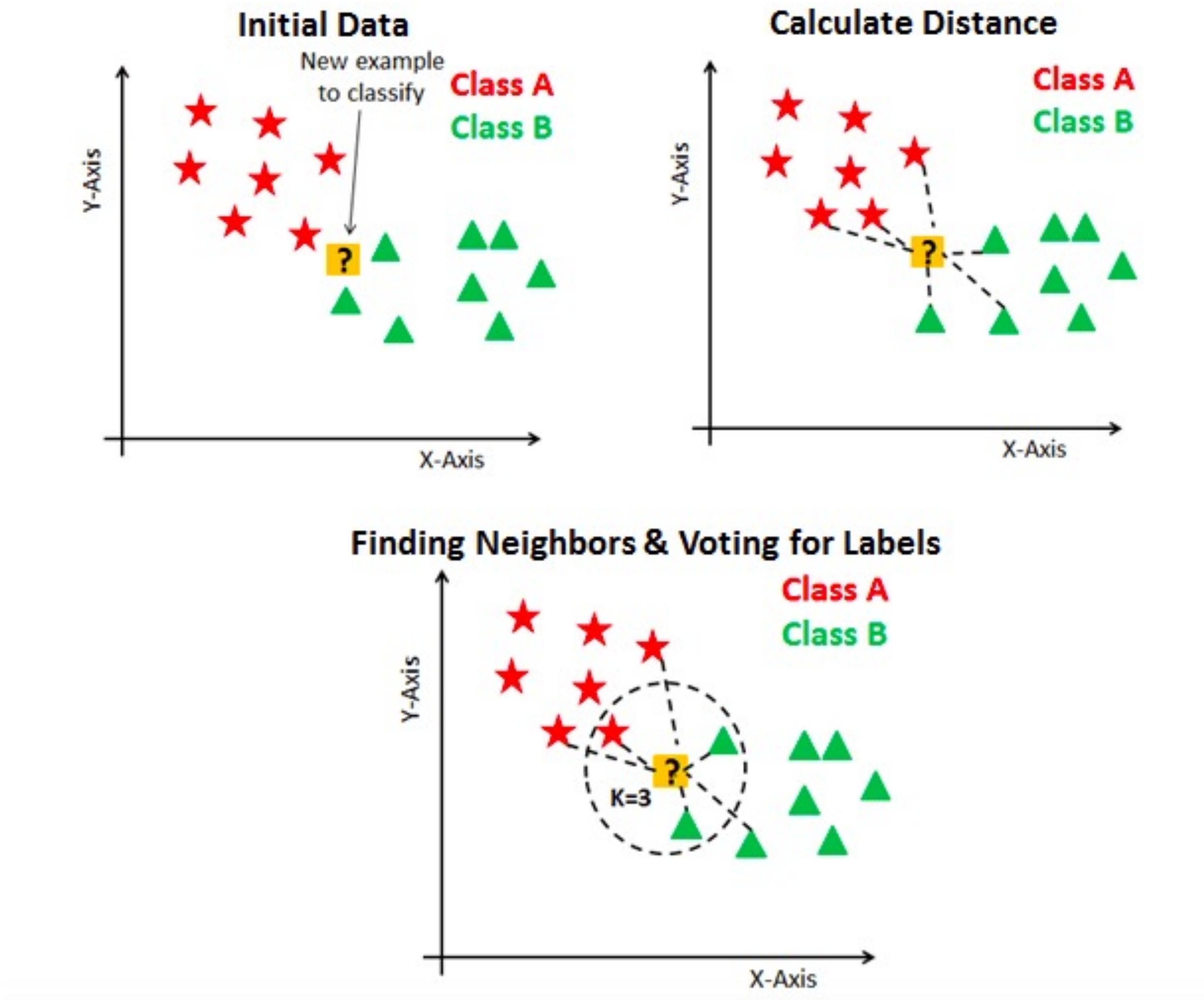
- 分类器的作用：针对一个新的特征向量，能判别其属于哪个类别。

- 简单方式：在已分类的特征向量空间内，寻找与新特征向量最相似的。

- 更为合理的方式：

  - 采用KNN，寻求K个与该向量相似度较高的特征向量
  - 观察K个特征向量的归属类别，分布最多的类别作为结果

KNN



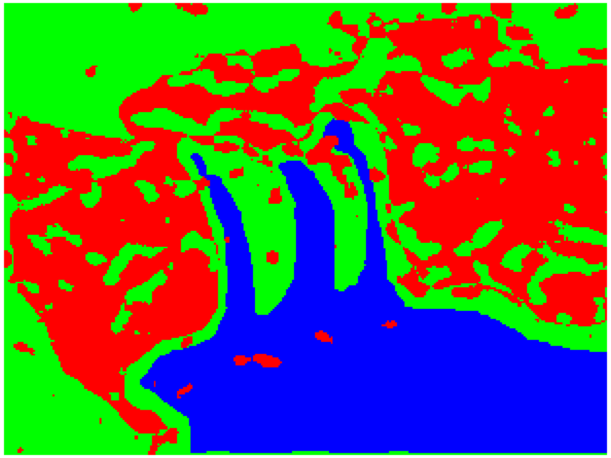
Initial Data
New example
to classify
Class A
Class B
Y-Axis
X-Axis
Calculate Distance
Class A
Class B
Y-Axis
X-Axis
Finding Neighbors & Voting for Labels
Class A
Class B
K=3
Y-Axis
X-Axis

更多结果

参考图像

手动分割

待上色图像

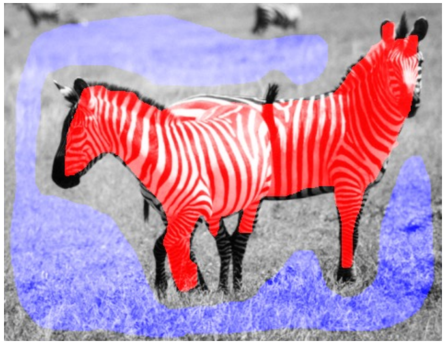


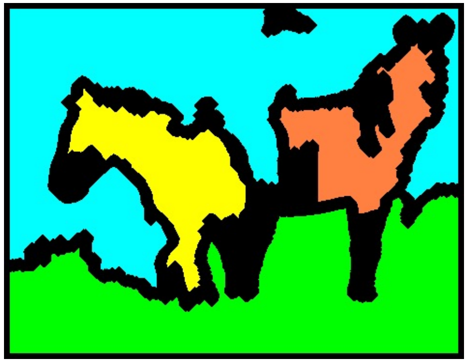
自动分割结果

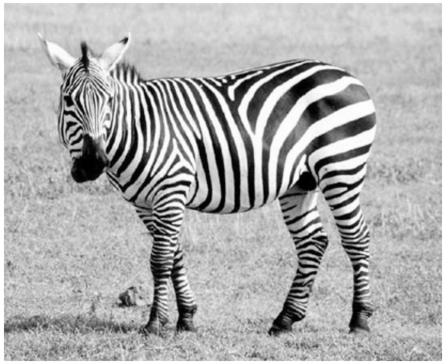
上色结果

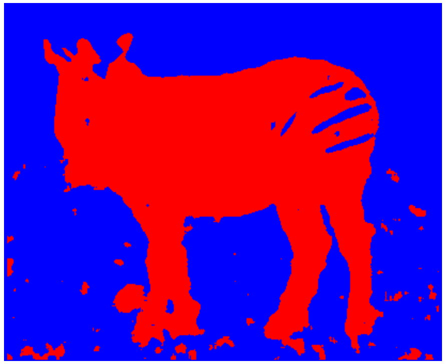
更多结果

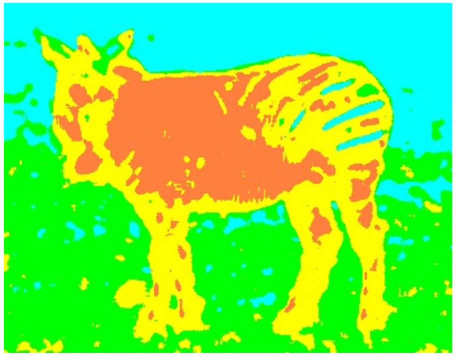

源图像


手动分割


自动分割


目标图像


基于手动分割分类


基于自动分割分类


得到相同结果

自动分割减轻人
工负担

Colorization Using Optimization

A. Levin, D. Lischinski, Y. Weiss

SIGGRAPH 2004

画笔交互式上色

input：带画笔的灰度图

output：上色图

基本思想


- 相邻的两个像素，如果亮度相似，那么颜色也应保持相似
- 基于该假设，上色问题转换为最小化目标方程：

$$J(U)=\sum_{r}\left(U(r)-\sum_{s\in N(r)}w_{rs}U(s)\right)^{2}$$

$$J(U)=\sum_{r}\left(U(r)-\sum_{s\in N(r)}w_{rs}U(s)\right)^{2}$$

$U(r),U(s)$: 像素r,s的U分量

$N(r)$: 像素r的邻域像素集

w_{rs} :权值 $$w_{rs}\propto e^{-(Y(r)-Y(s))^2/2\sigma_r^2}$$

μ_r,σ_r :r像素邻域范围内的亮度均值与标准差

约束条件：用户已指定的颜色区域

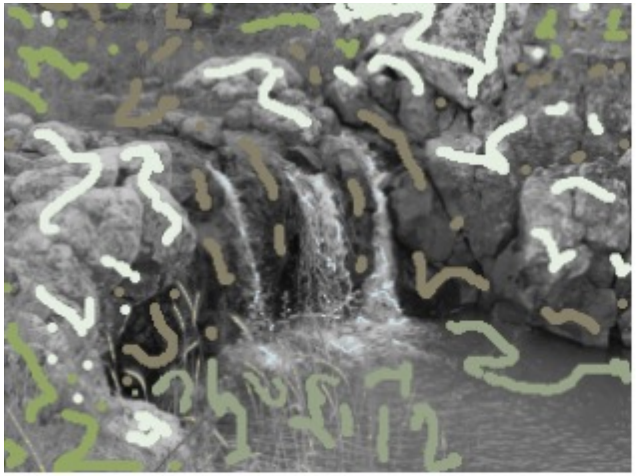
邻域

- 对于单张图片，设置邻域半径即可
- 对于视频序列：

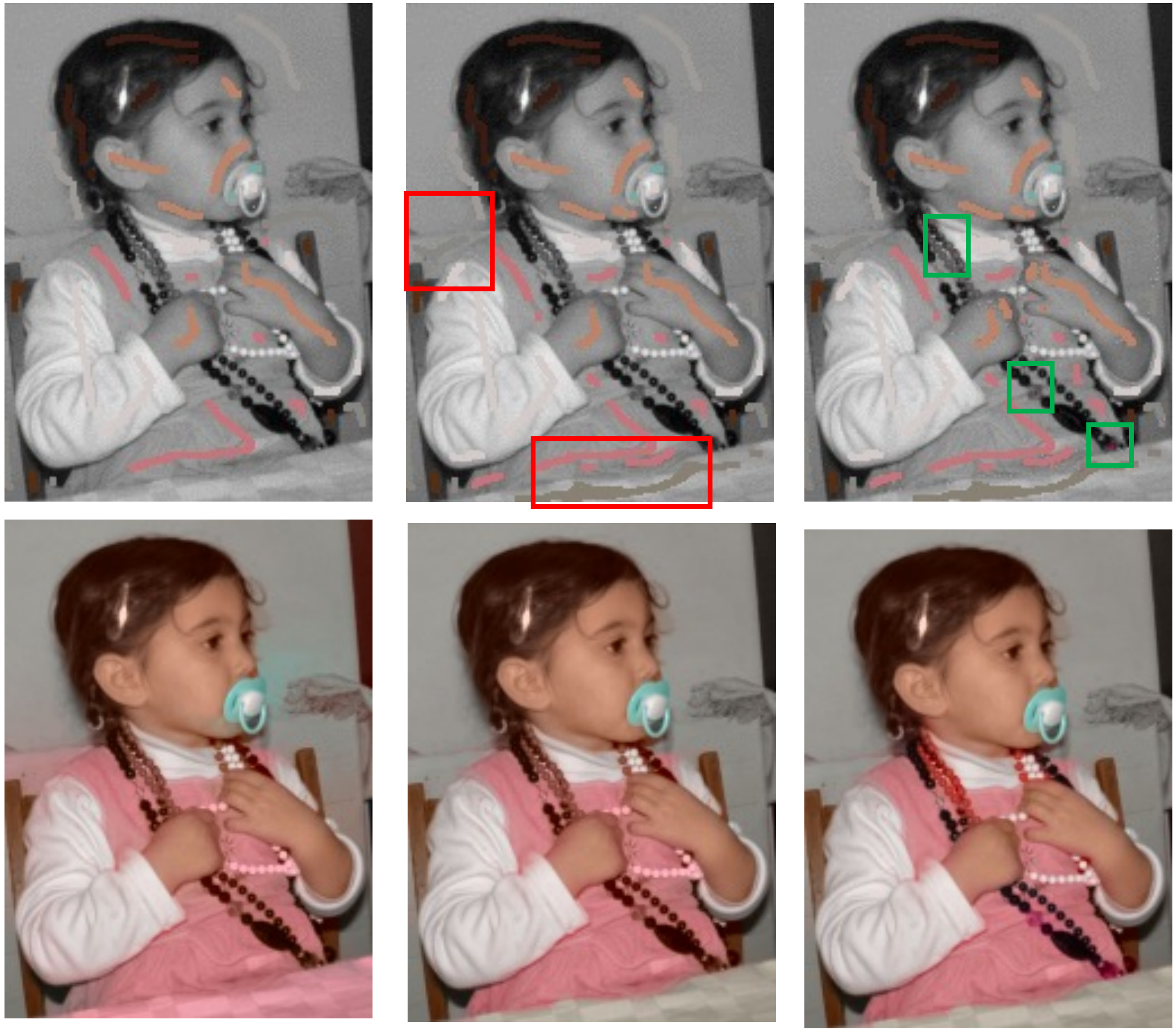
$$\left\|\left(x_0 + v_x(x_0), y_0 + v_y(y_0)\right) - (x_1, y_1)\right\| < T$$

