上色与重上色

章国锋
浙江大学CAD&CG实验室

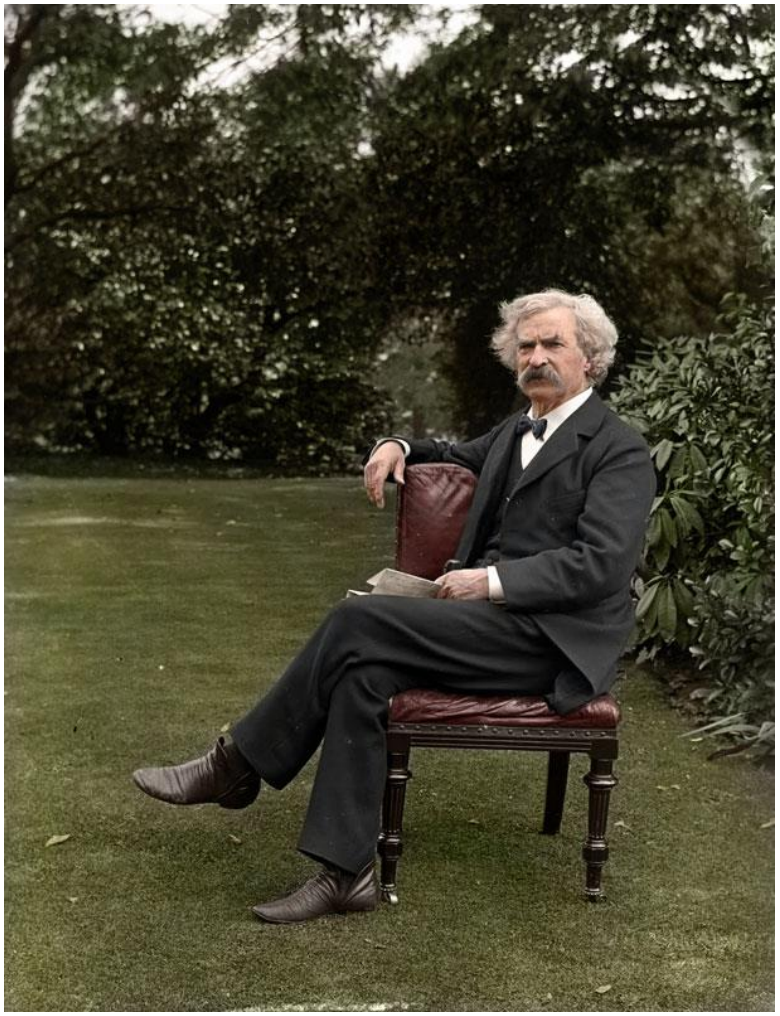
直至1970年，彩色相片仍然十分稀有，人们只能透过黑白照片回顾历史。

如今，利用技术手段，黑白变彩色轻而易举！

Albert Einstein 1939

日本弓箭手 ~1860

Joseph Goebbels（德国纳粹时期宣传部部长），对身为犹太人的摄影师很气愤，摄于1933

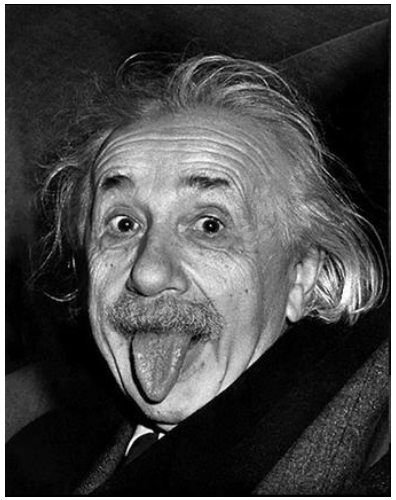


Mark Twain 1900

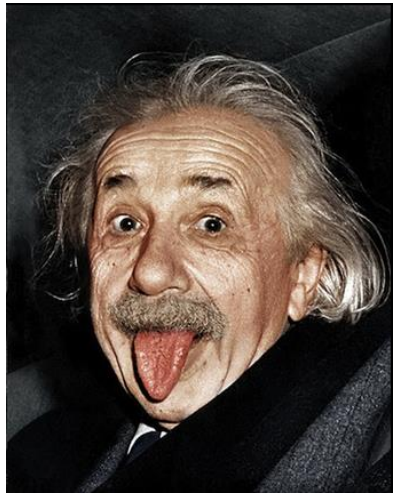
Washington一场车祸 1921

上色

input



output



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

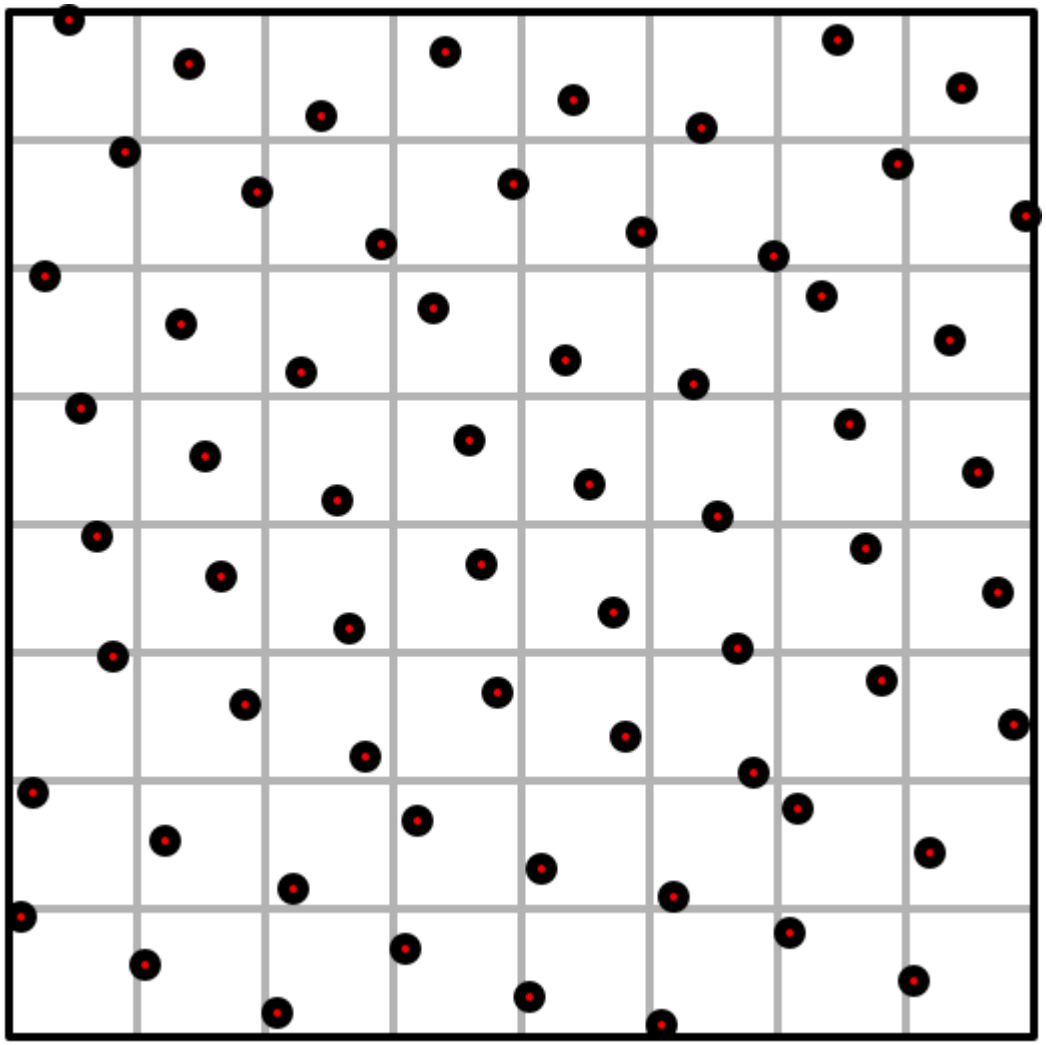
- 将源图像、目标图像从RGB空间转换到Lab空间
- 对转换后的源图像做 luminance remapping

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

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

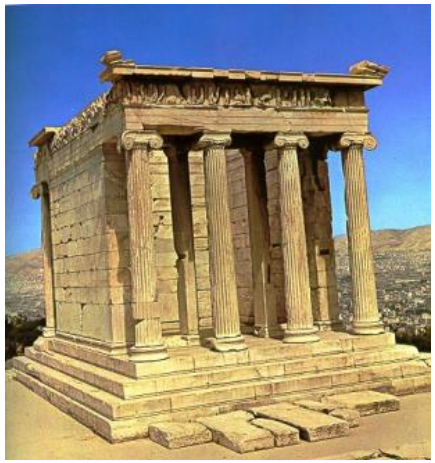
- 在源图像中做 jittered sampling，得到样本（约200个）
- 扫描目标图像，对于每个像素：
  - 在样本中找到最佳匹配点（综合考虑亮度以及与邻域像素的亮度标准差）
  - 将匹配点的 $\alpha\beta$ 赋予该像素，保留目标图像的L通道不变
  - 将目标图像从Lab空间转回RGB空间

Jittered sampling



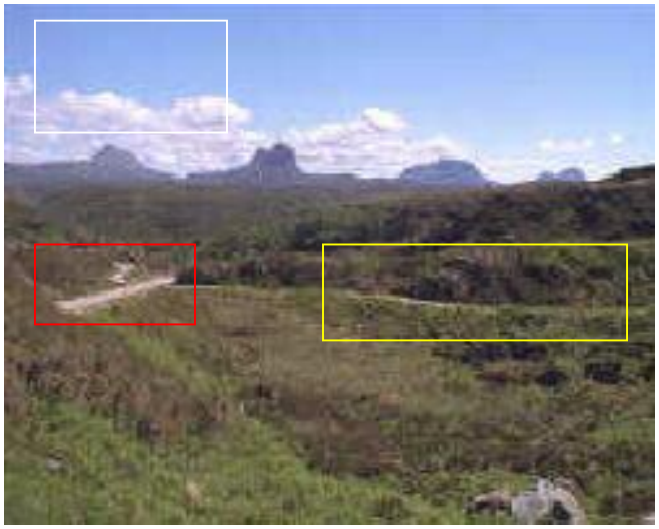
问题

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

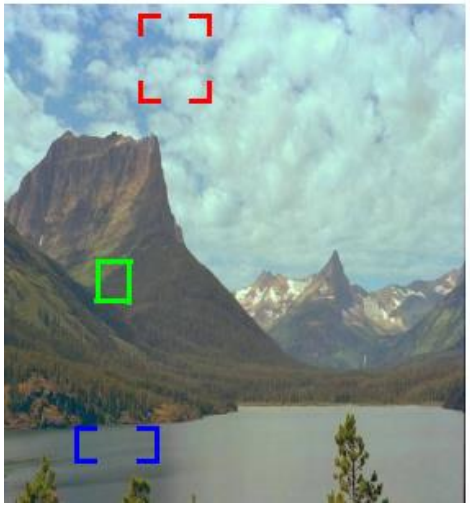
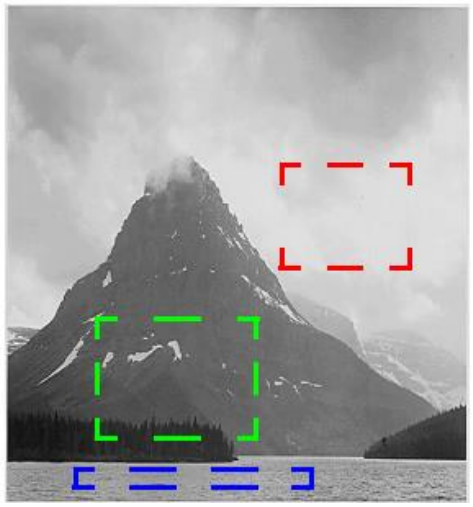
细节