$v_x(x,y), v_y(x,y)$ ：像素(x, y)在t时刻的光流

结果



结果



增加画笔

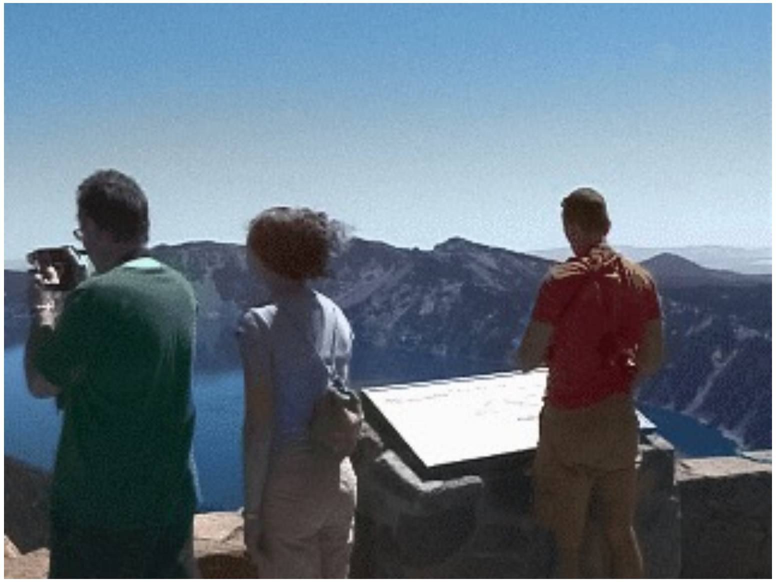
视频上色

原视频（83帧）

画笔（7帧）

视频上色

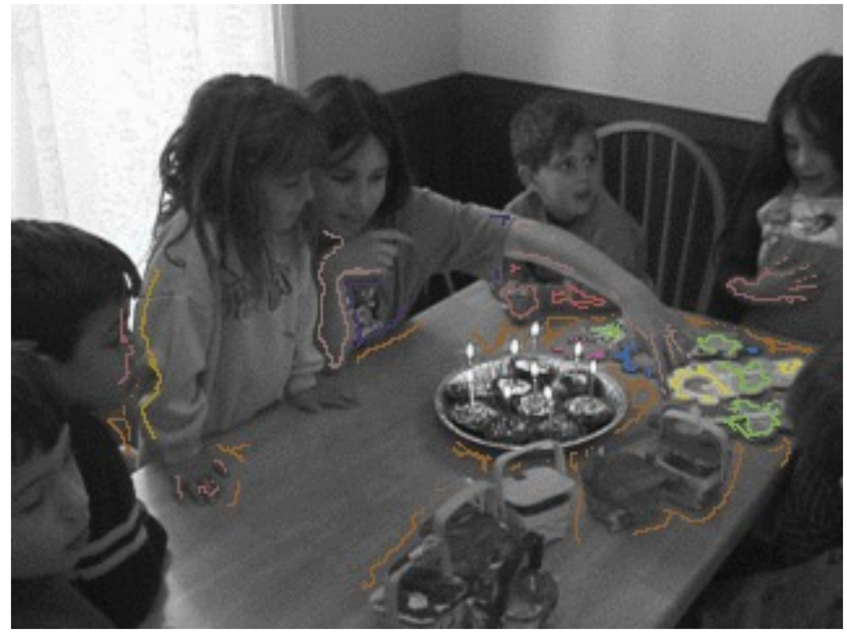
原视频



上色视频

视频上色

原视频（62帧）



画笔（10帧）

视频上色

原视频

上色视频

视频上色

原视频（43帧）

画笔（5帧）

视频上色

原视频

上色视频

视频上色

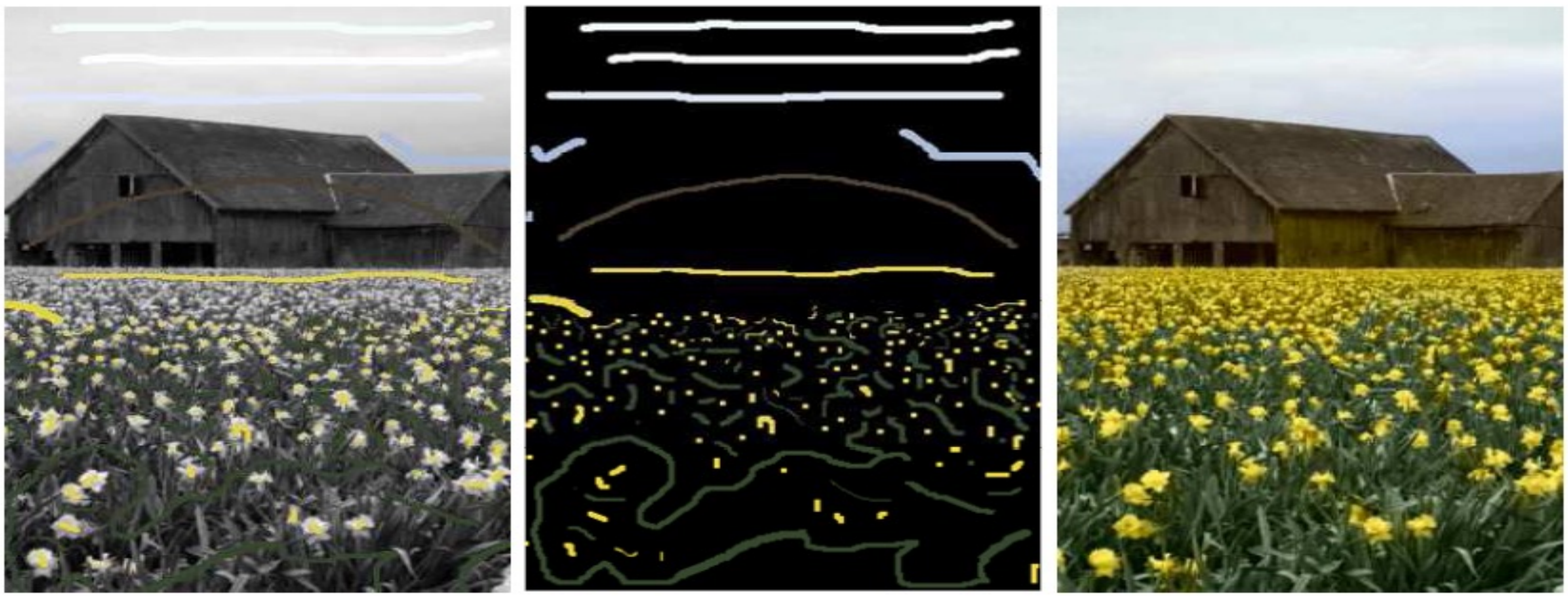
Natural Image Colorization

L. Qing, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu,
H. Shum

Eurographics Symposium on Rendering, 2007

难点

处理highly textured图像需要大量的交互



画笔上色（Levin et al.）

方法

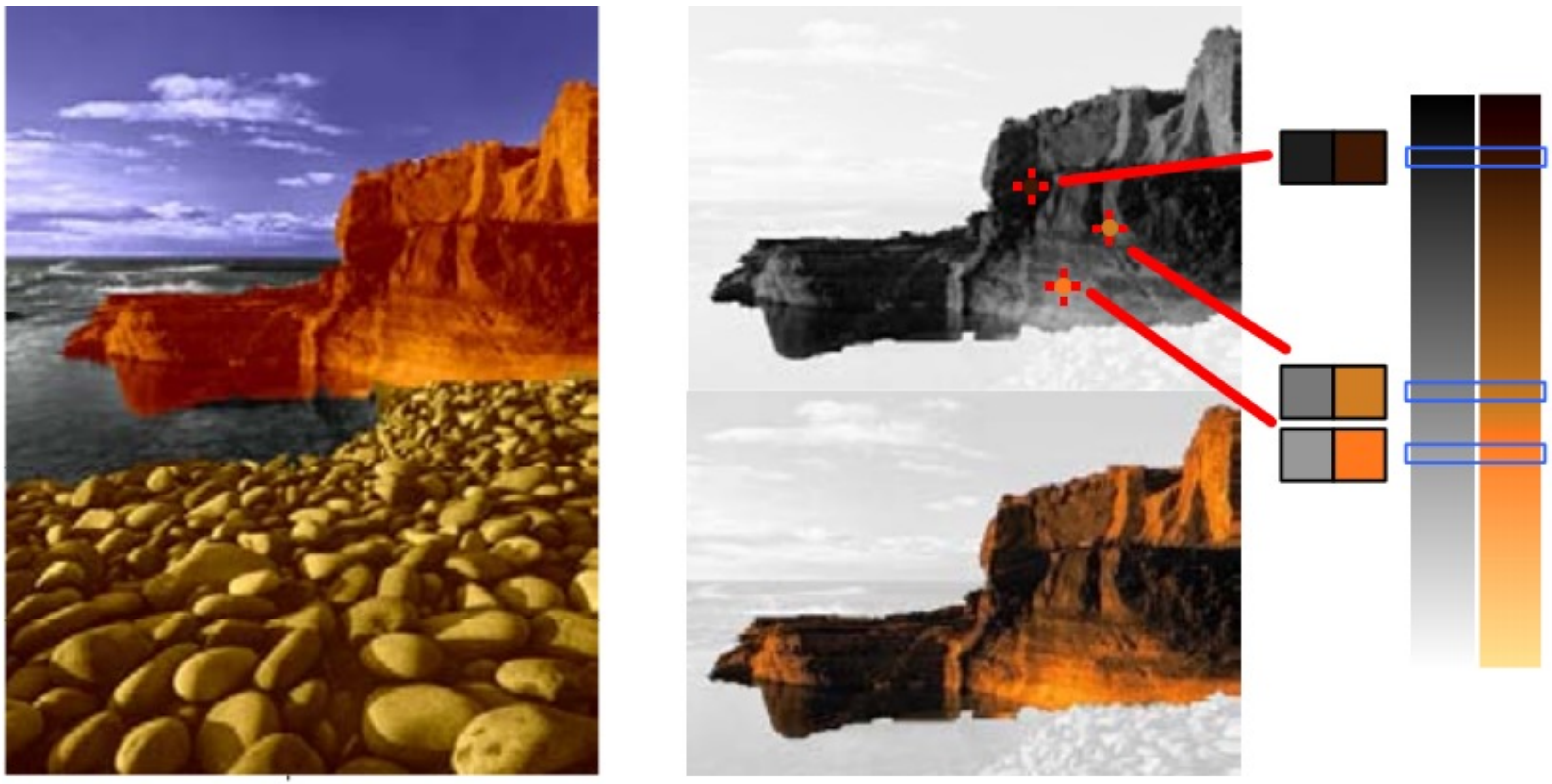
·用户画笔大致指定相同颜色的区域（左图）

·根据画笔，自动分割图像（中图）

·用户为每一块区域的少量像素指定颜色（中图）

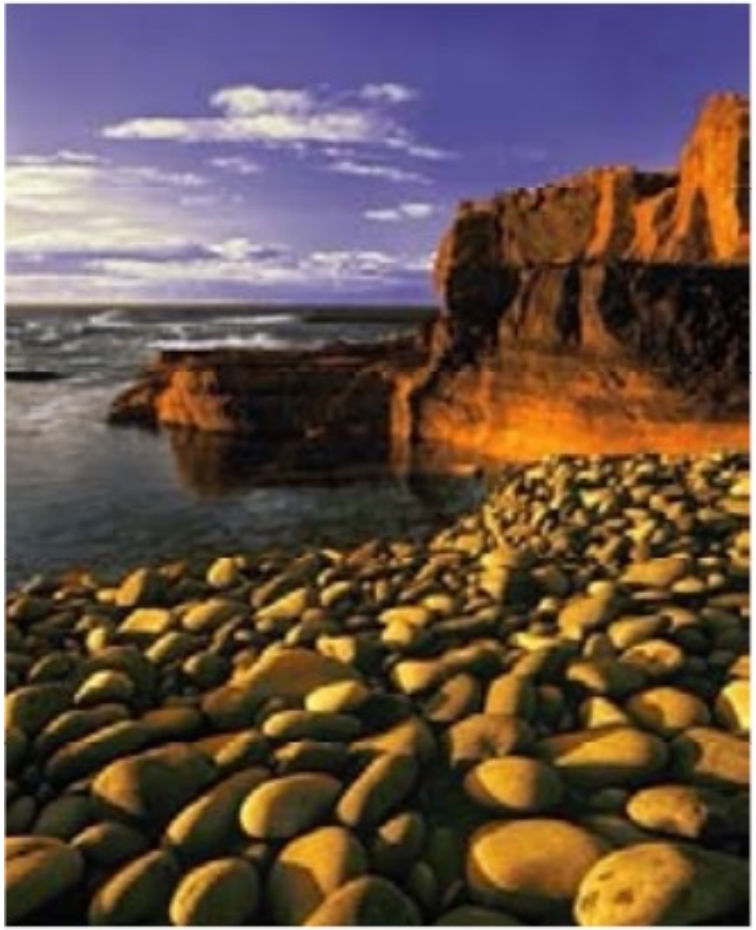
·根据分割结果与用户指定的颜色，给整幅图进行上色（右图）

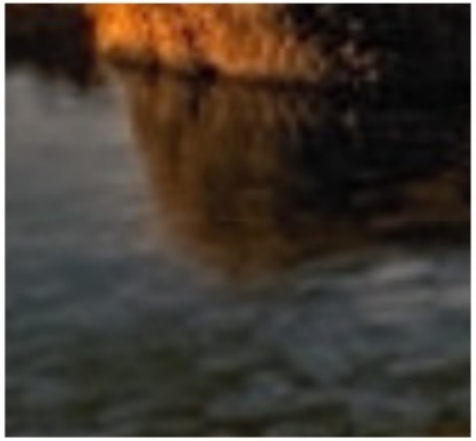
颜色映射



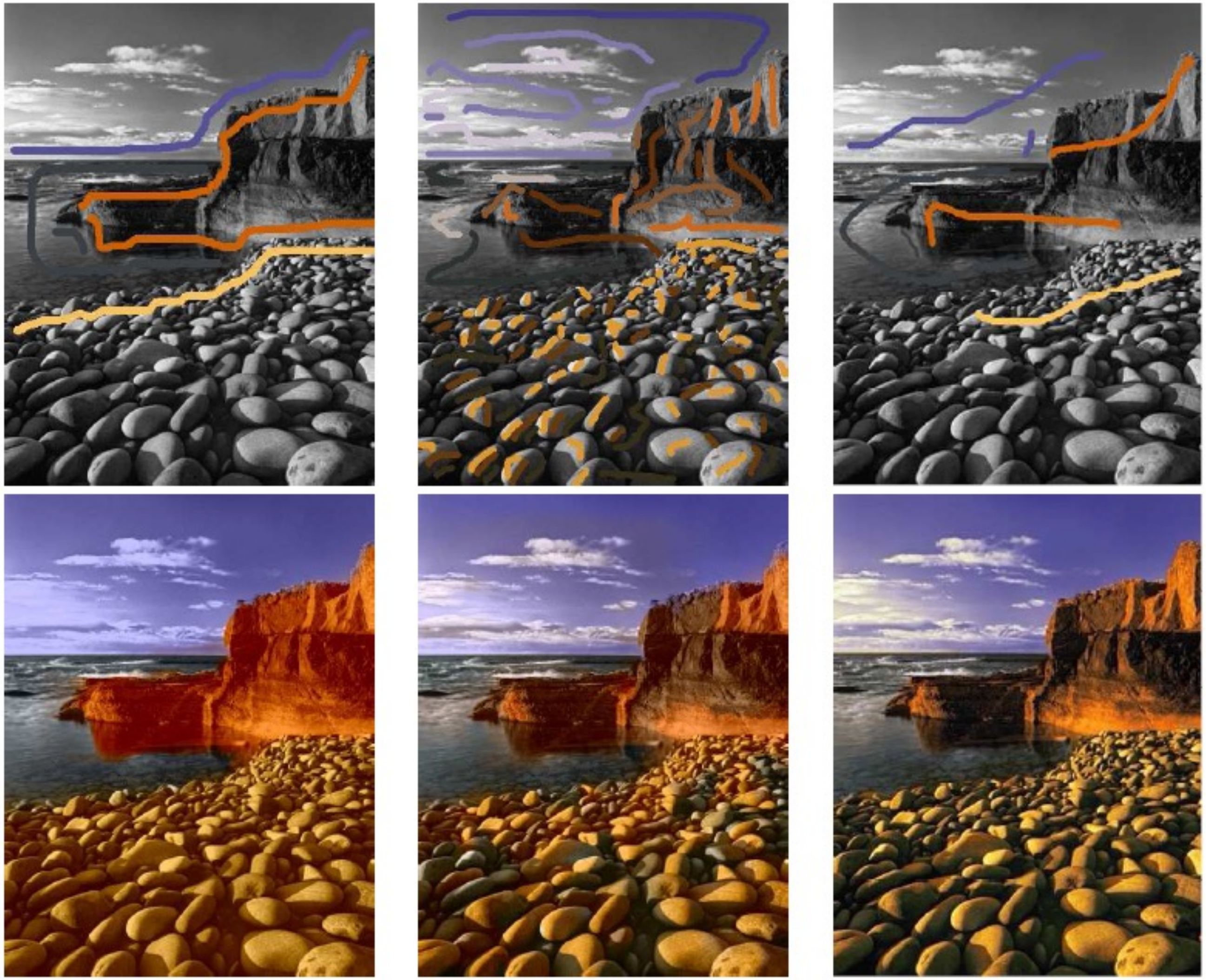
(a) (b) (c) (d)