- 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)

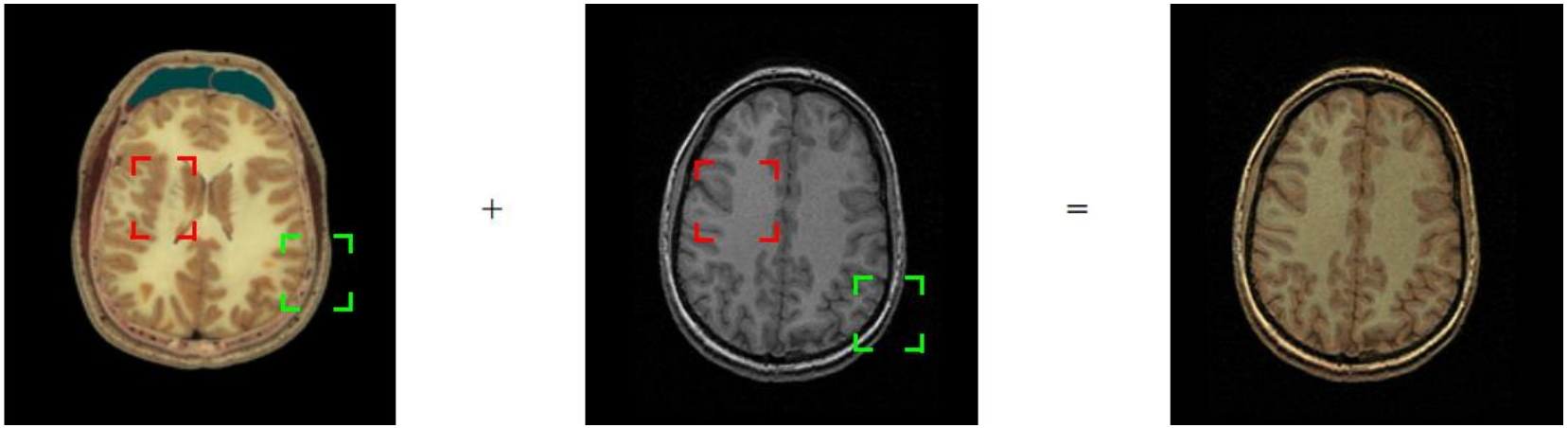
- 扩展剩余区域
  - 对目标图像上的每一个像素，在目标图像已上色的区域中找最佳匹配（根据亮度）
  - 对剩余部分上色