细节





对比



Levin　　Levin　　新方法

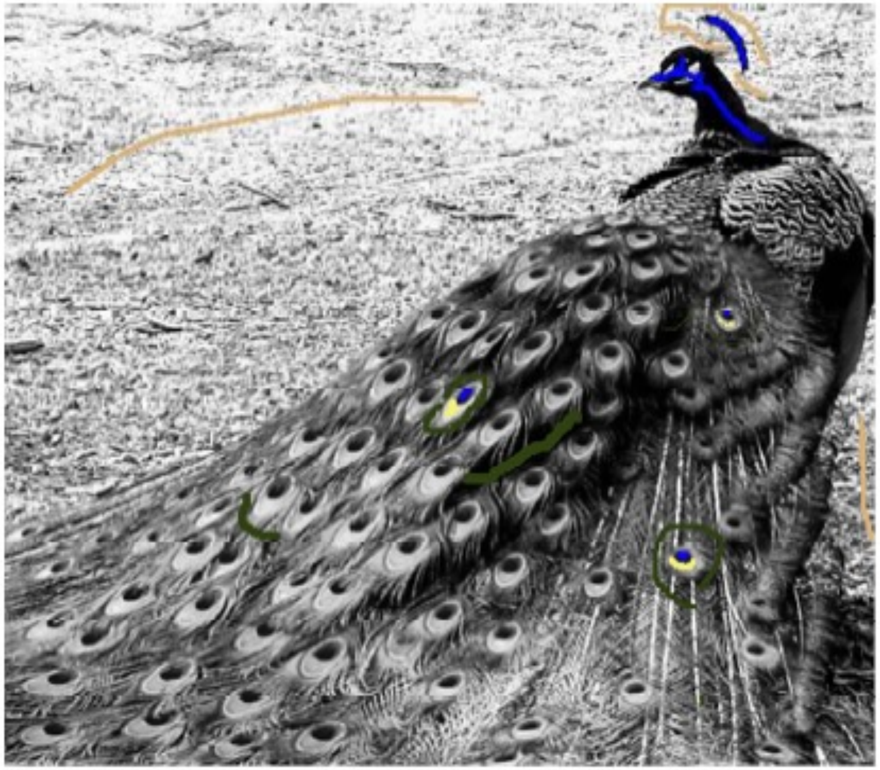
对比

Levin

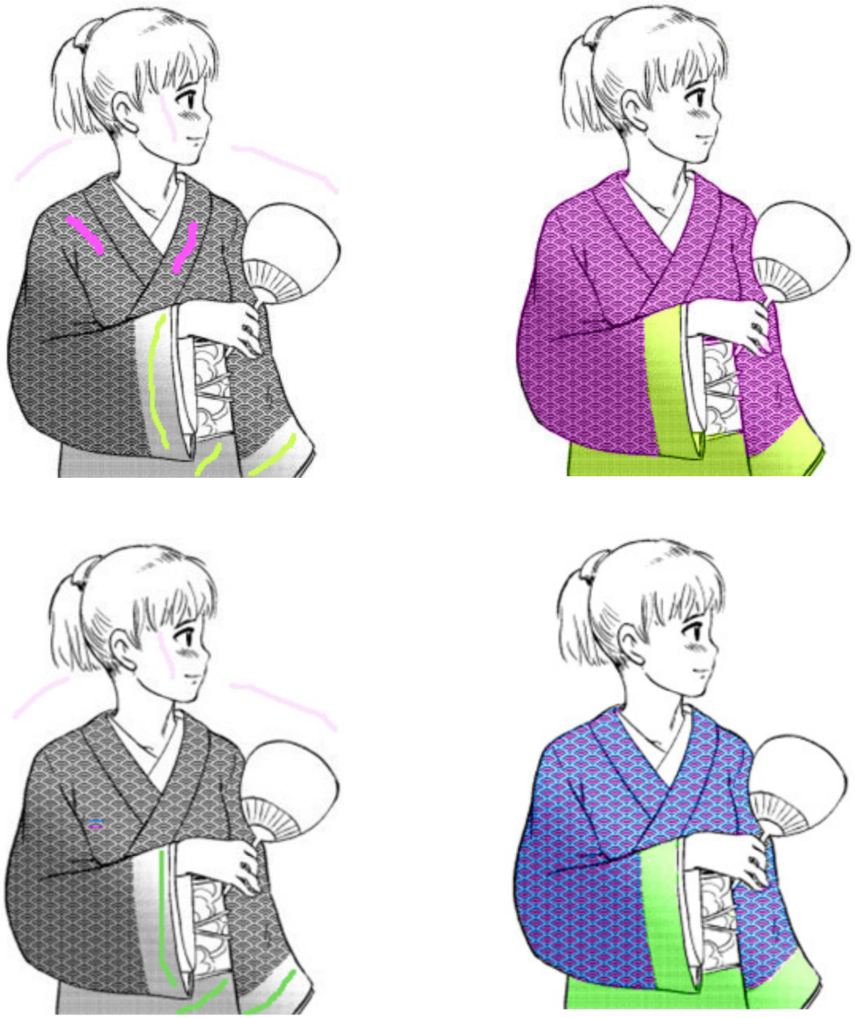

新方法


更多结果

高难度



漫画



上色总结

· 样本上色

- Transferring color to grayscale images (Welsh et al. 2002)
  - ·缺点：空间一致性
- Colorization by example (Irony et al. 2005)
  - ·解决空间一致性、加入分割

· 画笔上色

- Colorization using optimization (Levin et al. 2004)
  - ·缺点：复杂图像画笔需求过多，纹理不连续的图像效果不好
- Natural image colorization (Qing et al. 2007)
  - ·解决图像纹理不连续的上色问题

重上色

·重上色是调整图像颜色的过程，主要用于调整颜色的强度、整张图像的亮度和对比度，使得人更容易感知图像信息。

·可以先将彩色图像转为灰度图像，再对灰度图像重新上色。

转为灰度图像的目标：使得不同物体在灰度图像中仍保留着较强的对比度。

Color2Gray: Salience-Preserving Color Removal

Amy A. Gooch, Sven C . Olsen, Jack Tumblin,
and Bruce Gooch

SIGGRAPH 2005

转换灰度图

彩色图

仅使用亮度

Photoshop Grayscale

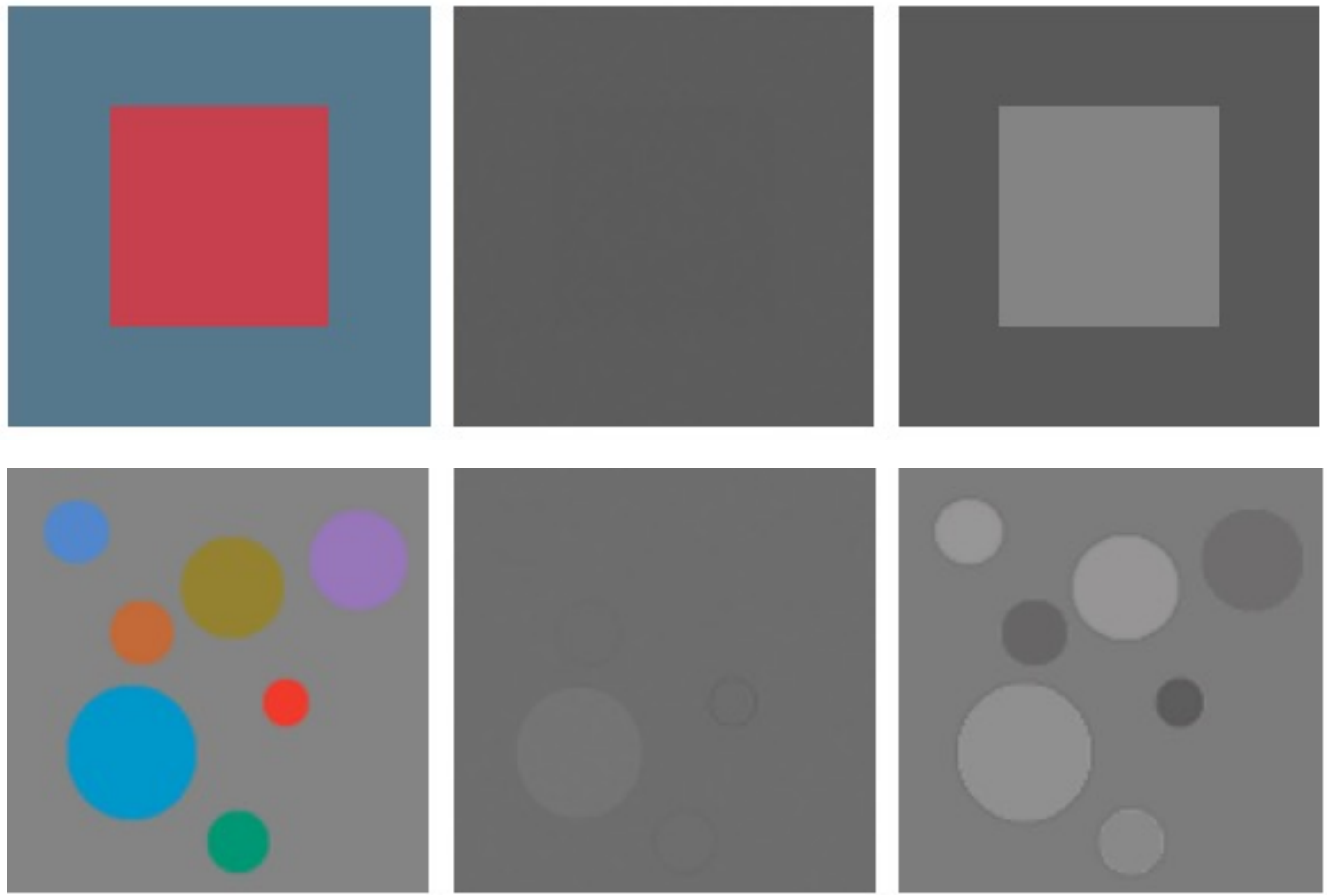
PSGray + Auto Contrast

传统方法致力于增强对比度，图像灰度校正。但是对于灰度值相同的区域没有作用

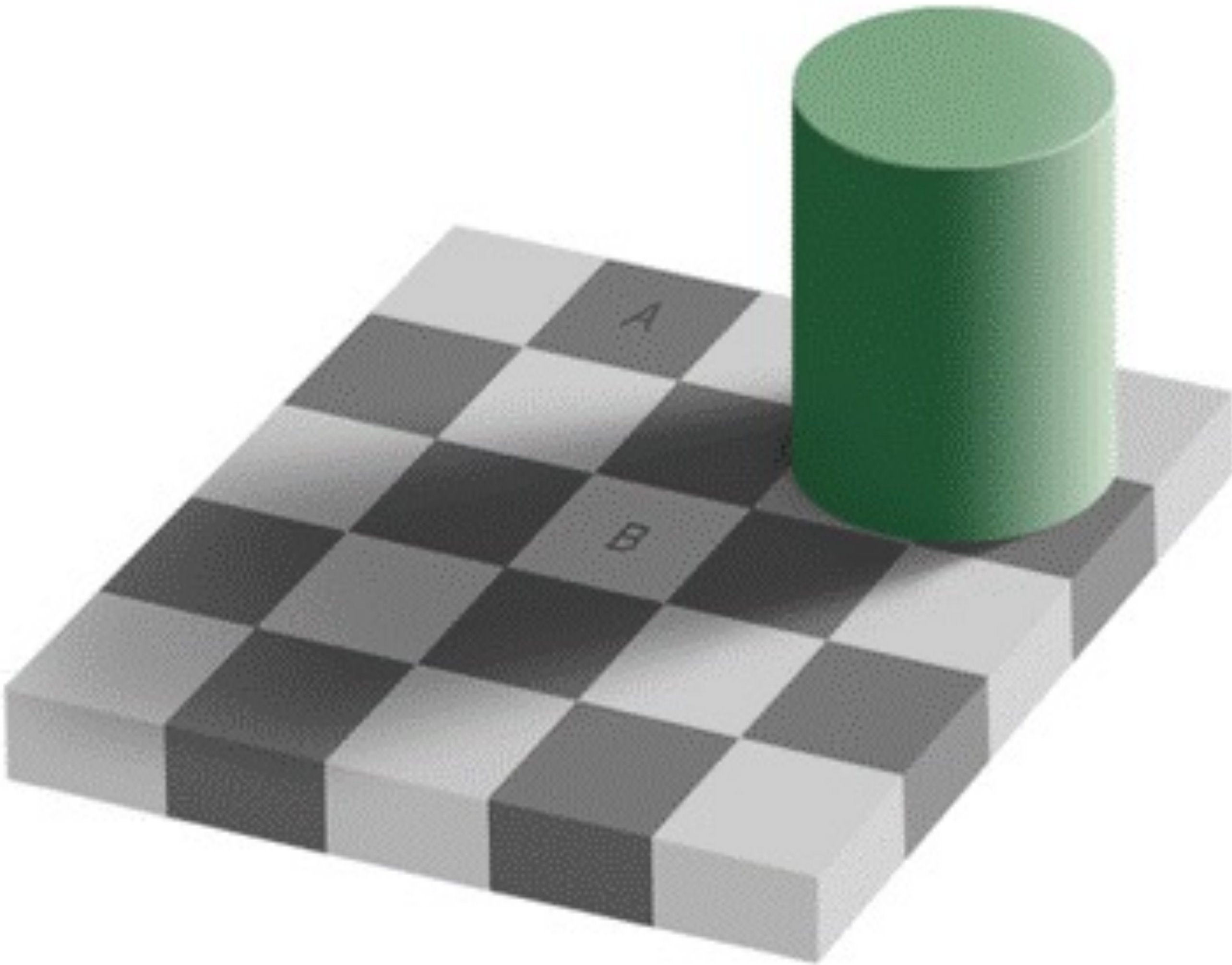
转换灰度图

彩色图

新方法



对于亮度相同的区域，传统方法（中图）会丢失特征，新方法能够保留特征（右图）


A
B

算法概览

·将图像从RGB空间转为Lab空间

·利用L通道初始化灰度图，记为g

·对于彩色图像中像素i的每个邻域内像素j

$$\left.\begin{array}{l}\text{– 计算亮度距离}\\ \text{– 计算色度距离}\end{array}\right\}\ \delta_{ij}$$

·根据亮度和色度距离进行联合优化，确定在灰度图像中的相应灰度值

最小化目标方程：$$f(g) = \sum_{(i,j)\in\mathcal{K}} ((g_i - g_j) - \delta_{ij})^2$$

保持相邻像素视觉上的差别

结果

Photoshop Gray

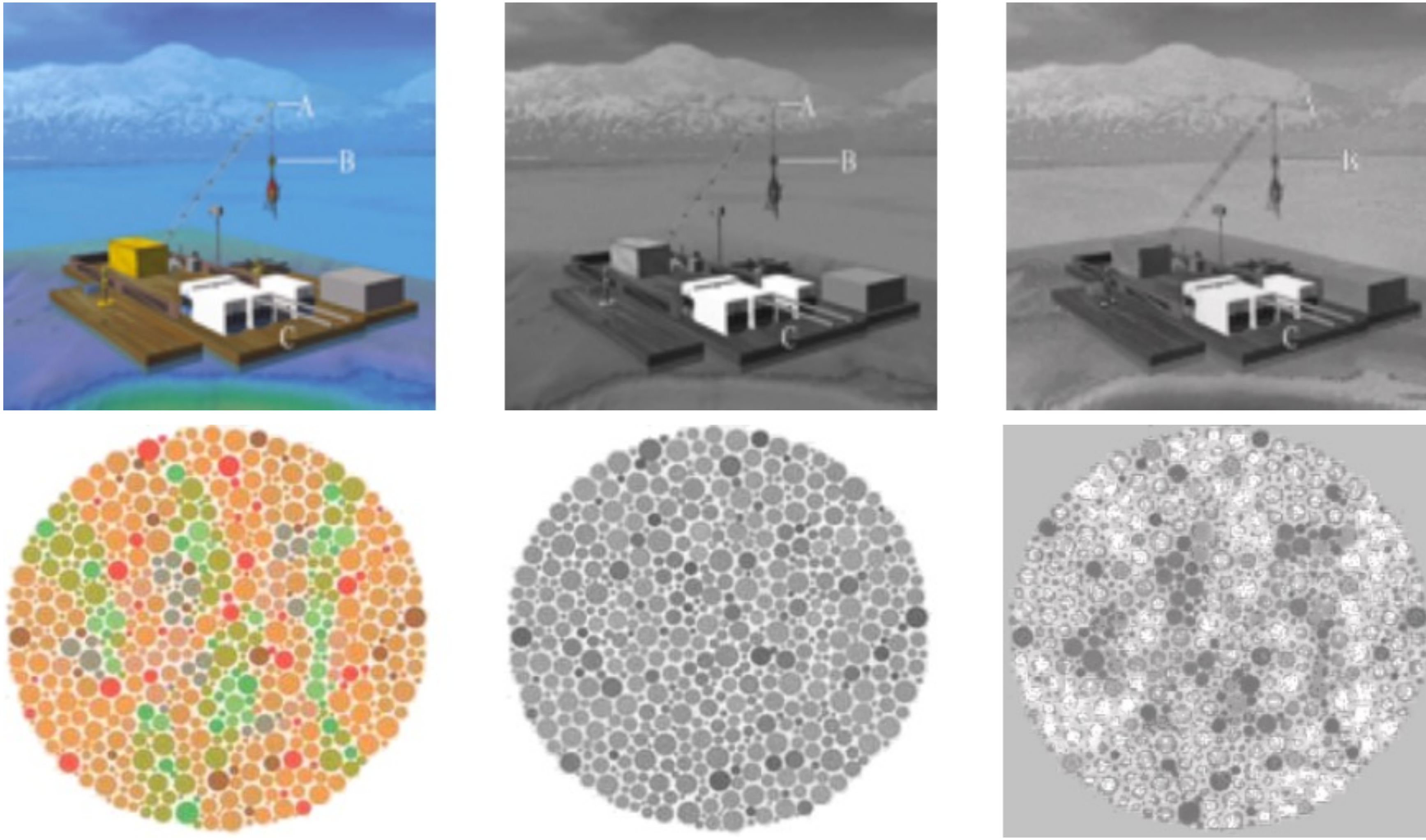
Color2Gray

结果

Photoshop Gray

Color2Gray

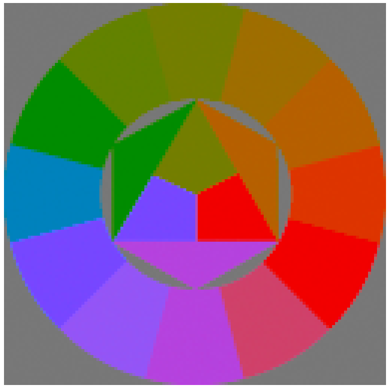
结果

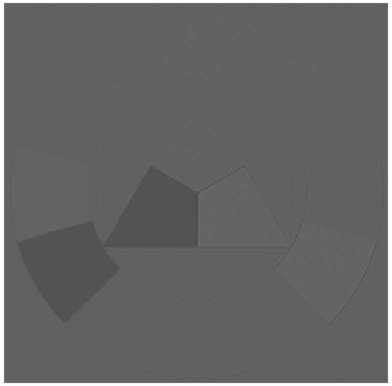


Photoshop Gray

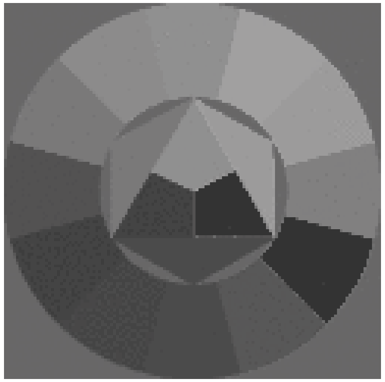
Color2Gray

结果





Photoshop Gray



Color2Gray

结果

Color2Gray

Color2Gray+Color

结果



原图　Photoshop　Color2Gray



结果

原图 | Photoshop | Color2Gray+Color

Color Harmonization

Daniel Cohen-Or, Olga Sorkine,
Ran Gal, Tommer Leyvand, and Ying-Qing Xu

SIGGRAPH 2006

Color Harmony？

Harmonic colors 是一个在人类视觉感知下较为舒适、美观的颜色集合。

不取决于某组特定的颜色，取决于各个颜色的协调搭配。

有经验的艺术家往往凭借经验或直觉来选择合适的颜色集，然后借助某些工具手动交互的调整图片的颜色搭配。

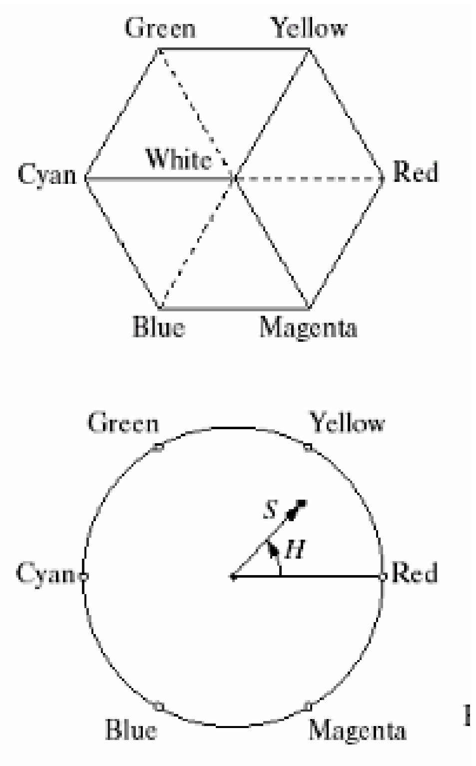
Color Harmony？

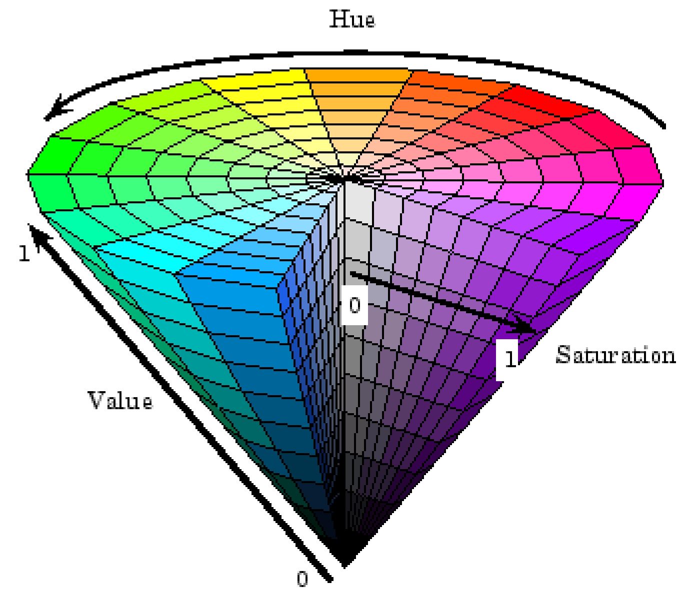
original image

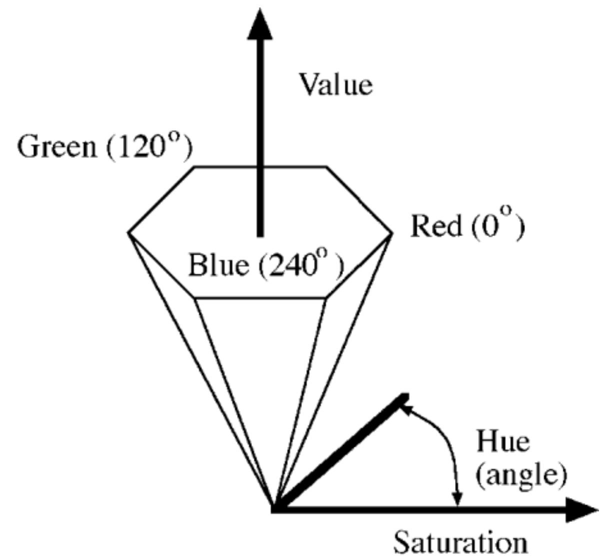
our method（automatically）

Color Harmonization

HSV

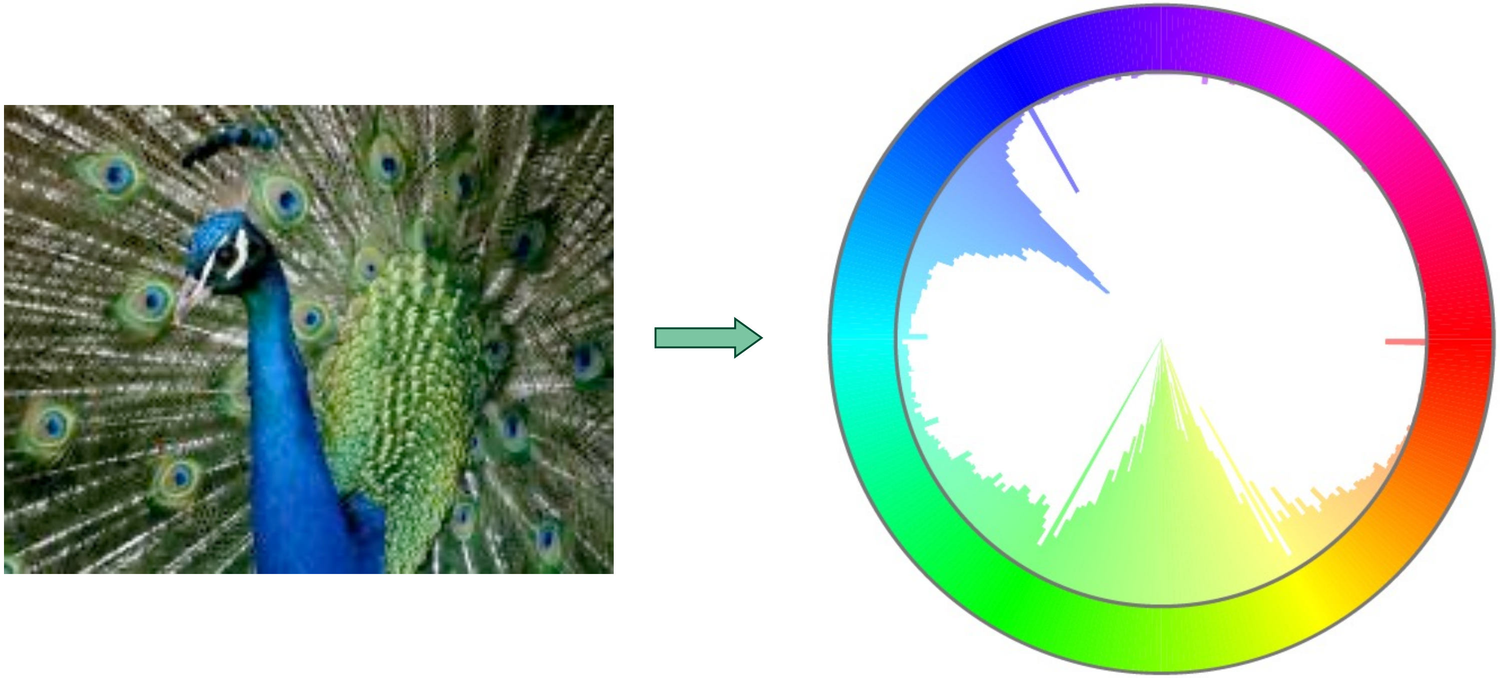




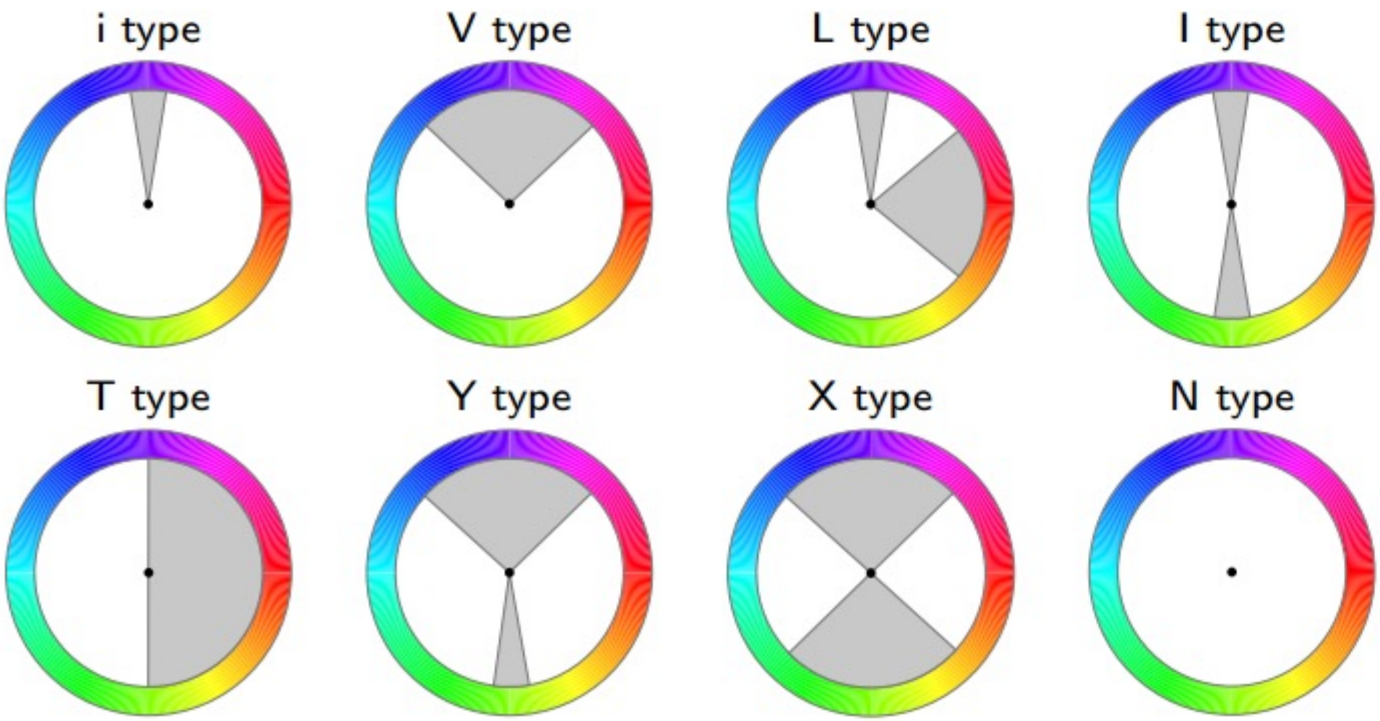


Hue, Saturation, Value

HSV色轮



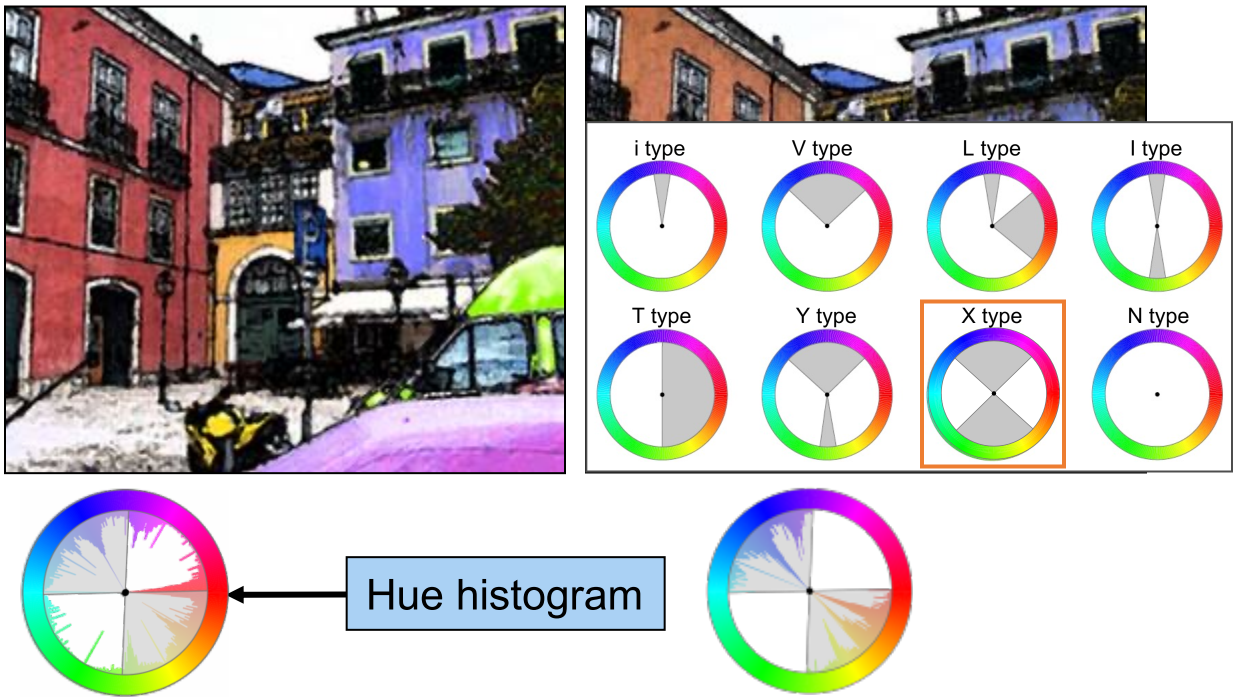
HSV色轮



根据经验得出的符合Harmony的颜色分布

TOKUMARU, M., MURANAKA, N., AND IMANISHI, S. 2002.
Color design support system considering color harmony.