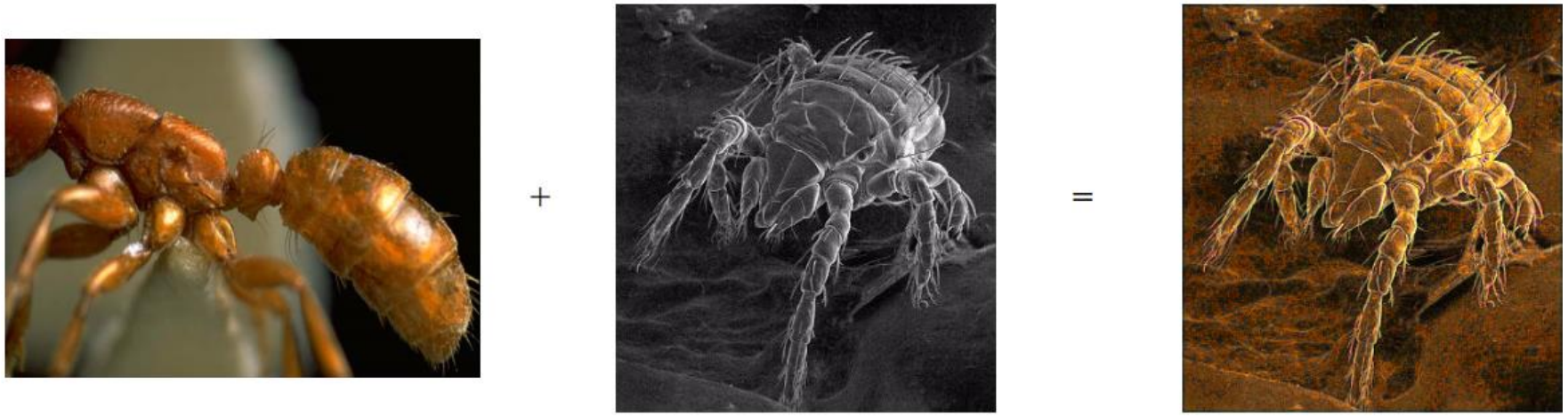
结果展示



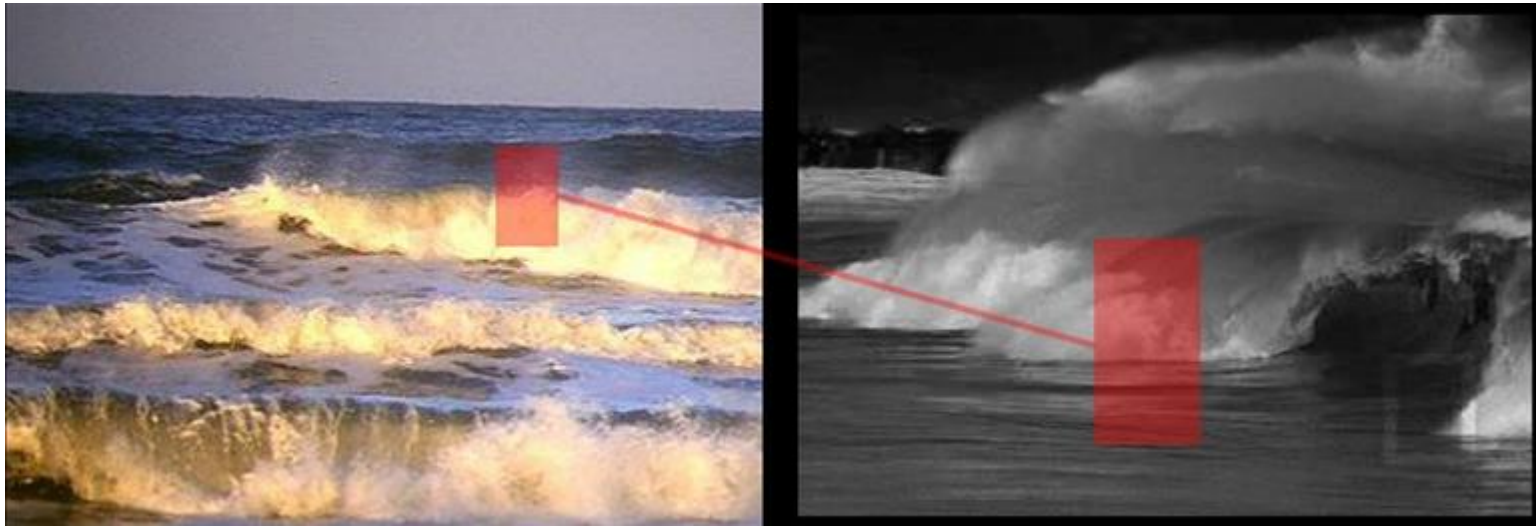

结果展示



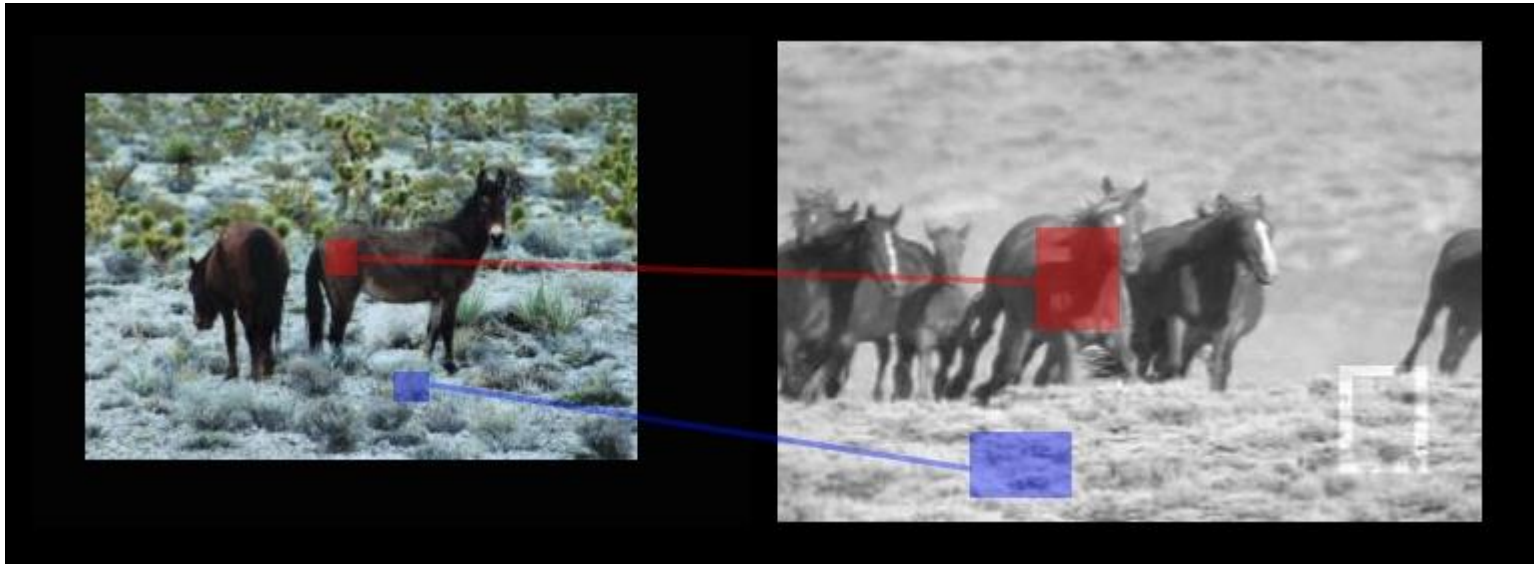


视频上色

- 选取视频某一帧进行上色，然后扩展至其它帧



视频上色



扩展

·选取更为有效的颜色空间

·在源图像上采用更为合理的采样策略

·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 Using Optimization

A. Levin, D. Lischinski, Y. Weiss

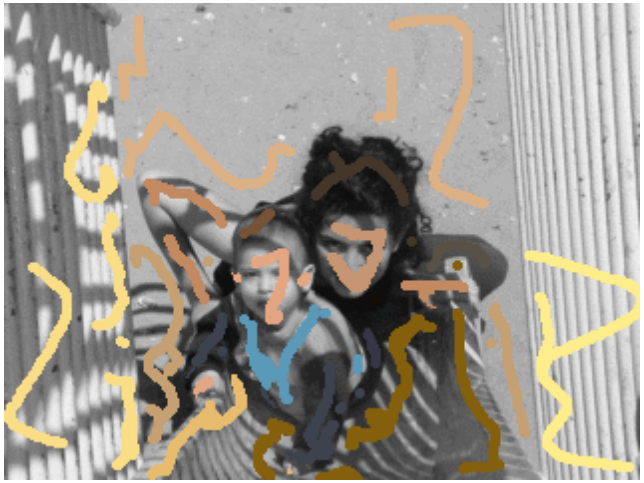
SIGGRAPH 2004

画笔交互式上色

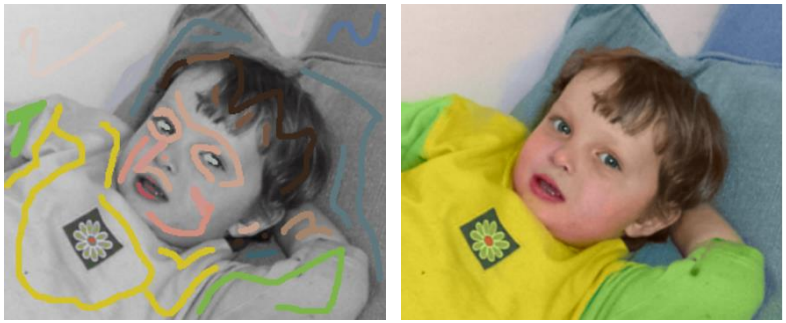
input：带画笔的灰度图

output：上色图

画笔交互式上色



基本思想



· 采用YUV颜色空间。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：

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

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

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

$U(r),U(s)$: 像素r,s的U分量

$N(r)$: 像素r的邻域像素集

w_{rs}：权值，两种形式可选，

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

$$w_{rs}\propto 1+\frac{1}{\sigma_r^2}(Y(r)-\mu_r)(Y(s)-\mu_r)$$

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

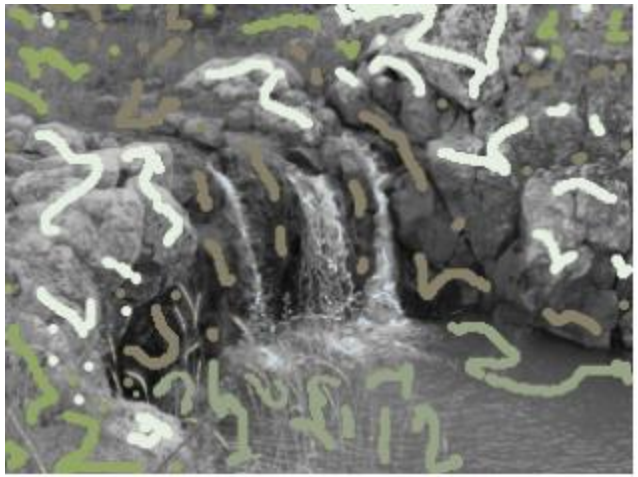
- 对于单张图片，设置邻域半径即可
- 对于视频序列：

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

$v_x(x,y), v_y(x,y)$ ：像素(x, y)在t时刻的光流

结果

速度：~15 秒/帧



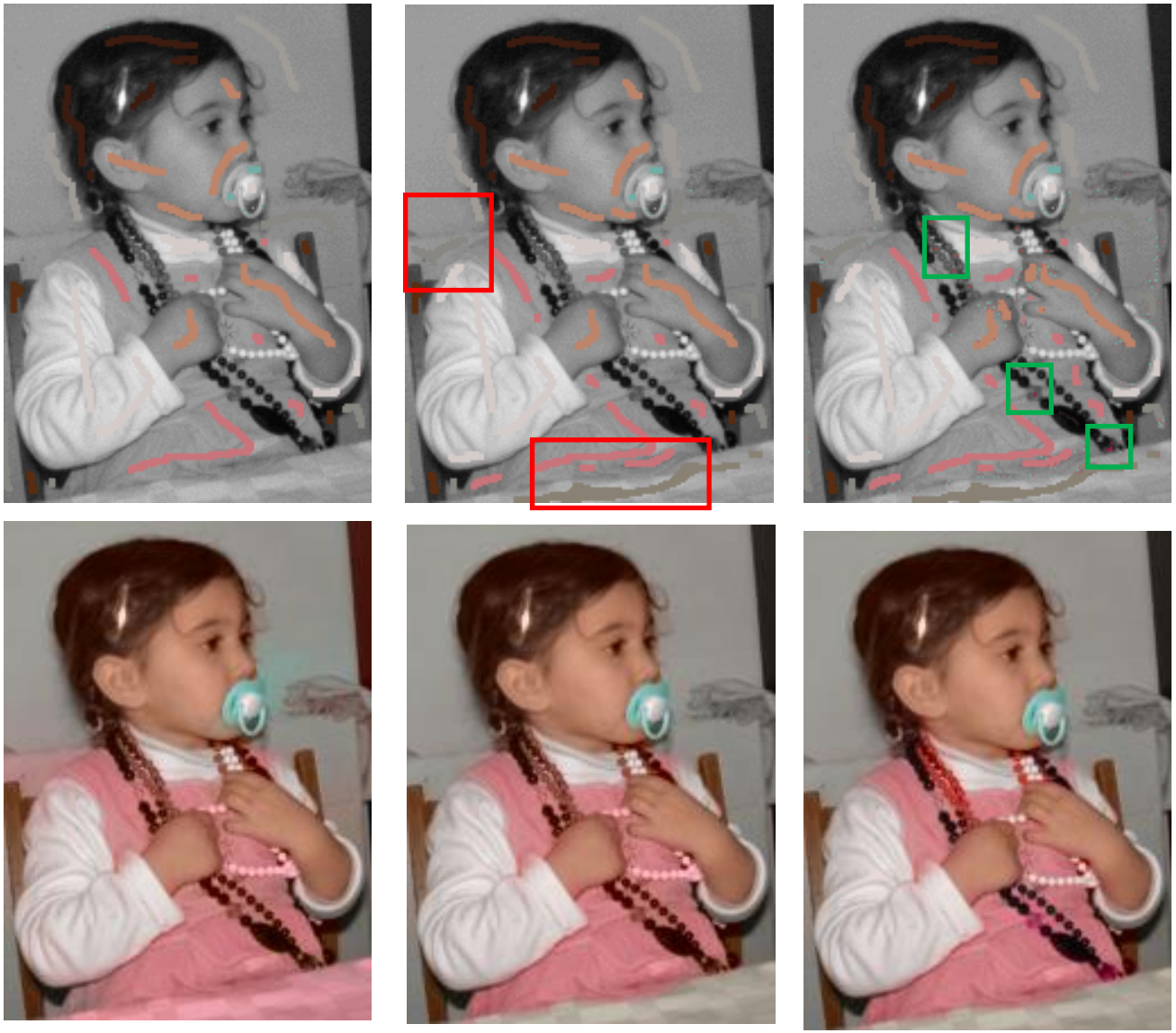
对比基于分割的方法

分割结果

分割上色

our method

结果



增加画笔

结果

重上色


原图像


画笔
（白色：保留原颜色）


重上色

结果






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

Colorization by Example

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

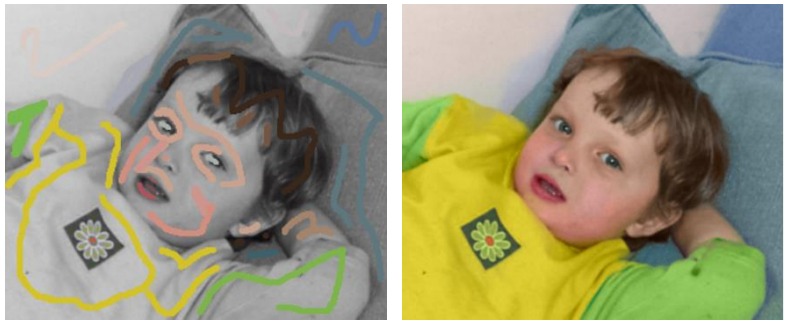
Eurographics Symposium on Rendering, 2005

优点

·对比Welsh et al.(2002) 的方法，提升空间一致性

源图像　　目标图像　　结果

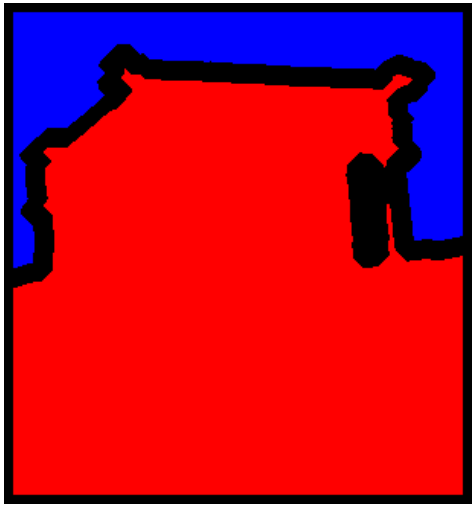
·对比Levin et al.(2004) 的方法，减少了交互

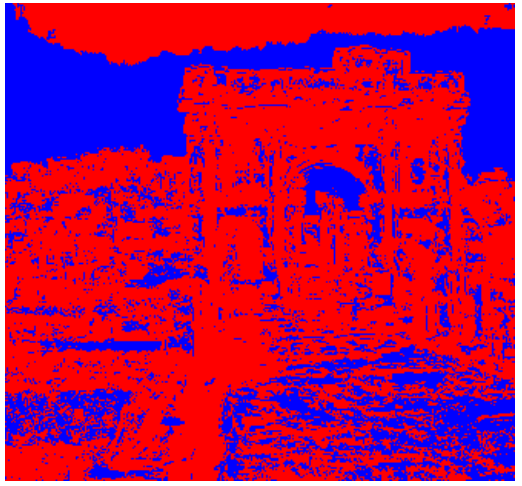


优点

·提升空间一致性

Welsh et al.