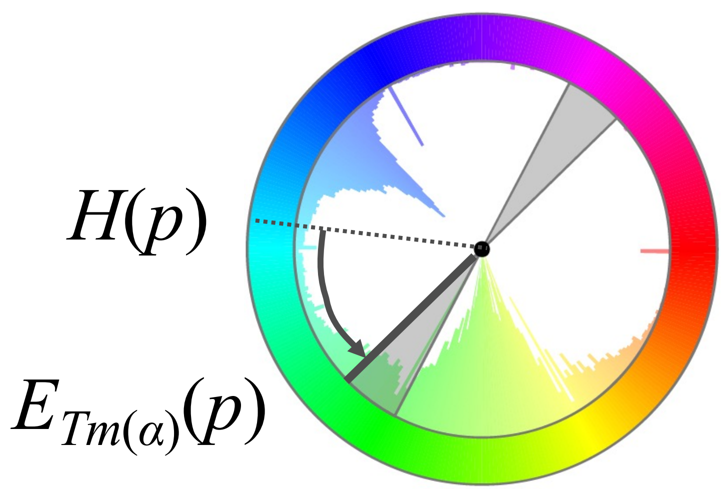
Harmonization



Harmony 方程

像素p色度　最近扇区边缘　像素p饱和度

$$F\left(X,(T_m,\alpha)\right)=\sum_{p\in X}\overbrace{\left\|H(p)-E_{Tm(\alpha)}(p)\right\|}^{\text{弧长}}\cdot S(p)$$

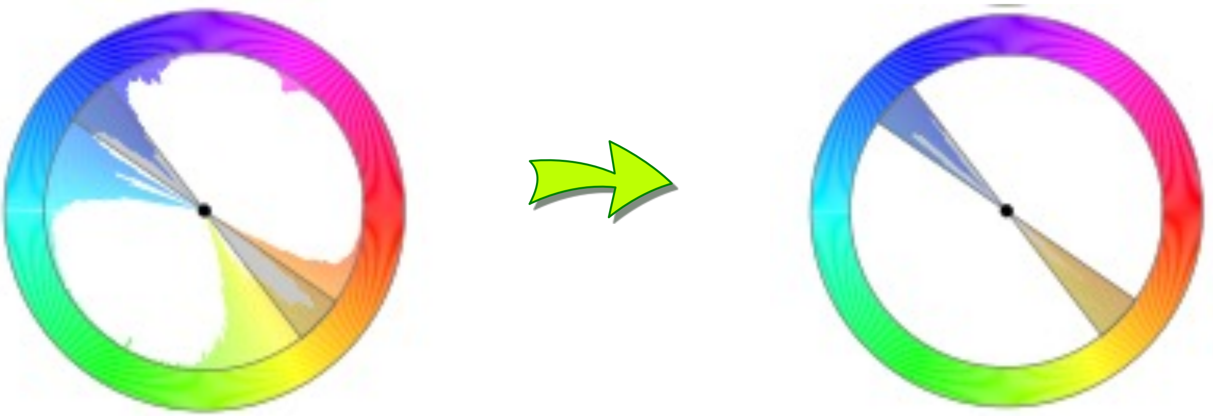


最佳harmony模板

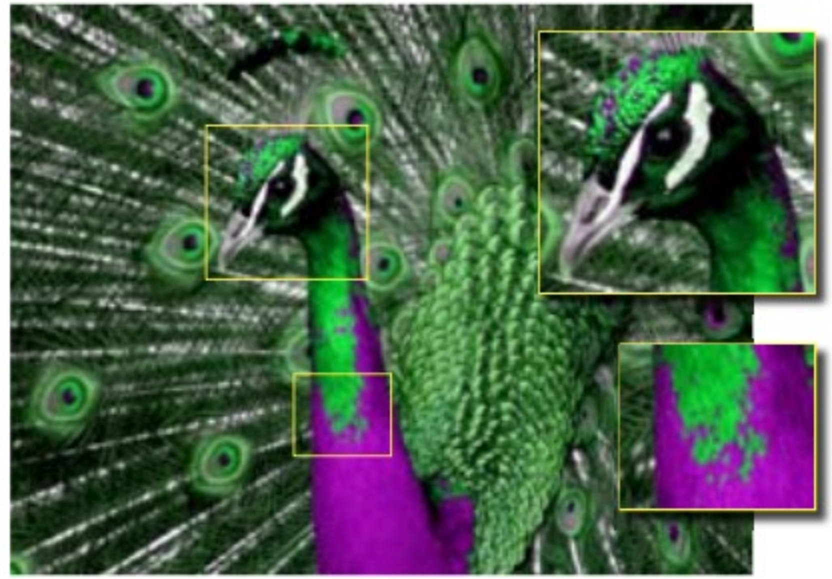
对每个模板T_m，根据$F\left(X,\left(T_m,\alpha\right)\right)$计算最佳角度 α，寻找最佳匹配，之后重新上色。

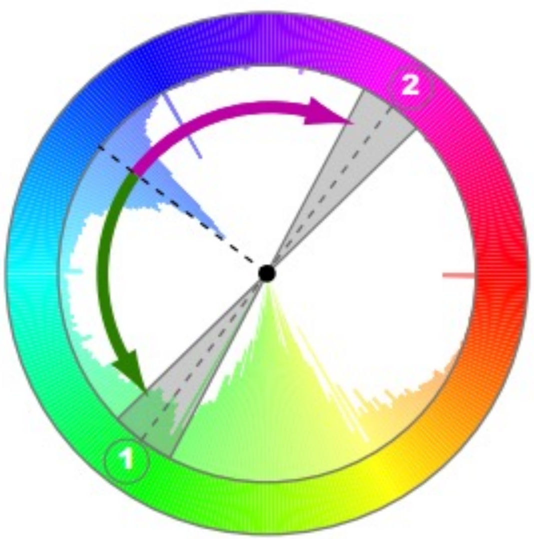
$$\left(T_{m_0},\alpha_0\right)=\underset{(m,\alpha)}{\arg\min}\,F\left(X,\left(T_m,\alpha\right)\right)$$

结果



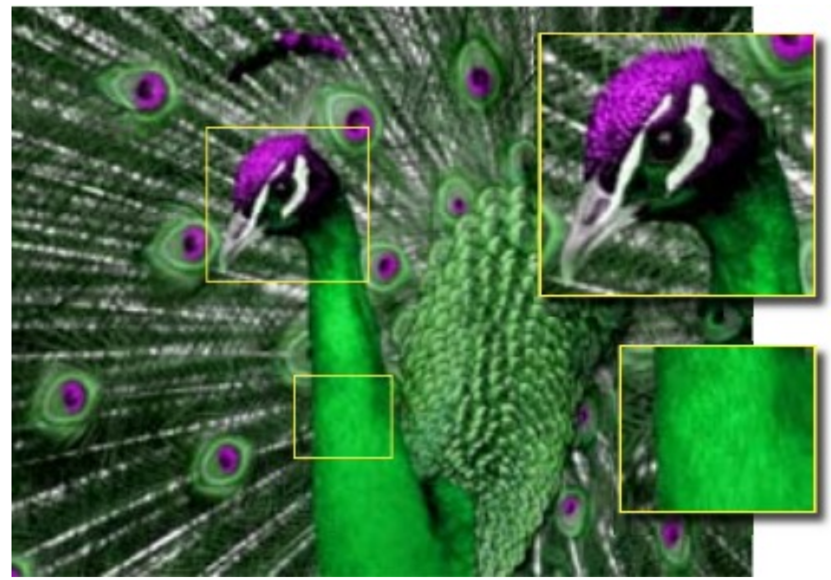
问题





不连续，蓝色区域的像素有的shift到1，有的shift到 2

Graph-cut

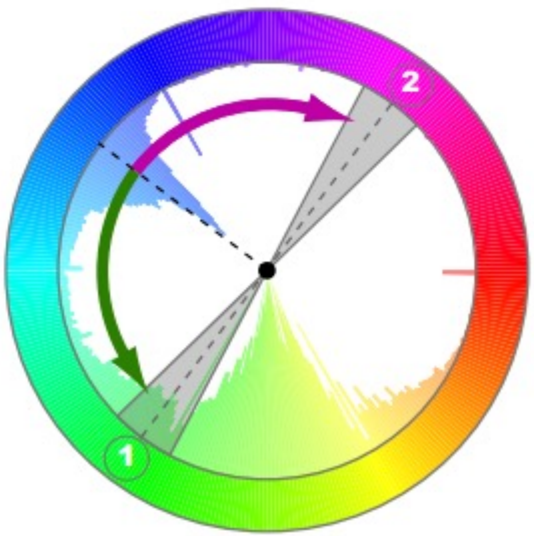


Graph-cut 优化

Graph-cut

$$E(V) = \lambda E_1(V) + E_2(V)$$

$$V = \{v(p_1), \ldots, v(p_{|\Omega|})\}$$



Graph-cut

$$E(V) = \lambda E_1(V) + E_2(V)$$

$$E_1(V) = \sum_{i=1}^{|\Omega|} \| H(p_i) - H(v(p_i)) \| \cdot S(p_i)$$

$$E_2(V) = \sum_{\{p,q\} \in N} \delta(v(p), v(q)) \cdot S_{\max}(p,q) \cdot \| H(p) - H(q) \|^{-1}$$

结果

原图　　　　　　　　　　选择不同模板进行重上色

结果

(d) original

(e) harmonized foreground

根据国旗的配色类型，重上色图片

总结：

- 上色方法
    - 利用样本上色
    - 交互式上色

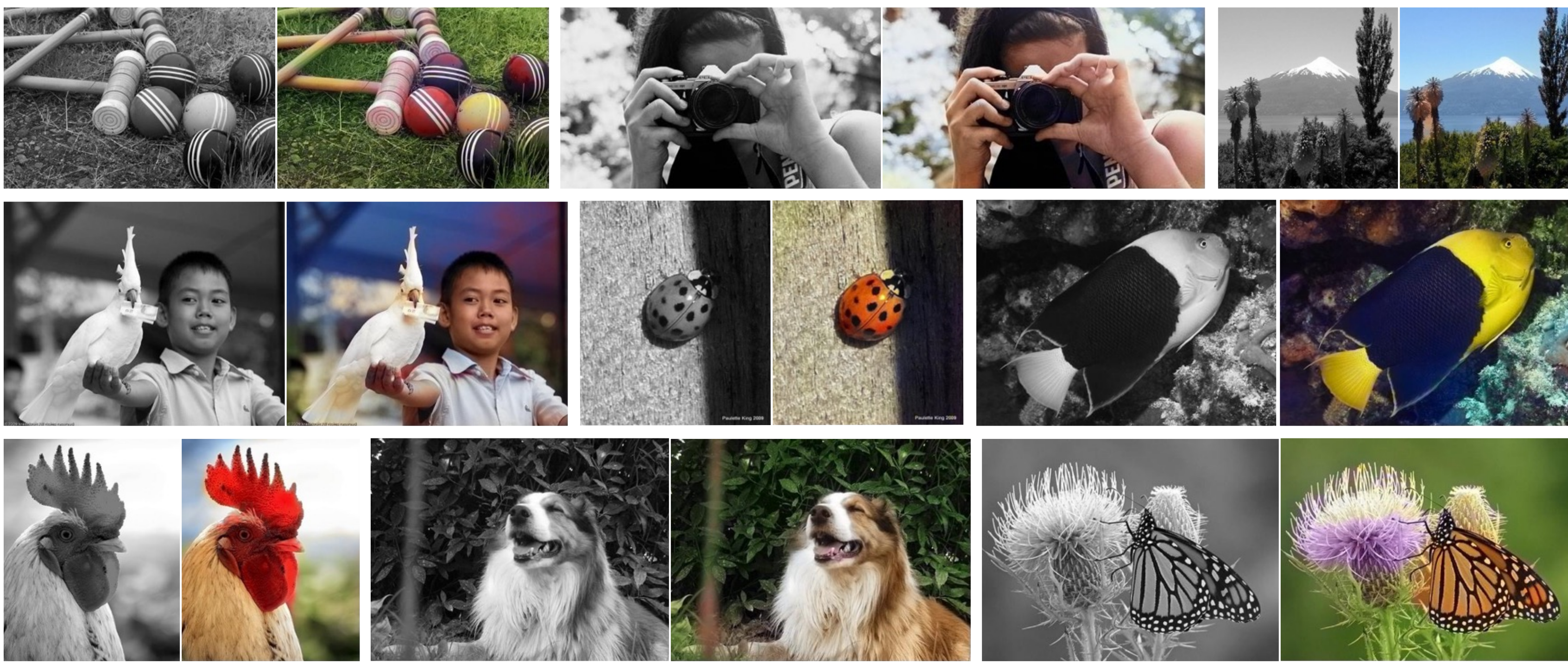
- 关键技术
    - 如何匹配
    - 如何分割
    - 如何保证空间时间一致性

The modern approach?

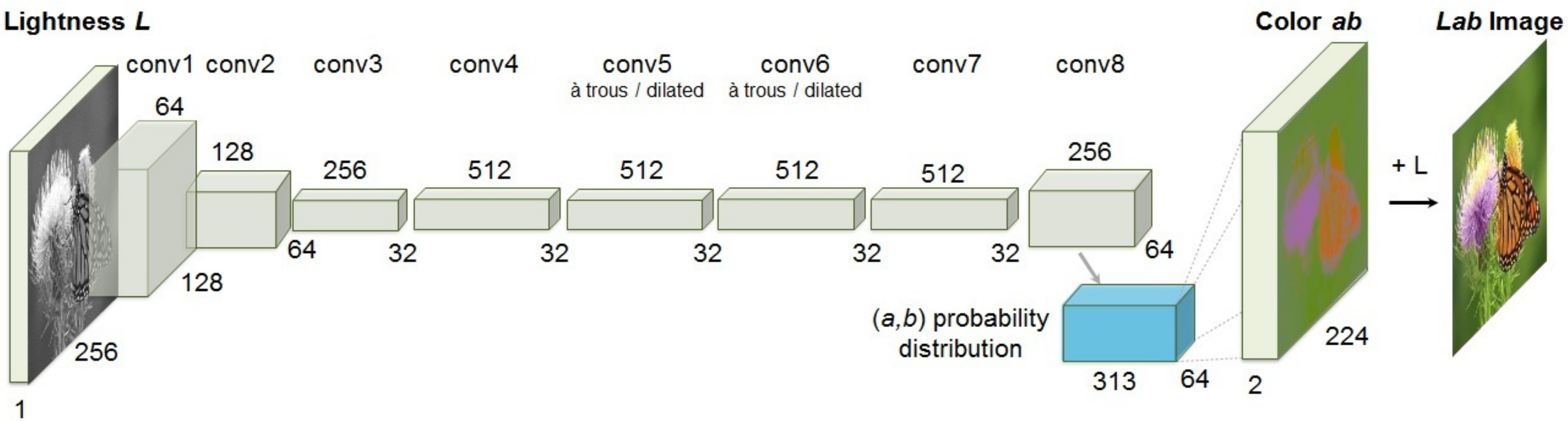
Colorful Image Colorization

Richard Zhang, Phillip Isola, Alexei A. Efros
{rich.zhang,isola,efros}@eecs.berkeley.edu

University of California, Berkeley



Convolutional Neural Networks



http://richzhang.github.io/colorization/

Loss function for image synthesis

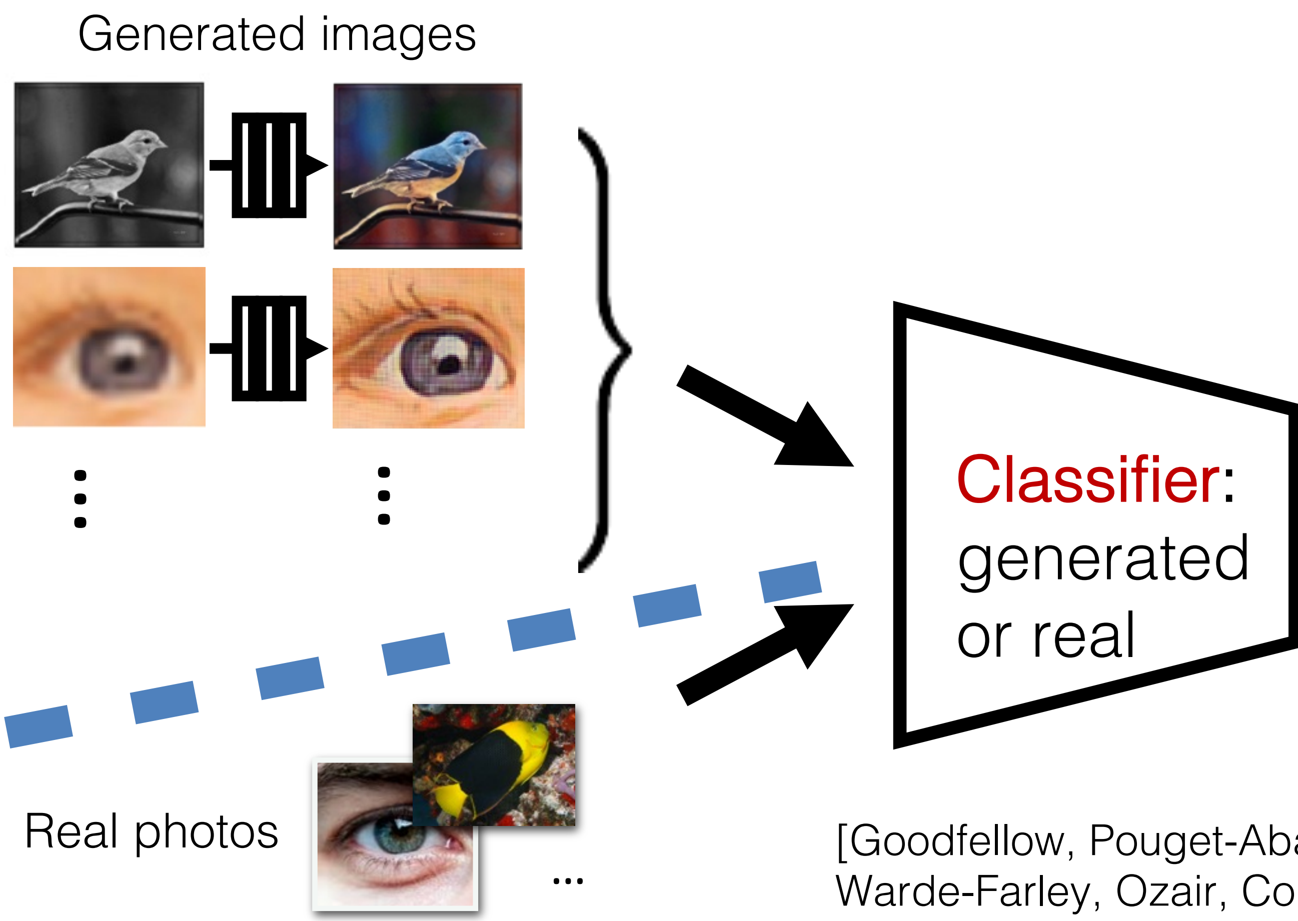
- Reconstruction loss

$$L(\Theta) = ||F(X;\Theta) - Y||^2$$

- Problem with reconstruction loss
  - Cannot handle the case with multiple solutions
  - Cannot measure if an image is realistic

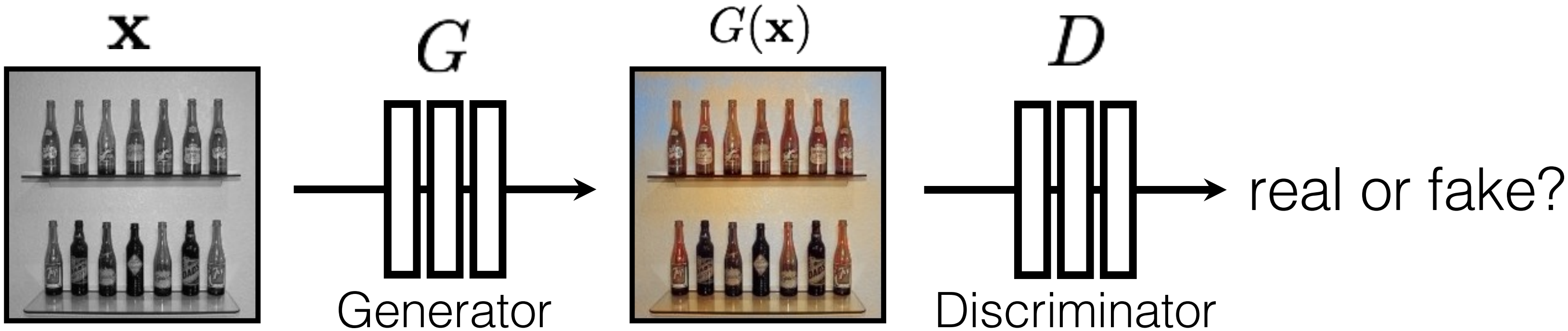
- Can we have a loss function that measures if an image is real?

Generative Adversarial Network (GAN)



[Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, Bengio 2014]

Generative Adversarial Network (GAN)

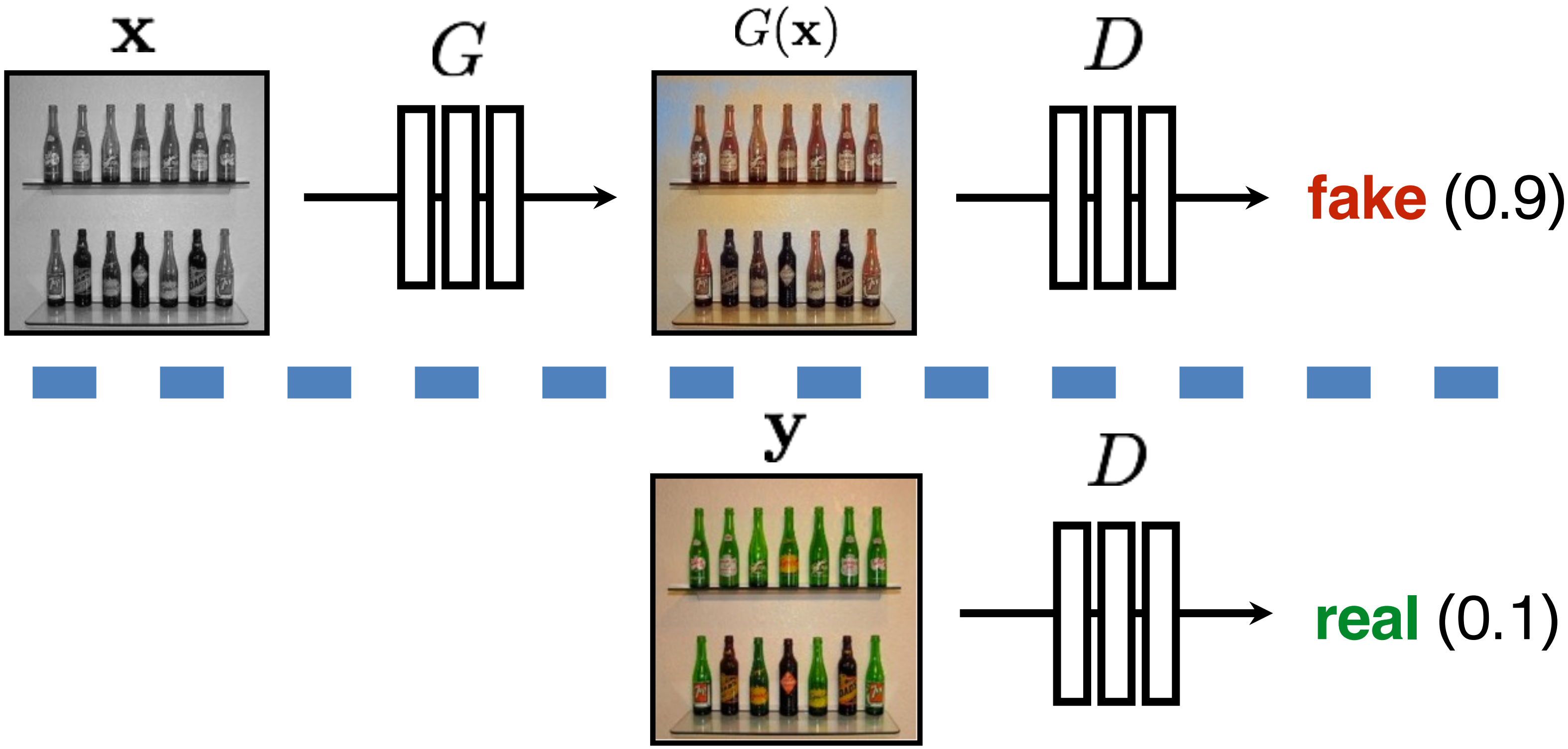


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

[Goodfellow et al., 2014]

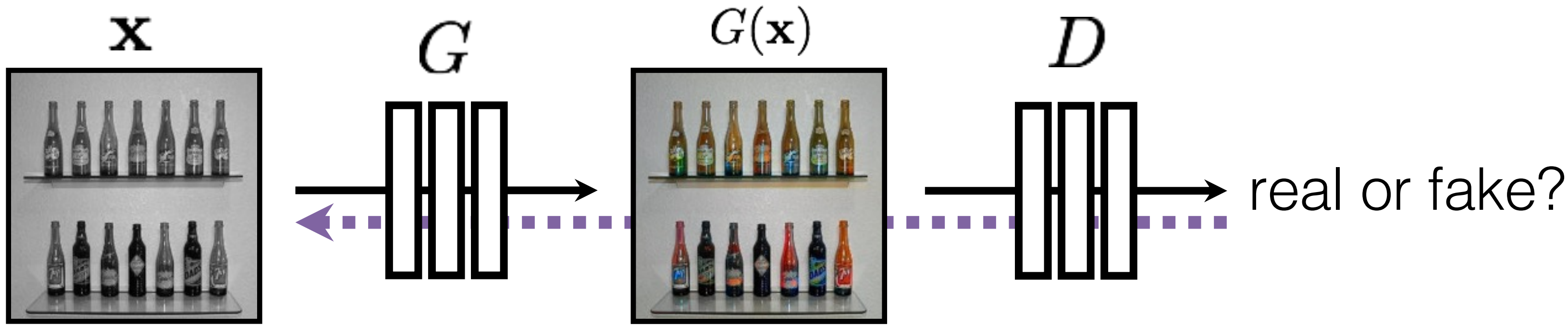
Generative Adversarial Network (GAN)



$$\arg\max_{D} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \boxed{\log D(G(\mathbf{x}))} \;+\; \boxed{\log(1 - D(\mathbf{y}))} \;]$$

[Goodfellow et al., 2014]

Generative Adversarial Network (GAN)

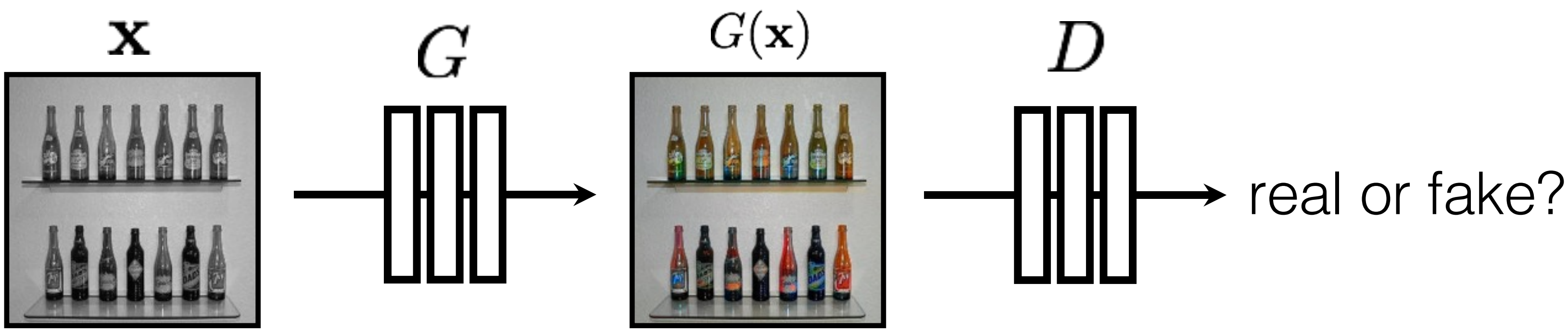


G tries to synthesize fake images that ***fool*** **D**:

$$\arg\boxed{\min_{G}}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \ + \ \log(1 - D(\mathbf{y}))\ ]$$