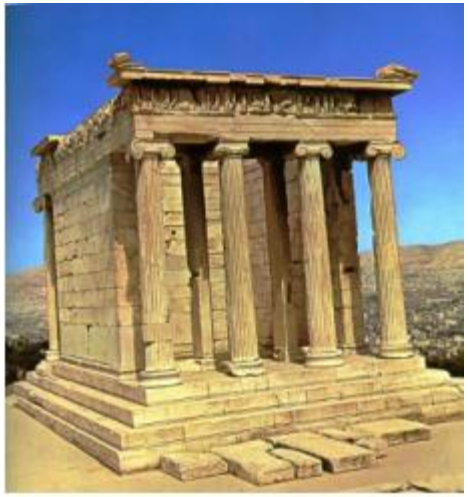


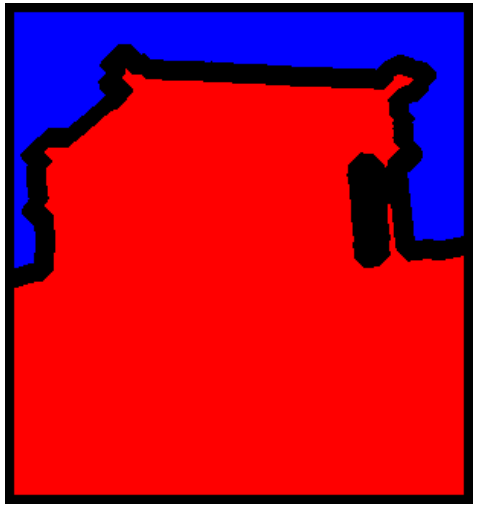


优点

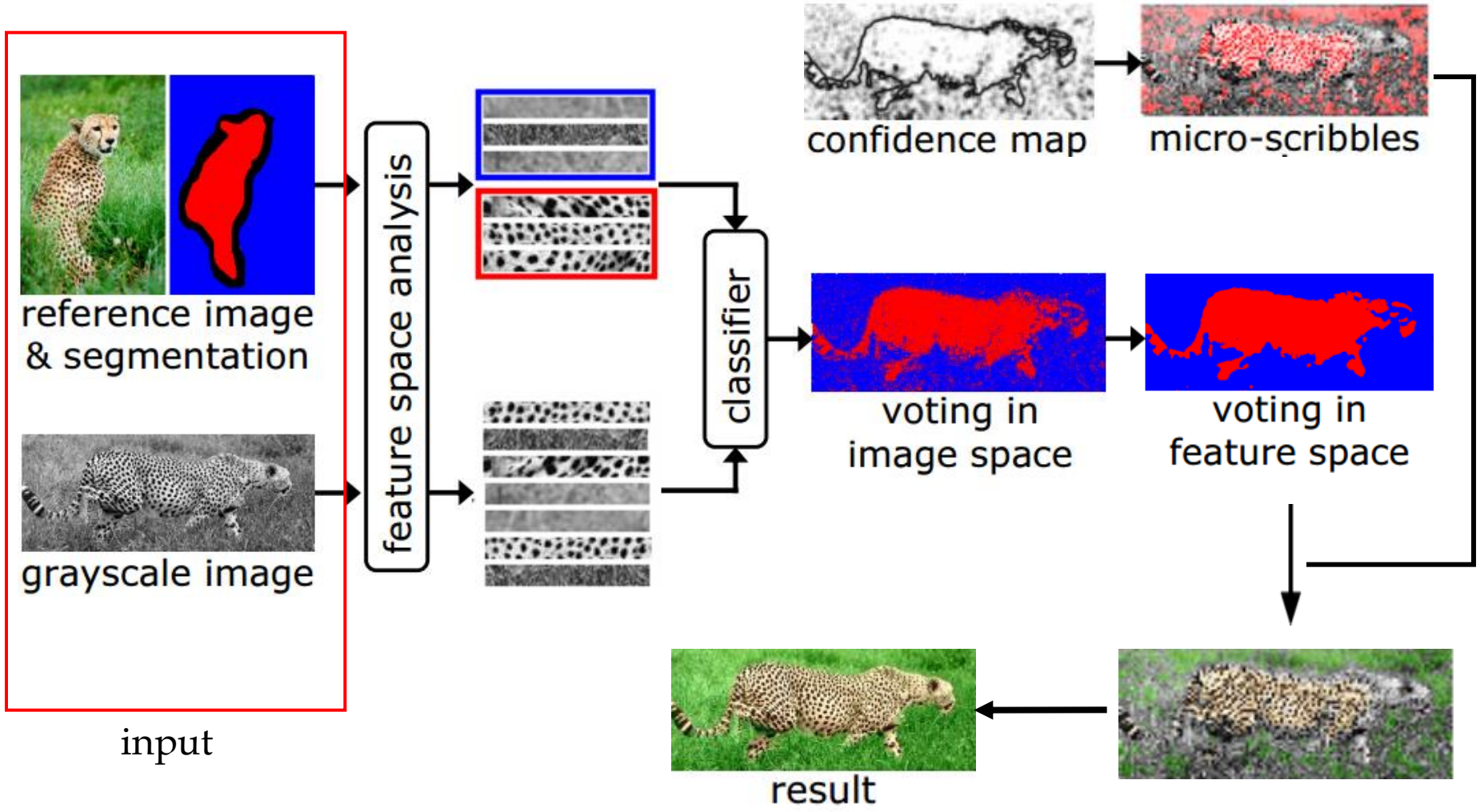
·提升空间一致性

新方法

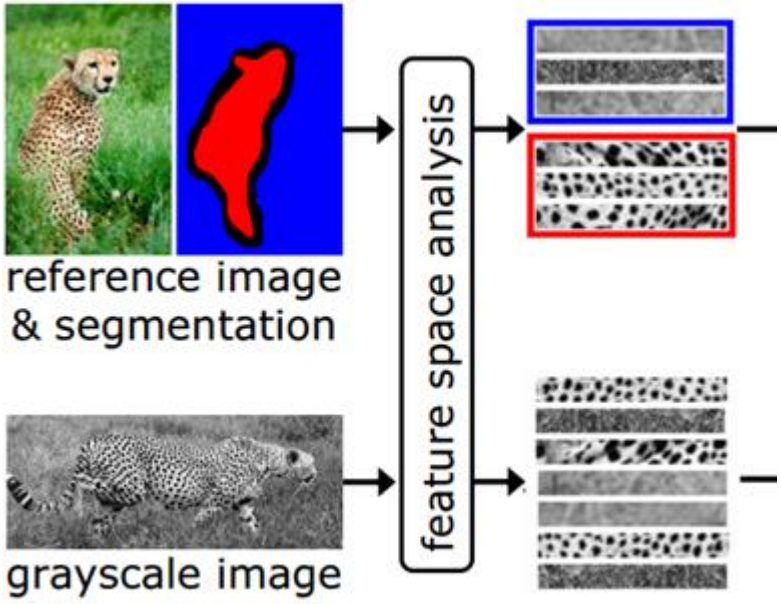




概述


confidence map
micro-scribbles
reference image
& segmentation
feature space analysis
classifier
voting in
image space
voting in
feature space
grayscale image
result
input