[Goodfellow et al., 2014]

Generative Adversarial Network (GAN)

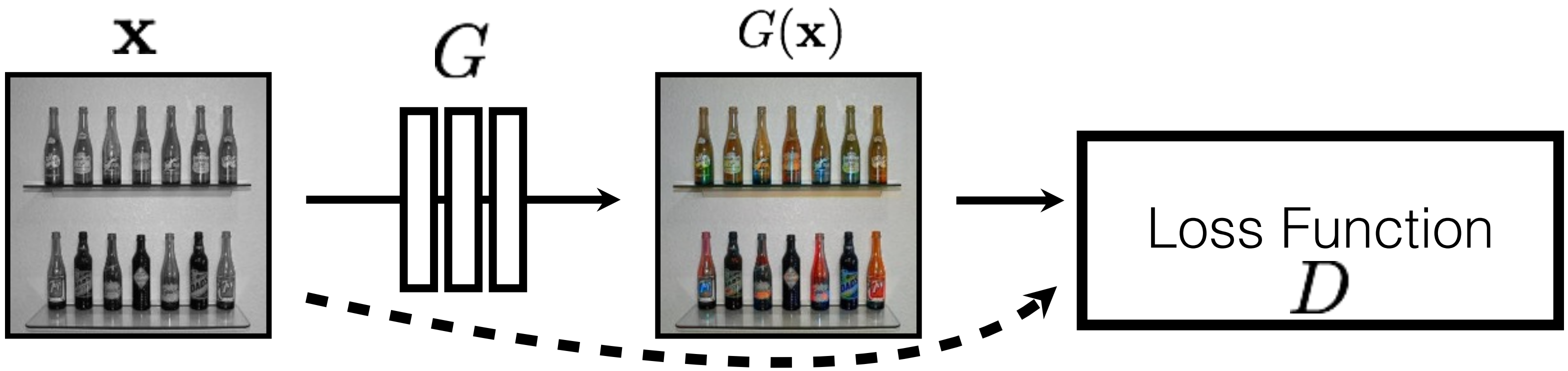


G tries to synthesize fake images that ***fool*** the ***best*** **D**:

$$\arg \boxed{\min_{G}} \boxed{\max_{D}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \ + \ \log(1 - D(\mathbf{y}))\ ]$$

[Goodfellow et al., 2014]

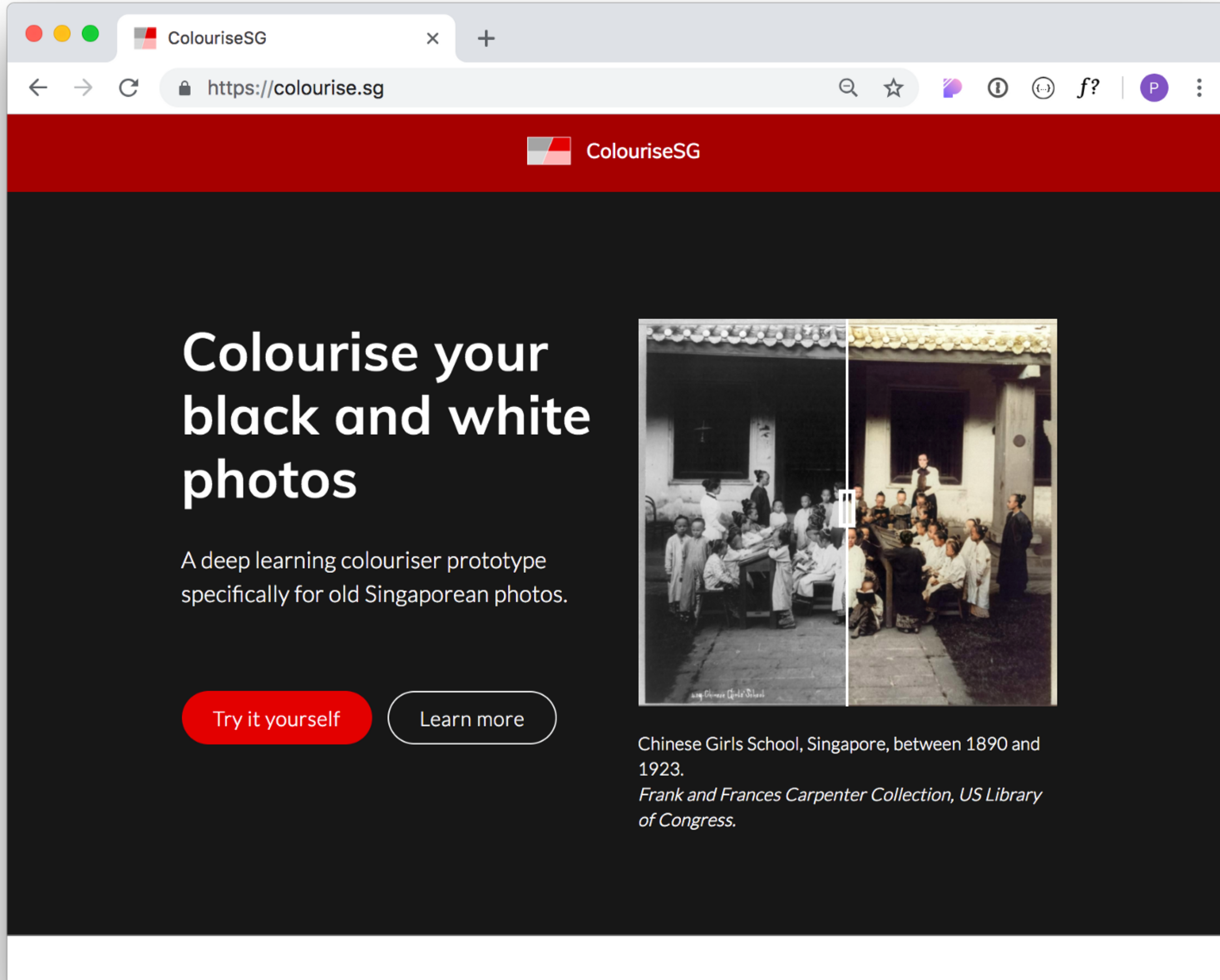
Generative Adversarial Network (GAN)



$$\arg \boxed{\min_{G}}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x}))\ ]$$

- D can be viewed as a loss function to train G
  - Called adversarial loss
  - Learned instead of being hand-designed
  - Can be applied to any image synthesis tasks

[Goodfellow et al., 2014]
[Isola et al., 2017]


ColouriseSG
https://colourise.sg
ColouriseSG
Colourise your black and white photos
A deep learning colouriser prototype specifically for old Singaporean photos.
Try it yourself
Learn more
Chinese Girls School, Singapore, between 1890 and 1923.
Frank and Frances Carpenter Collection, US Library of Congress.

Real-Time User-Guided Image Colorization with Learned Deep Priors

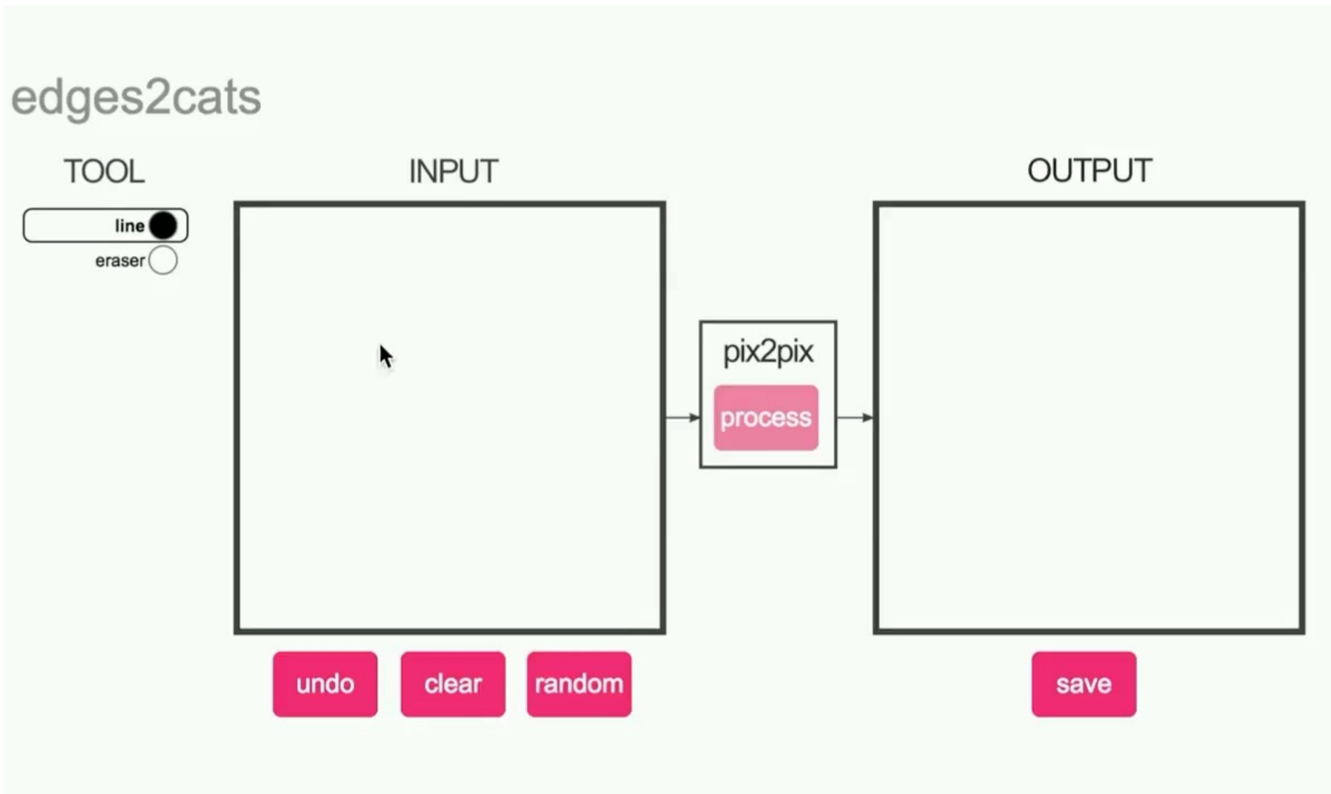
Richard Zhang* Jun-Yan Zhu* Phillip Isola
Xinyang Geng Angela S. Lin Tianhe Yu Alexei A. Efros

More image sysnthesis tasks

- Image to Image Translation

More image sysnthesis tasks

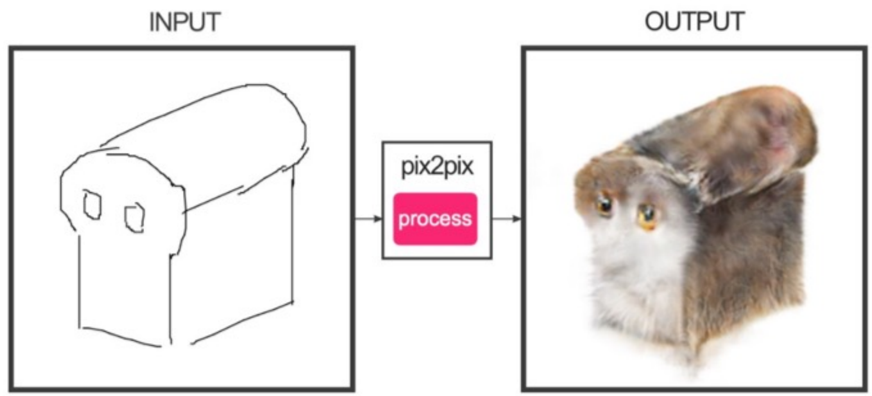
- Image to Image Translation



https://affinelayer.com/pixsrv/

More image sysnthesis tasks

- Image to Image Translation

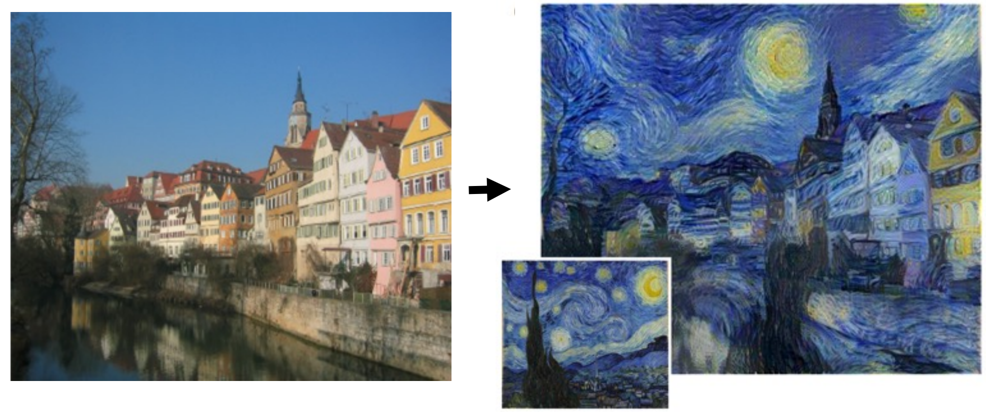


Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

More image sysnthesis tasks

- Style transfer



[Gatys et al. 2016, …]

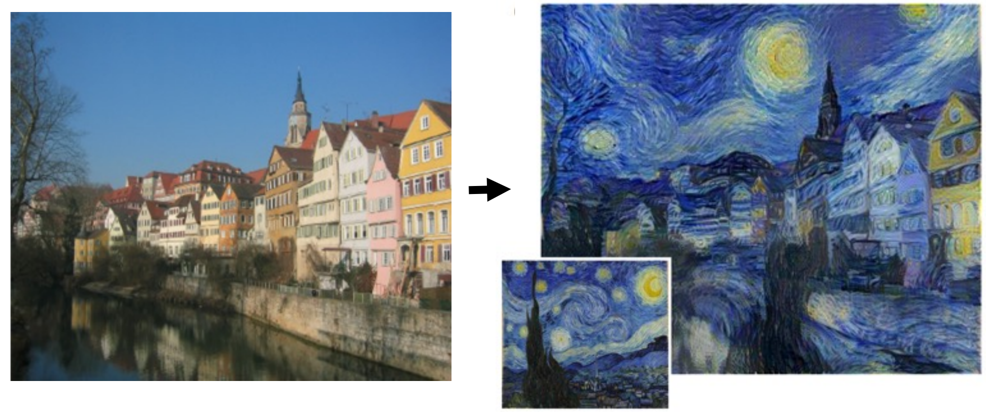
- Text-to-Photo

"this small bird has a pink breast and crown…"

[Reed et al. 2014, …]

More image sysnthesis tasks

- Style transfer



[Gatys et al. 2016, …]