概述



1. 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

特征空间

- 监督学习仅使用参考图像的亮度通道，无法使用颜色信息

- 考虑某像素与其邻域半径内的像素一起，构建一个特征向量

- 取 K * K 的邻域像素，附加Discrete Cosine Transform(DCT) 系数，作为像素的特征向量

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

分类器

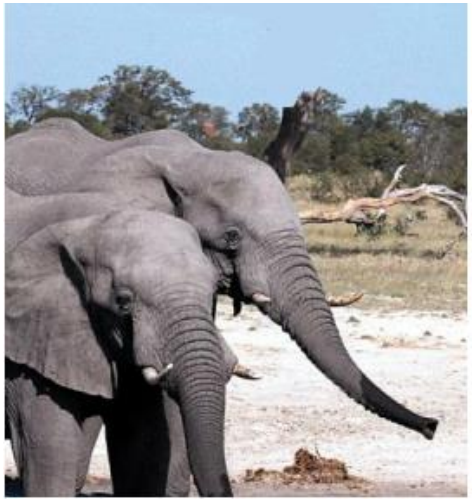
· 分类器的作用：针对一个新的特征向量，能判别其属于哪个类别。

· 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

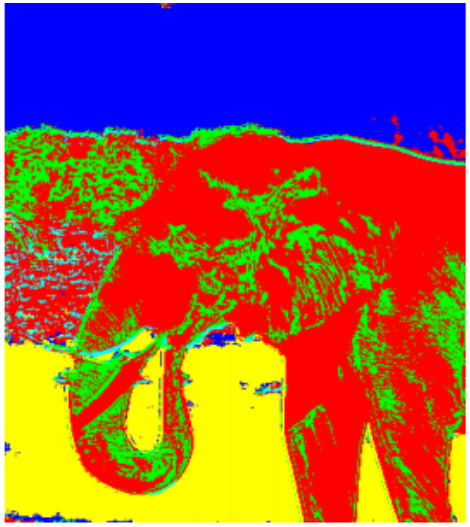
问题



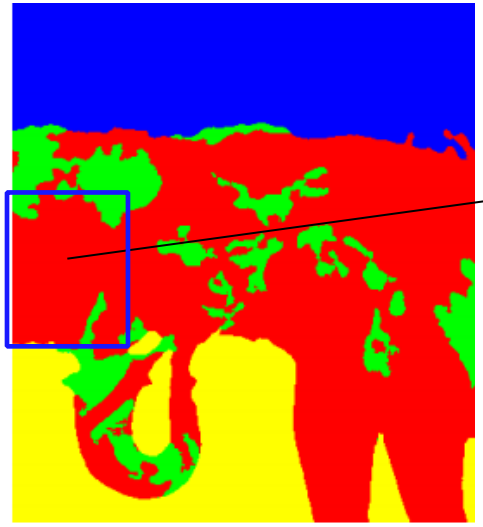
参考图像

已知分类

待上色图像



KNN匹配

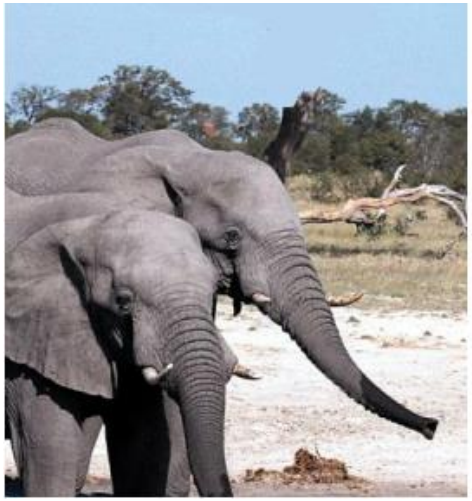


分类结果

KNN使用高维的特征向量，依然有部分像素分类错误

采用linear discriminant analysis，将高维特征压缩

问题

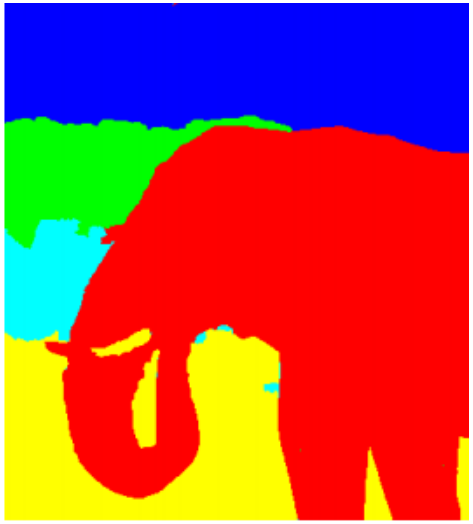


参考图像

已知分类

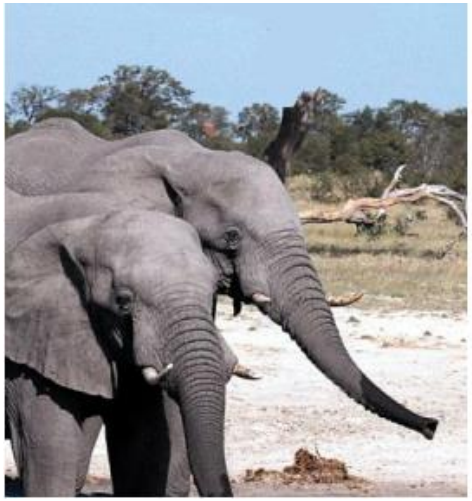
待上色图像

低维KNN



分类结果

问题

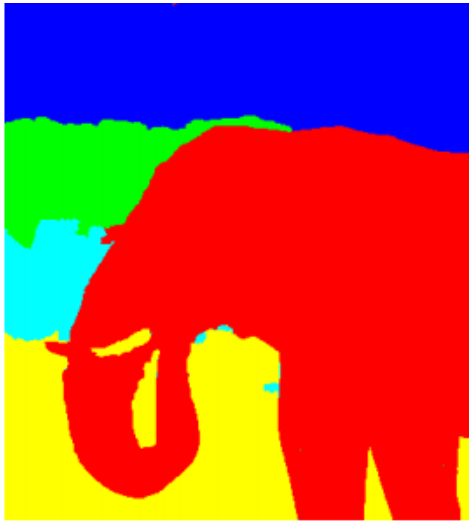


参考图像

已知分类

待上色图像

低维KNN

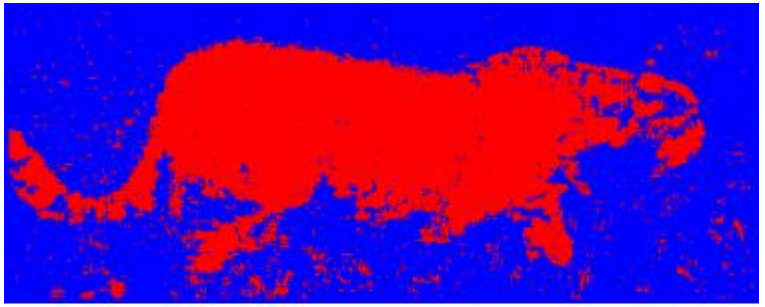


分类结果

上色结果

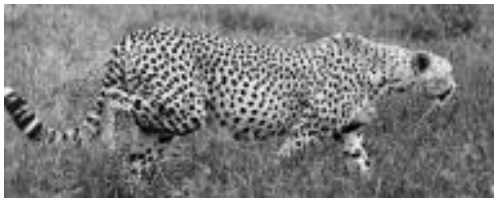
概述

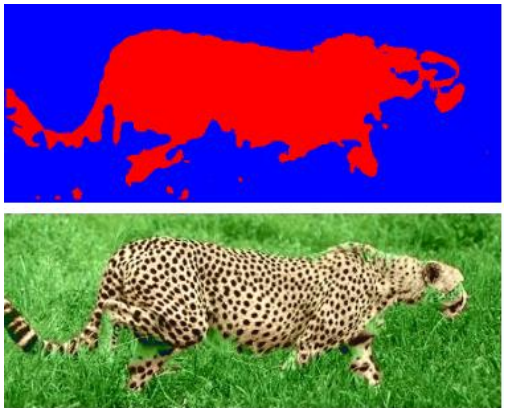
2. 仅仅依靠特征空间，分类依然存在问题



概述

3. 引入图像空间





- 对于像素p的K*K邻域像素，其label或许不同
- 计算置信度，选取最高的作为像素p的label

$$conf(p,l)=\frac{\sum_{q\in N(p,l)}W_q}{\sum_{r\in N(p)}W_r}$$

概述

$$conf(p,l)=\frac{\sum_{q\in N(p,l)}W_q}{\sum_{r\in N(p)}W_r}$$

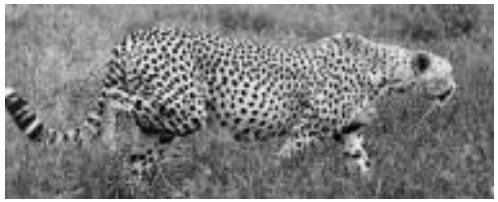
$N(p,l)$：像素p邻域内label为l的像素集

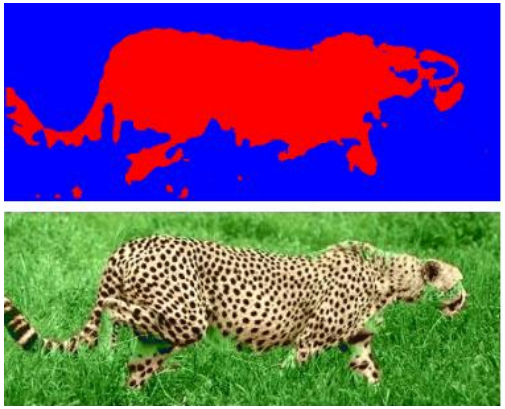
$$W_q=\frac{\exp\left(-D\left(q,M_q\right)\right)}{\sum_{r\in N(q)}\exp\left(-D\left(r,M_r\right)\right)}$$

$D(q,M_q)$：像素q与其最佳特征匹配M_q，在特征空间内的距离

概述

4. 整体优化





优化前

优化后

概述

4. 整体优化

优化前

优化后

$$J(C)=\sum_{p\in I}\left(C(p)-\sum_{q\in N(p)}w_{pq}C(q)\right)$$

$$W_{pq}\propto e^{-(Y(p)-Y(q))^2/2\sigma_p{}^2}$$

$C(p)$：颜色

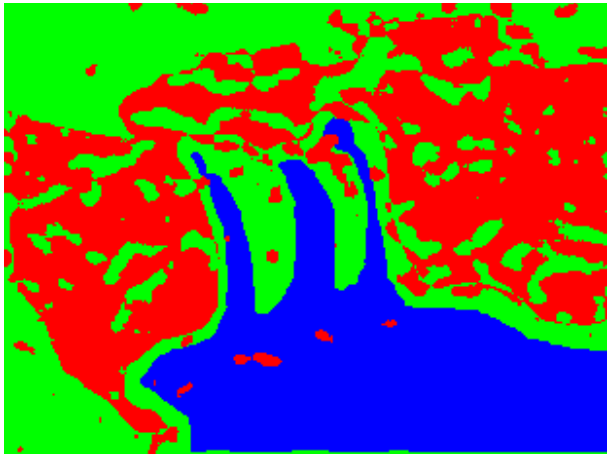
结果

更多结果

参考图像

手动分割

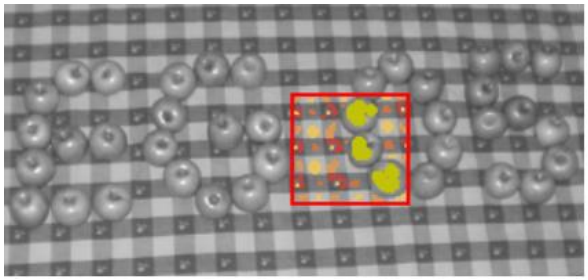
待上色图像

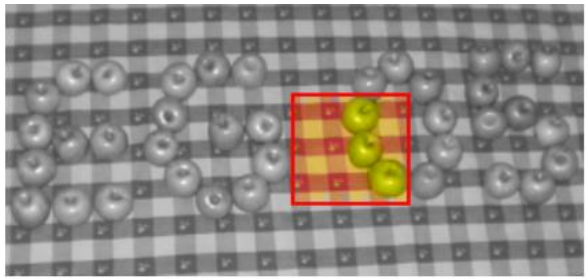


自动分割结果

上色结果

更多结果



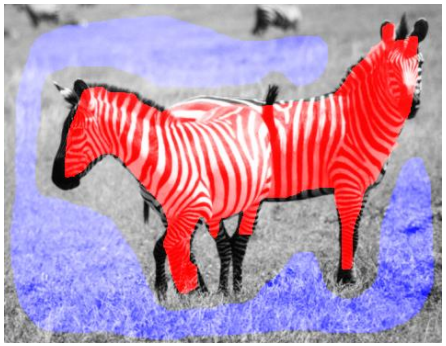


Levin et al.

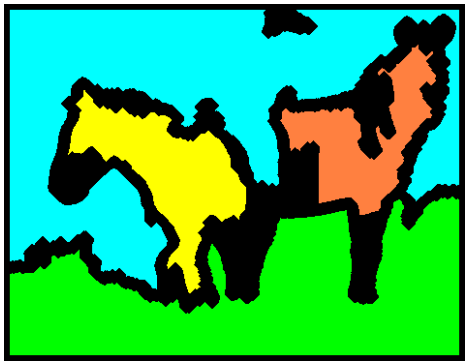
Our method

更多结果

源图像



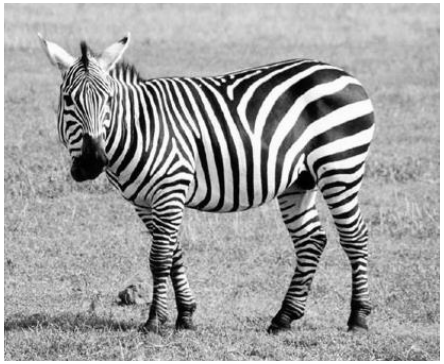
手动分割



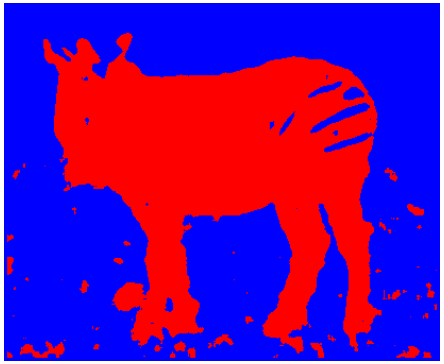
自动分割

得到相同结果

自动分割减轻人工负担



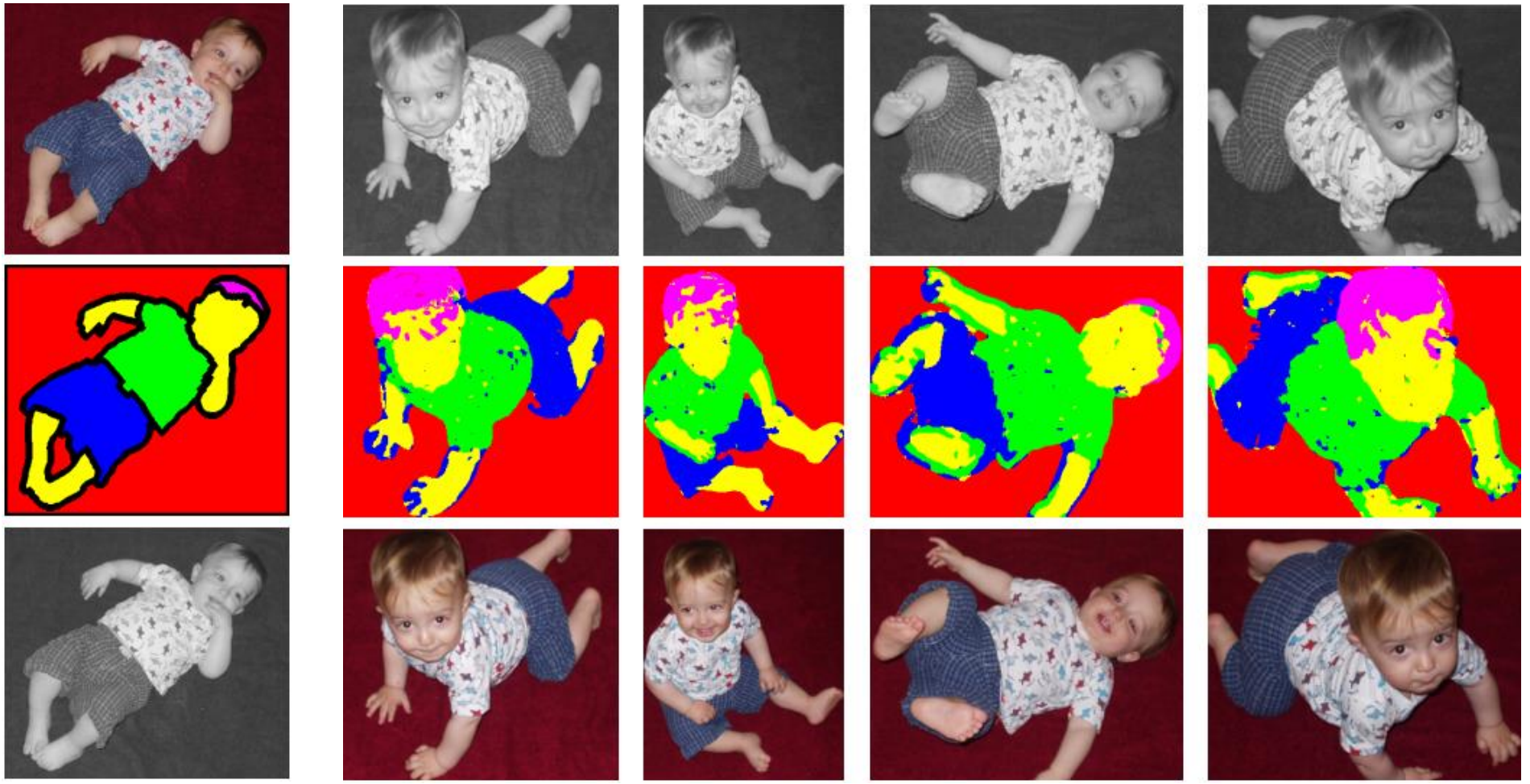
目标图像



基于手动分割分类

基于自动分割分类

多张图片上色



Natural Image Colorization

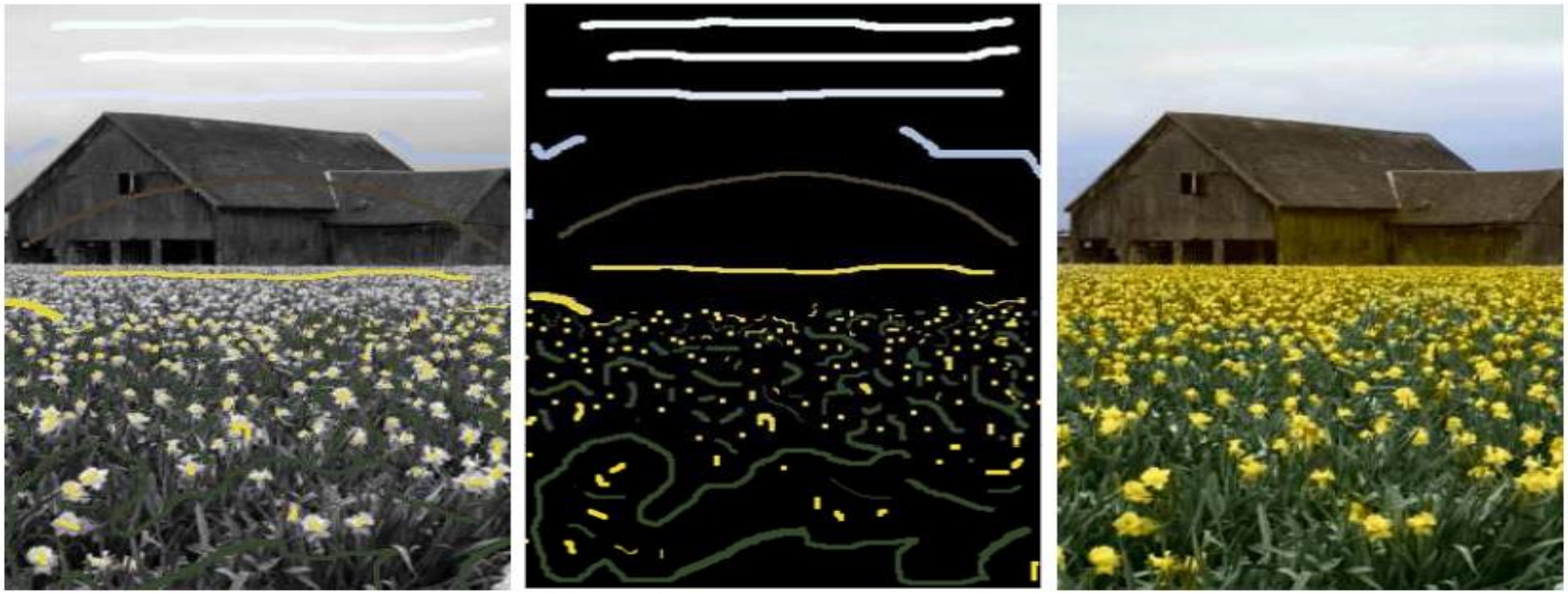
L. Qing, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, H. Shum