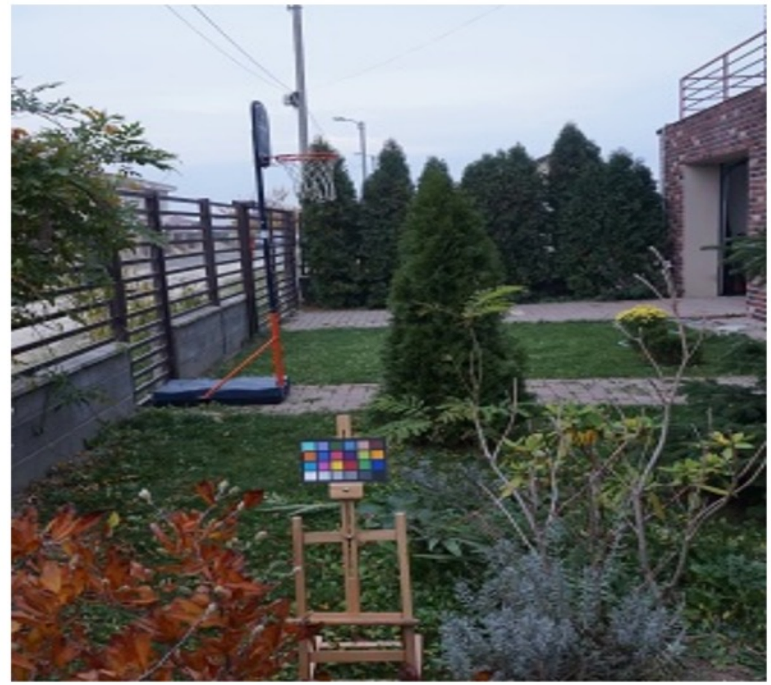
- Text-to-Photo

"this small bird has a pink breast and crown…" →

[Reed et al. 2014, …]

More image sysnthesis tasks

- Image dehazing



Cycle-Dehaze: Enhanced CycleGAN for Single Image Dehazing. CVPRW 2018
Deniz Engin∗ Anıl Genc∗, Hazım Kemal Ekenel

More image sysnthesis tasks

- Customized gaming

Battle royale games

Fortnite Input

+

PUBG Style

=

Final result

Cycle-GAN

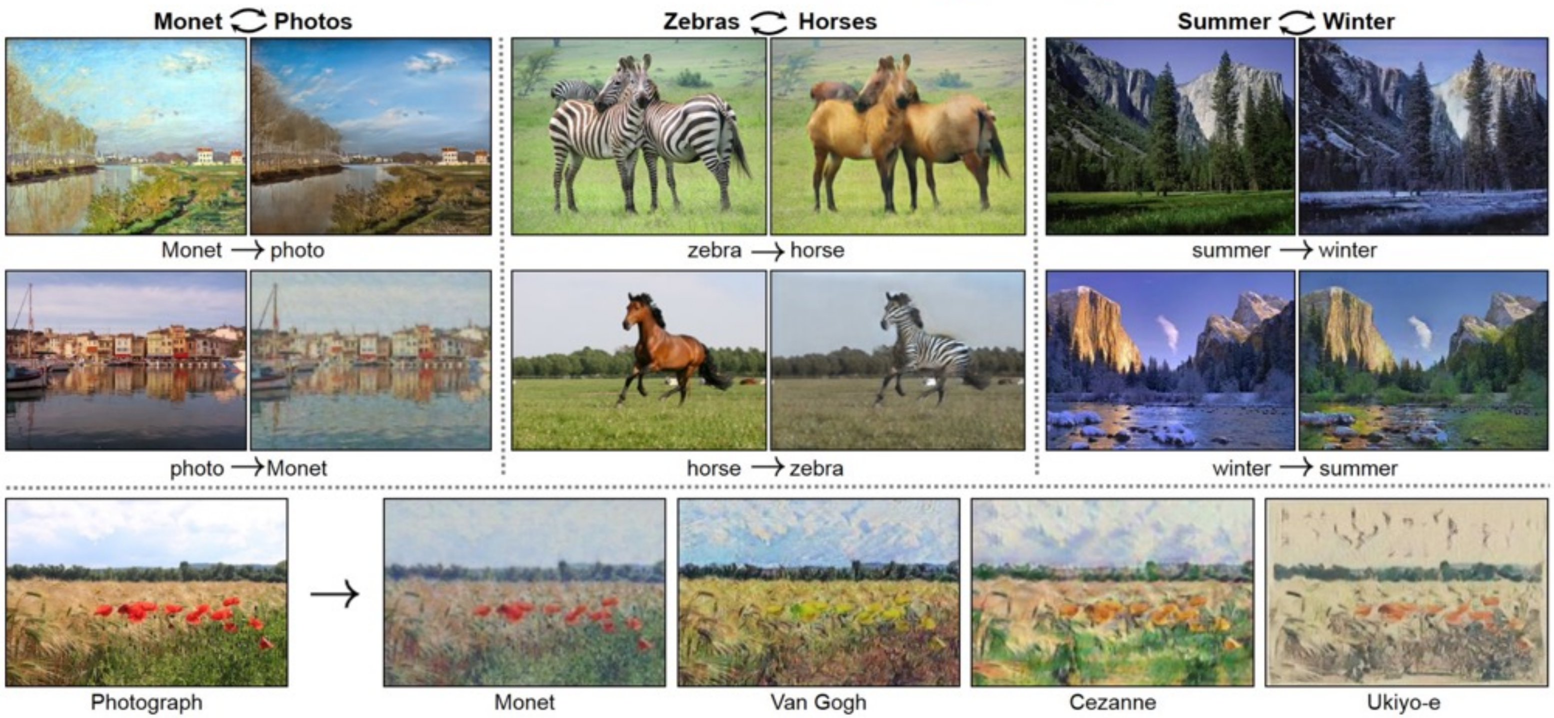
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu* **Taesung Park*** **Phillip Isola** **Alexei A. Efros**

UC Berkeley

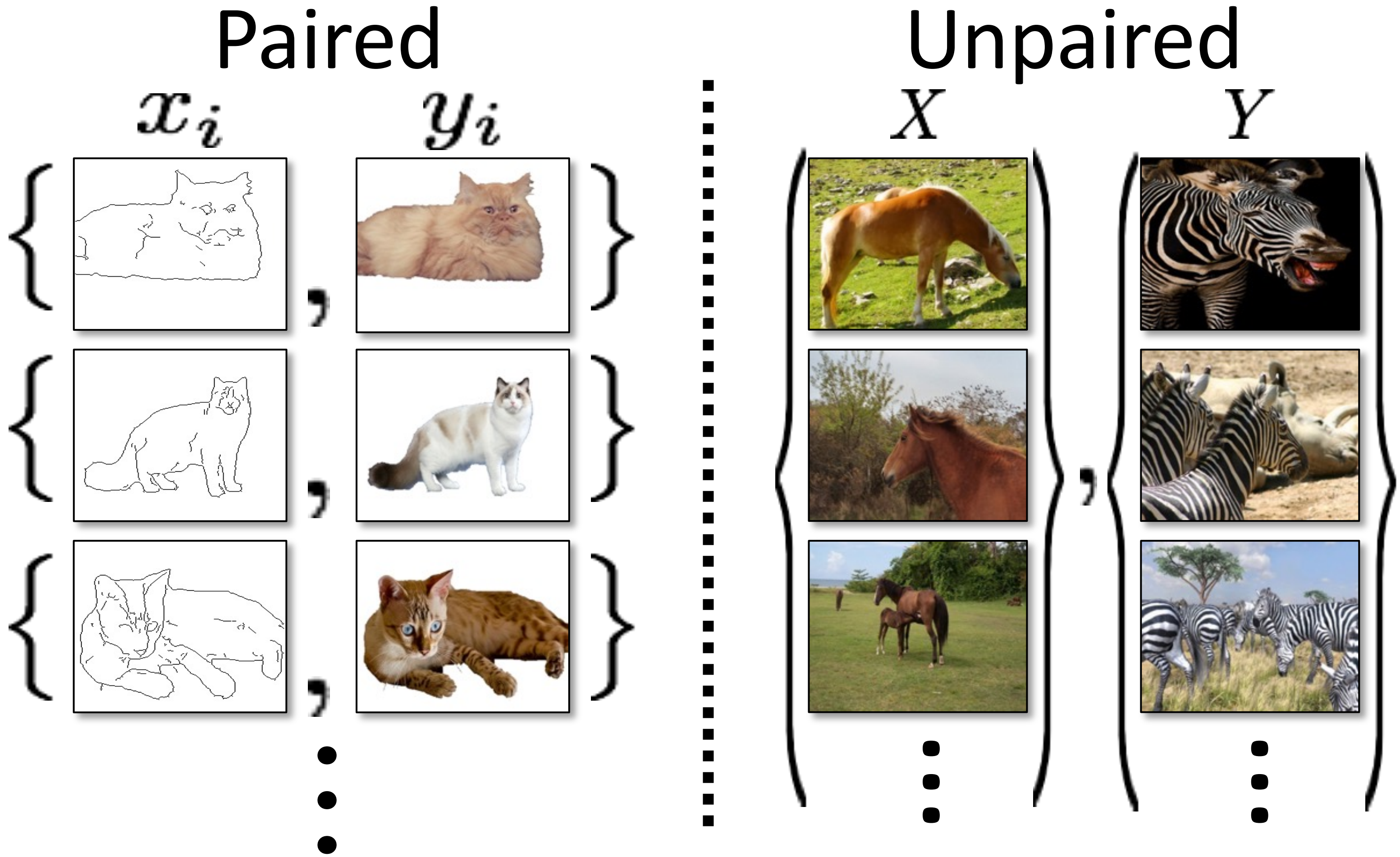
In ICCV 2017

[Paper] [Code (Torch)] **[Code (PyTorch)]**

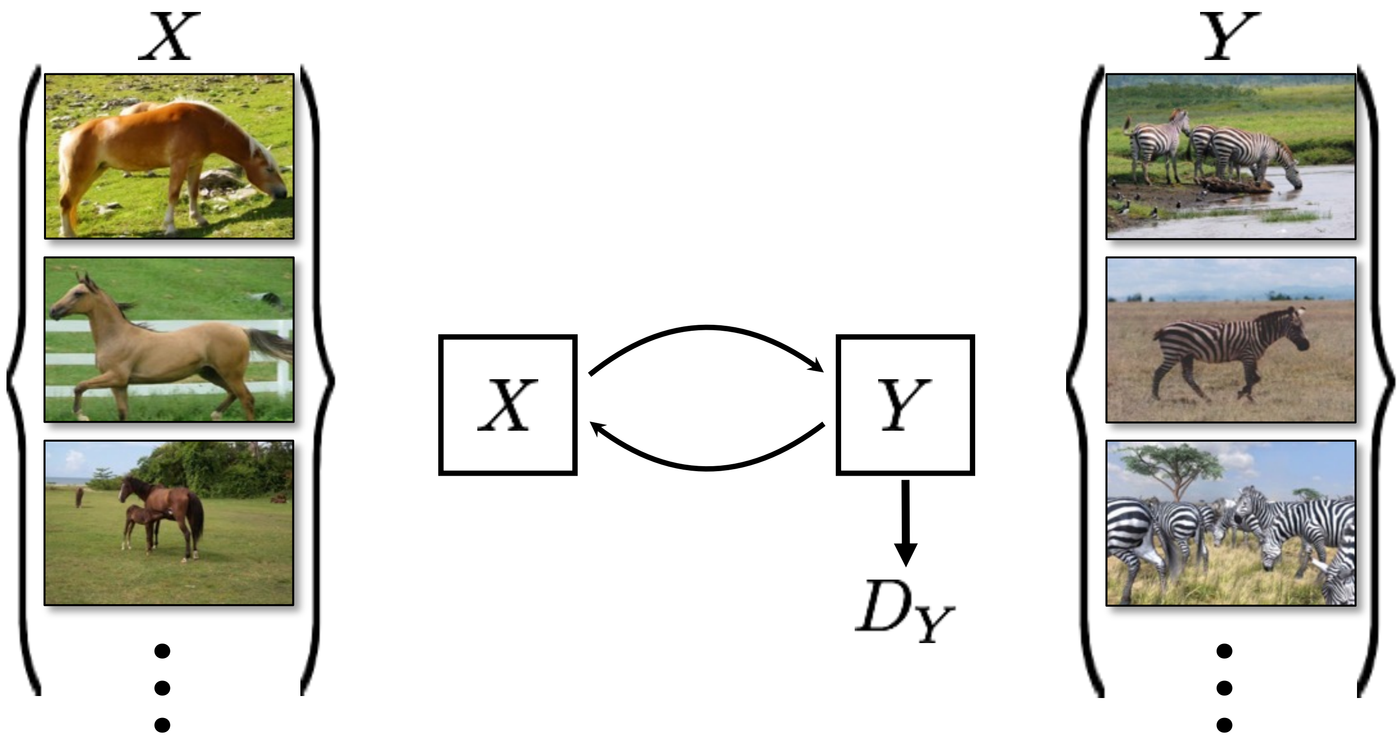


https://junyanz.github.io/CycleGAN/

Paired vs. unpaired data

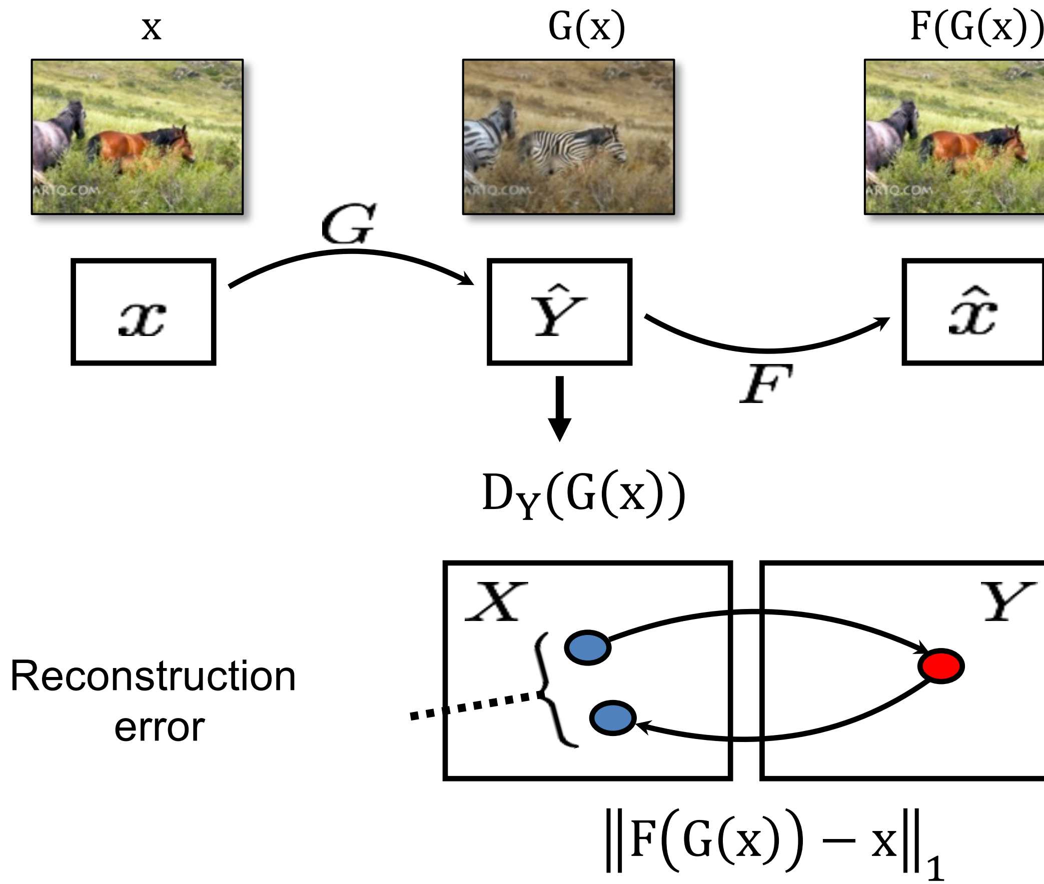


How can we learn from unpaired data?



[Zhu*, Park*, Isola, and Efros, 2017]

Cycle consistency loss



See also [Yi et al., 2017], [Kim et al, 2017]

[Zhu*, Park*, Isola, and Efros, 2017]

Cycle-GAN results

Failure case

Thank you!