Eurographics Symposium on Rendering, 2007

优点

·减少交互

·能处理highly textured图像



画笔上色（Levin et al.）

优点

·减少交互

·能处理highly textured图像

新方法

方法

·用户画笔大致指定相同颜色的区域（左图）

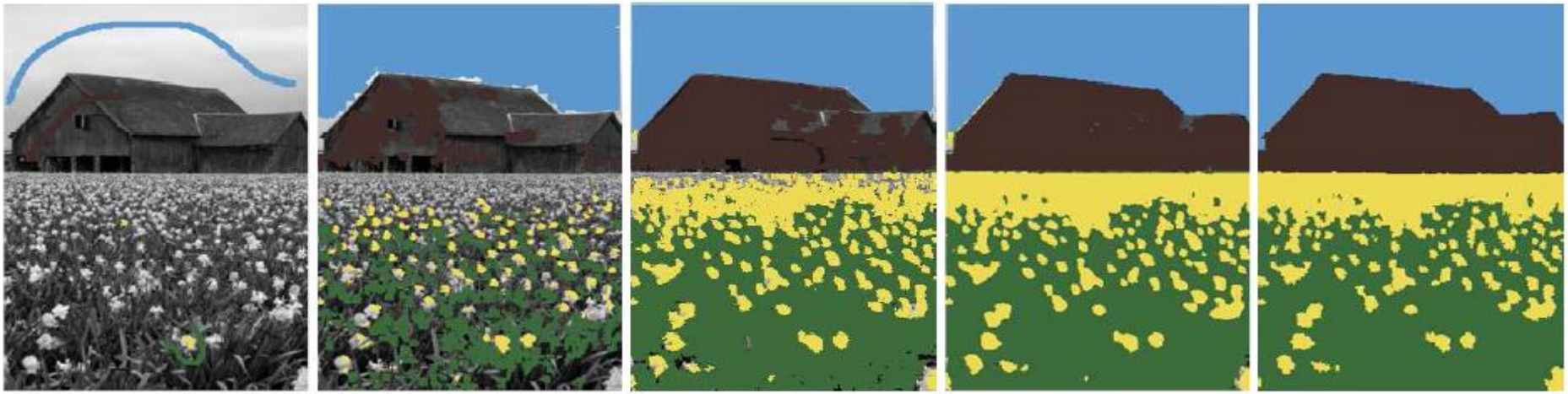
·根据画笔，自动分割图像（中图）

·用户为每一块区域的少量像素指定颜色（中图）

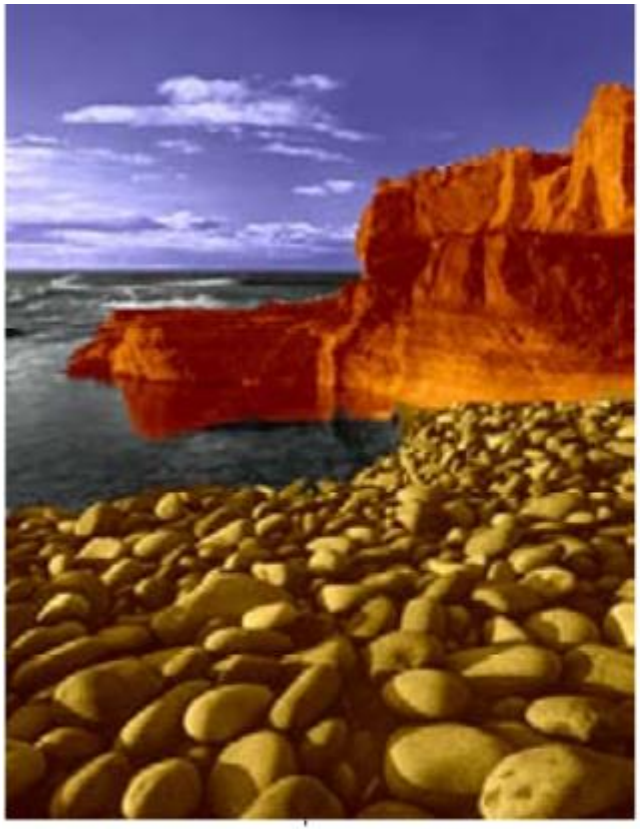
·根据分割结果与用户指定的颜色，给整幅图进行上色（右图）

分割

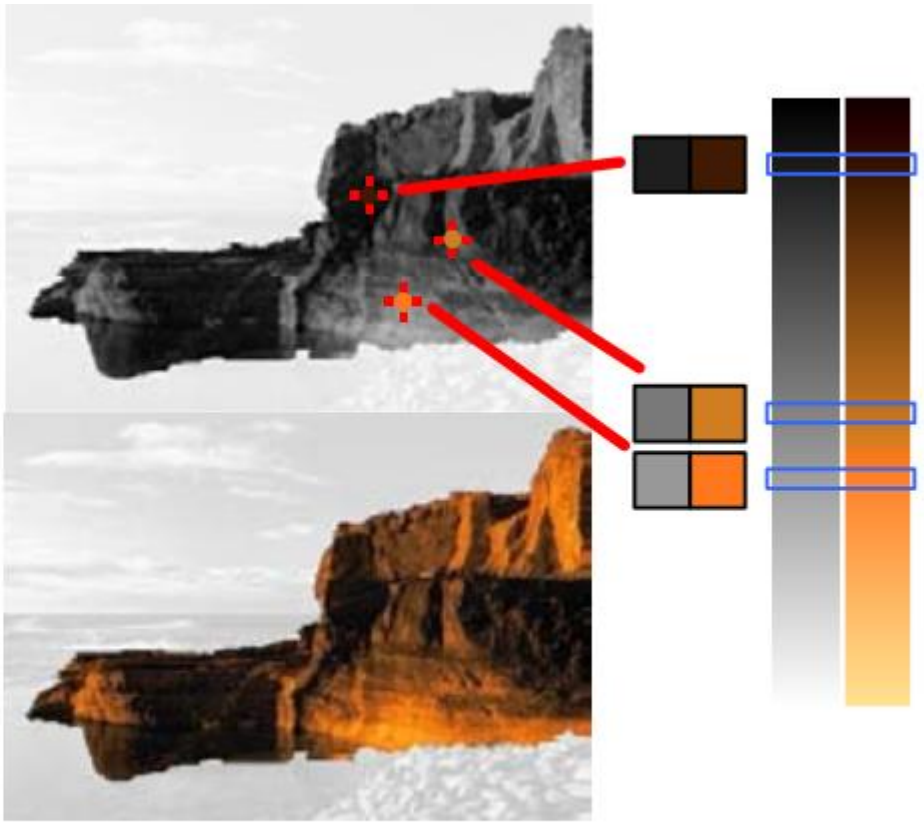
·迭代处理：根据亮度以及纹理，逐步优化分割结果。分割区域不要求空间上连续。



颜色映射

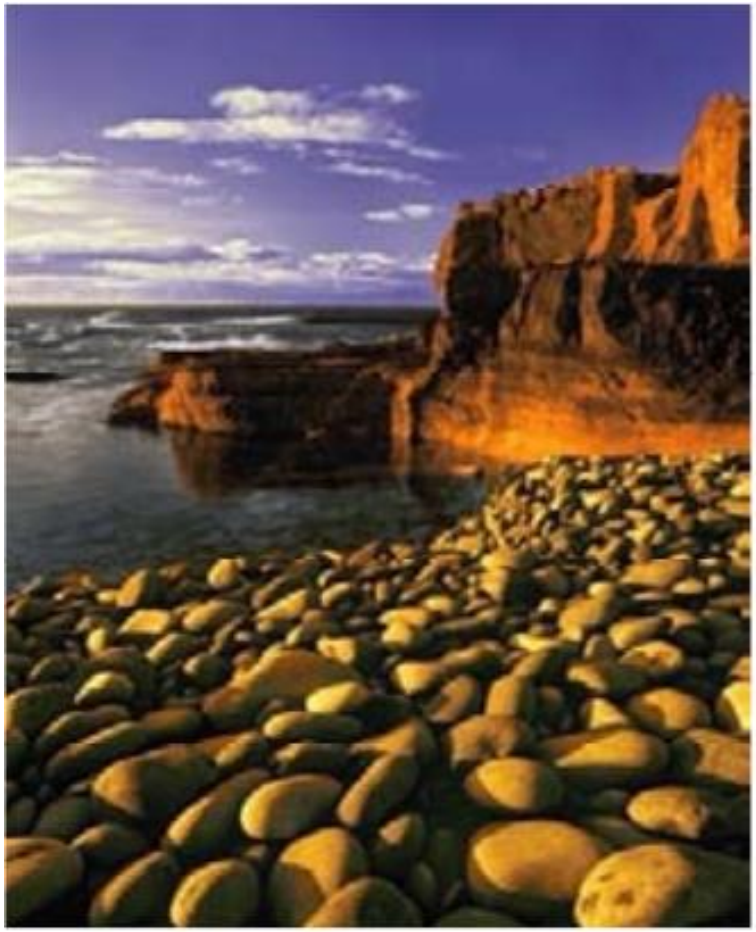


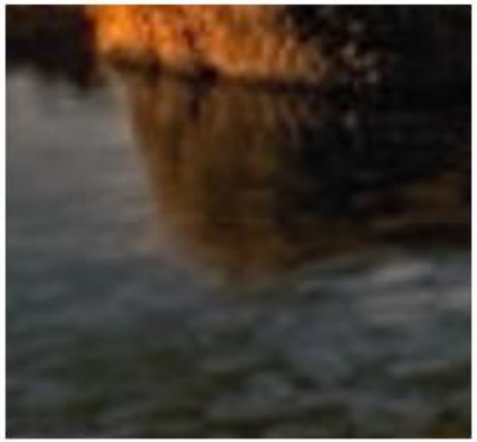
(a)



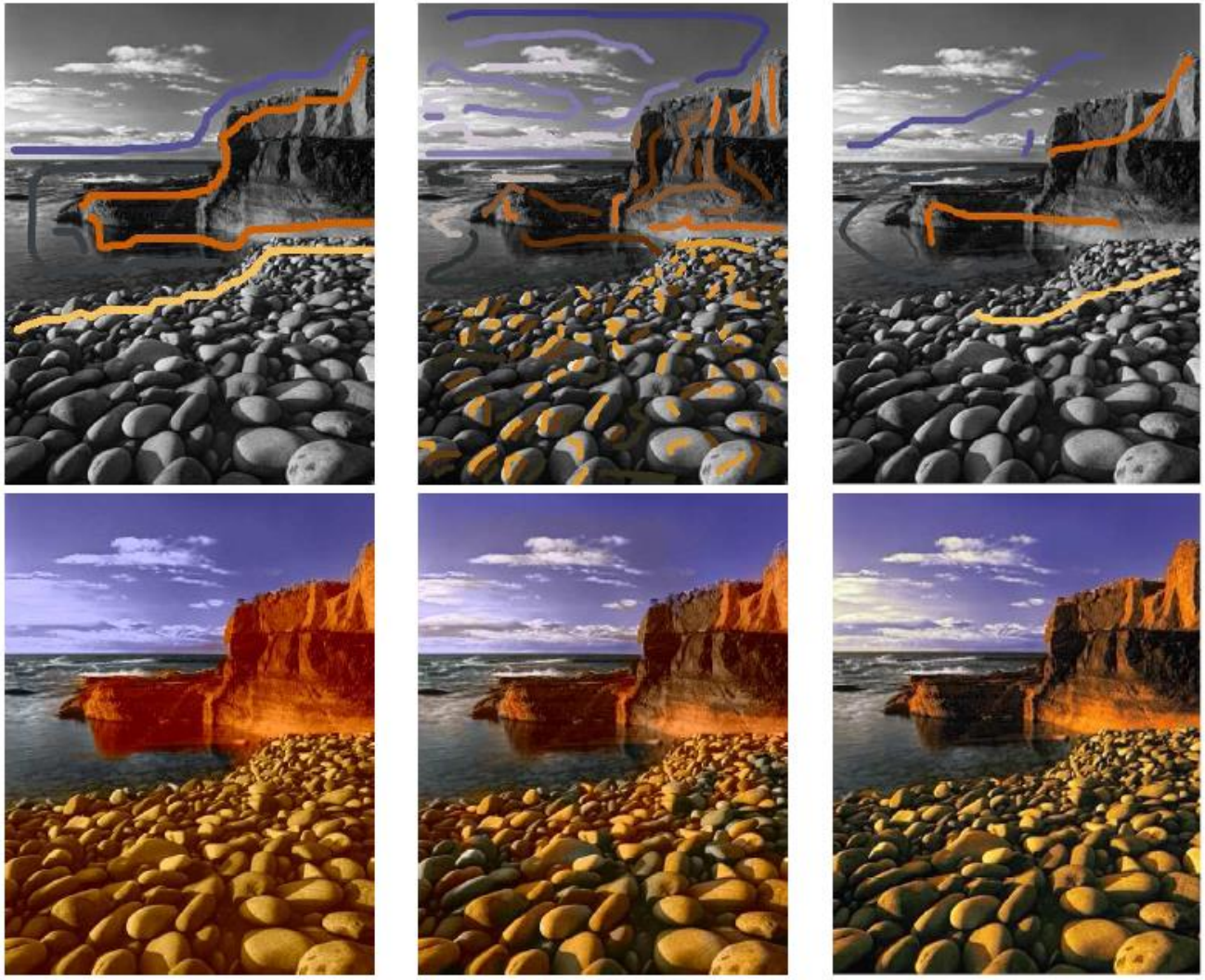
(b) (c) (d)

细节





对比



Levin　　Levin　　新方法

对比

Levin

新方法

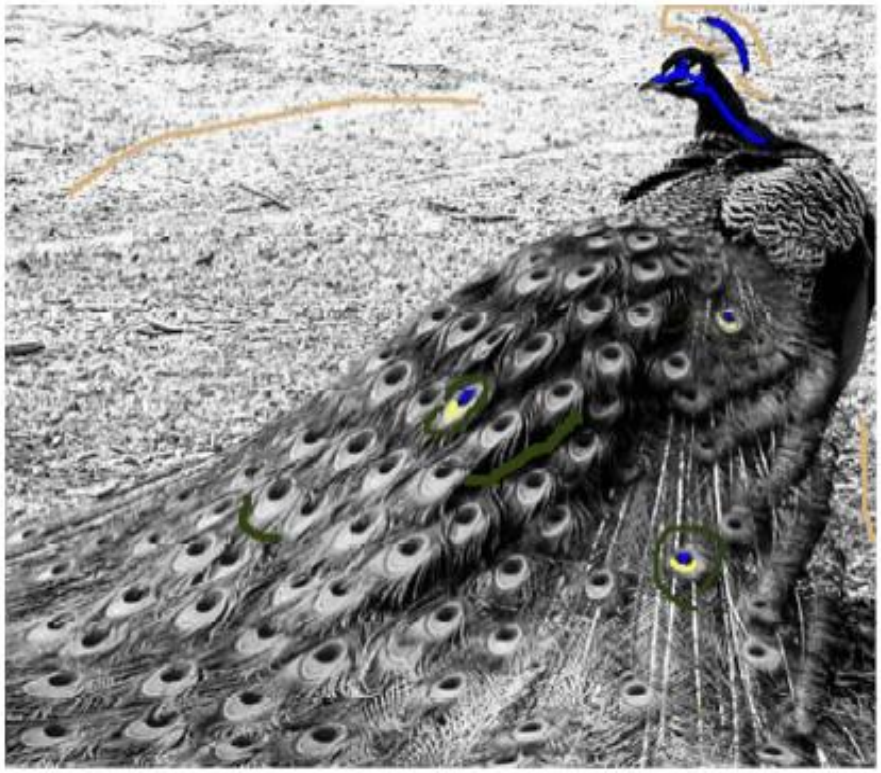
更多结果



更多结果

更多结果

高难度



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

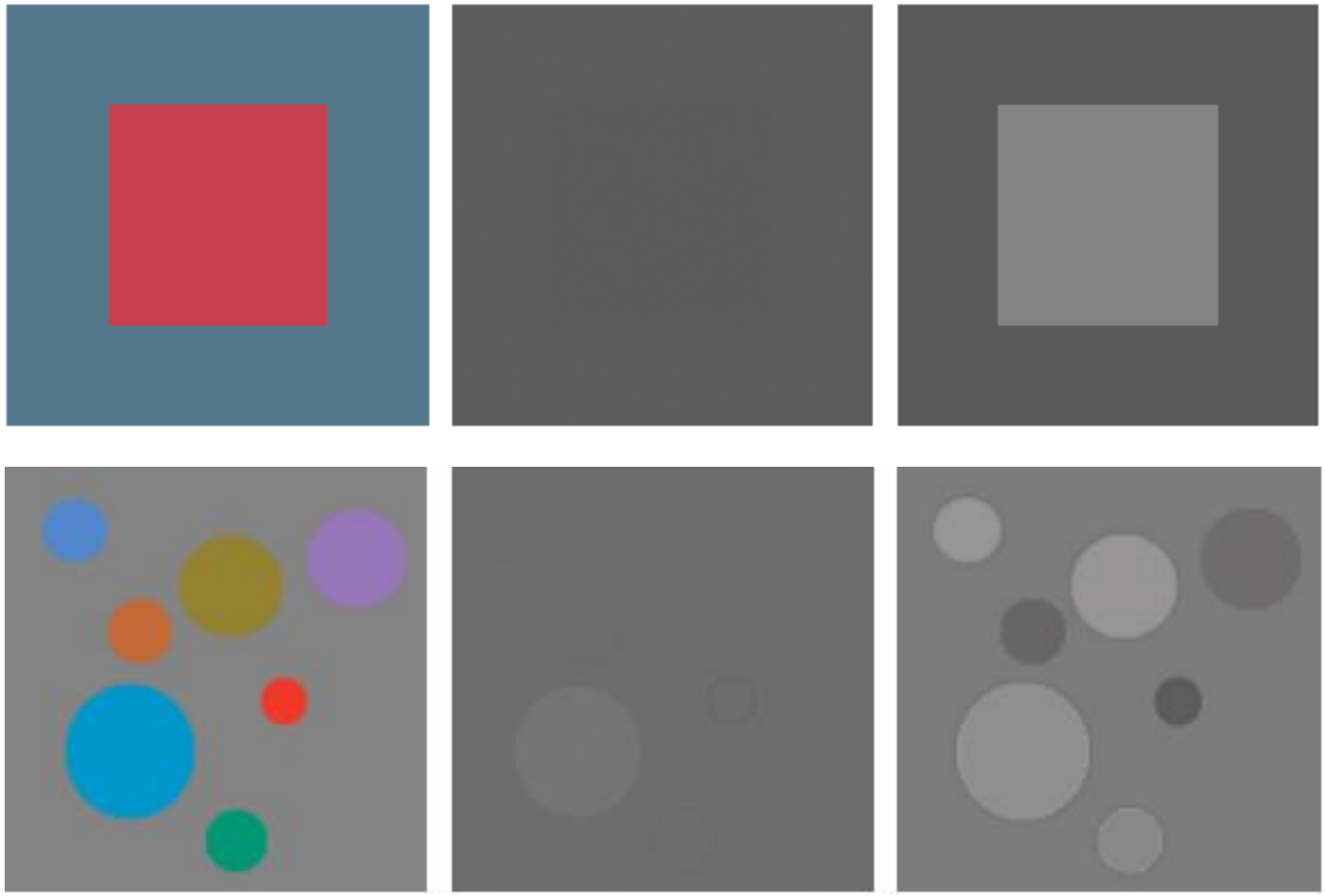
彩色图

仅使用亮度

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）

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

彩色图

CIECAM97 Lum

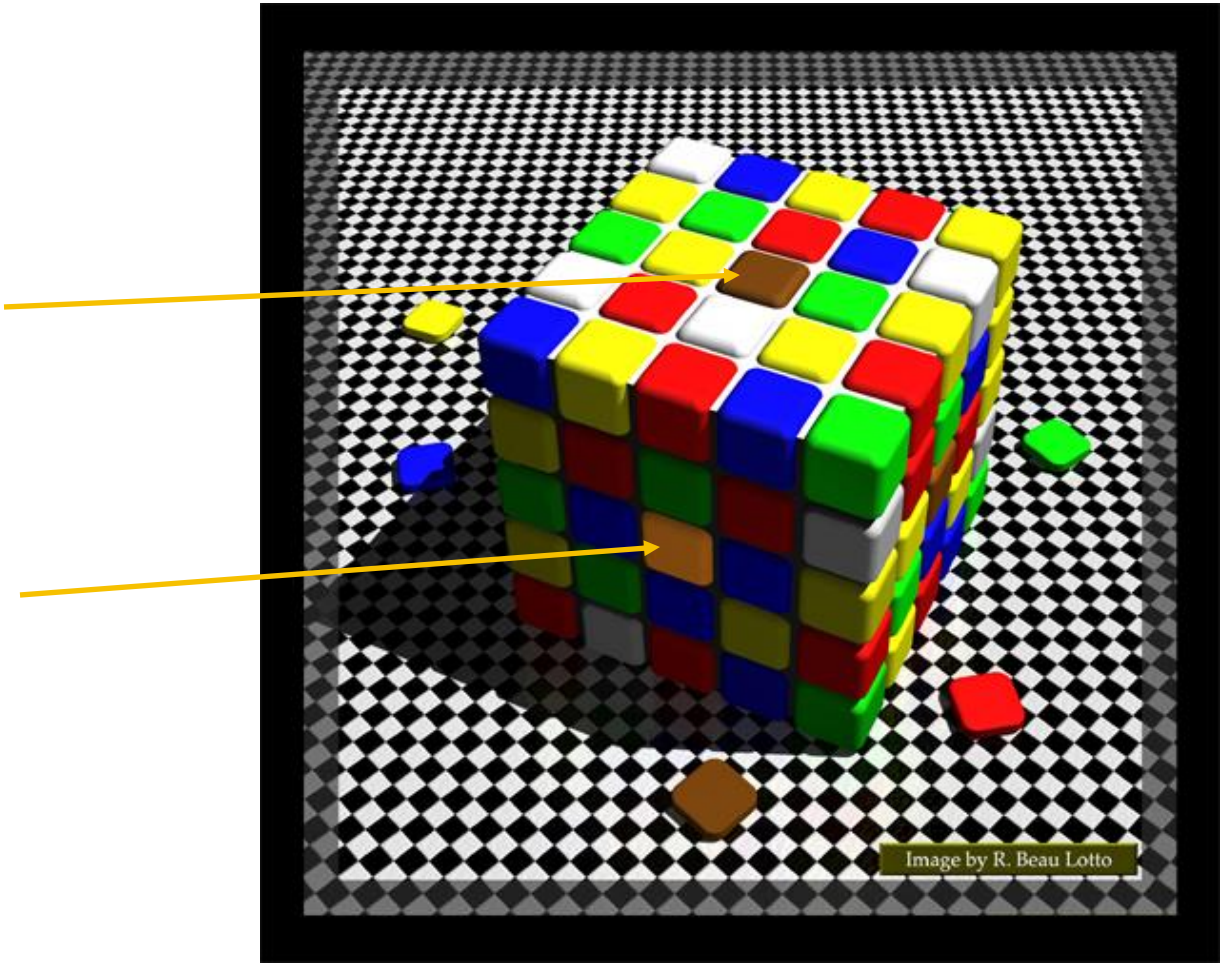
Lab Lum

XYZ Lum

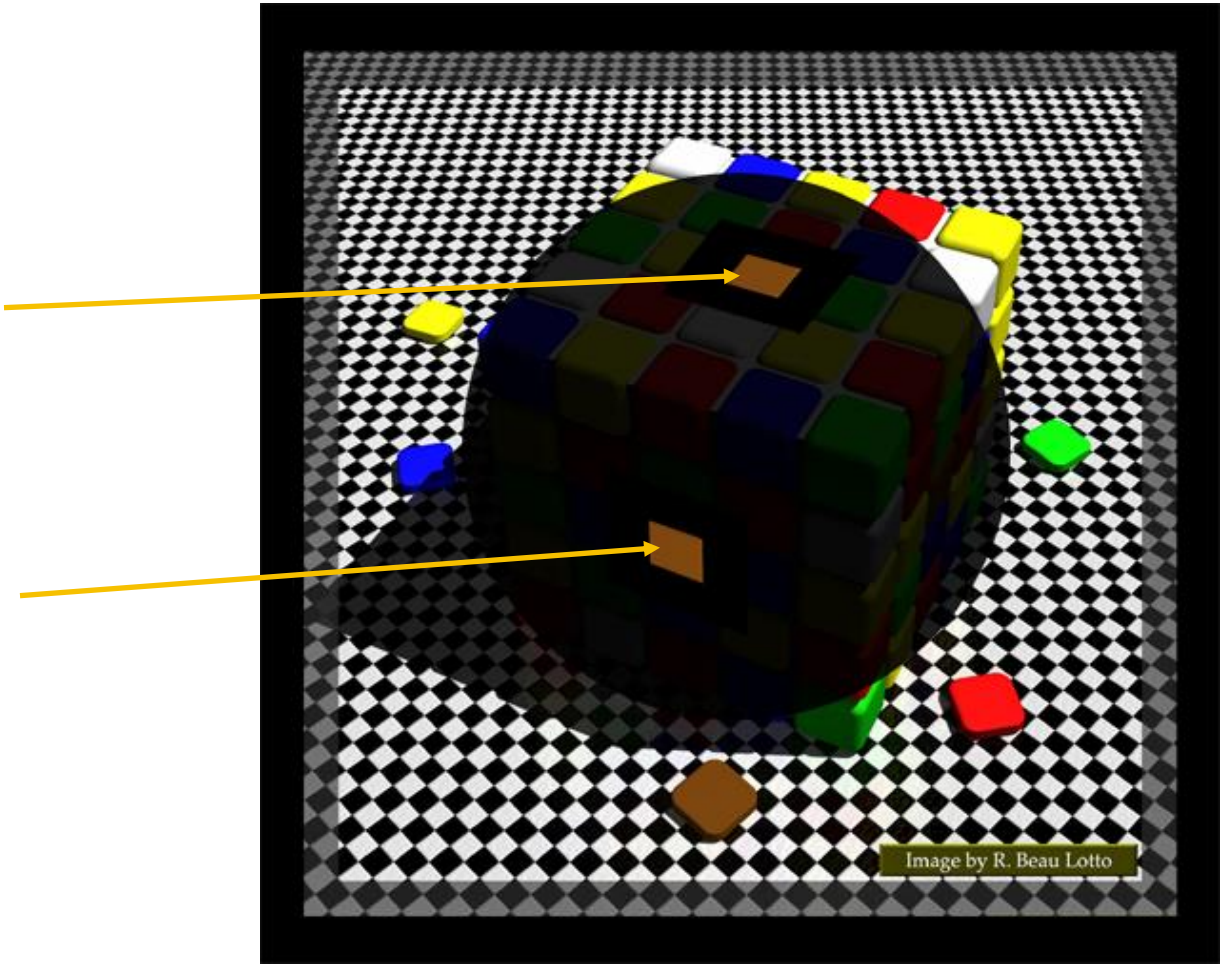
Photoshop

Color2Gray（our method）

人类视觉

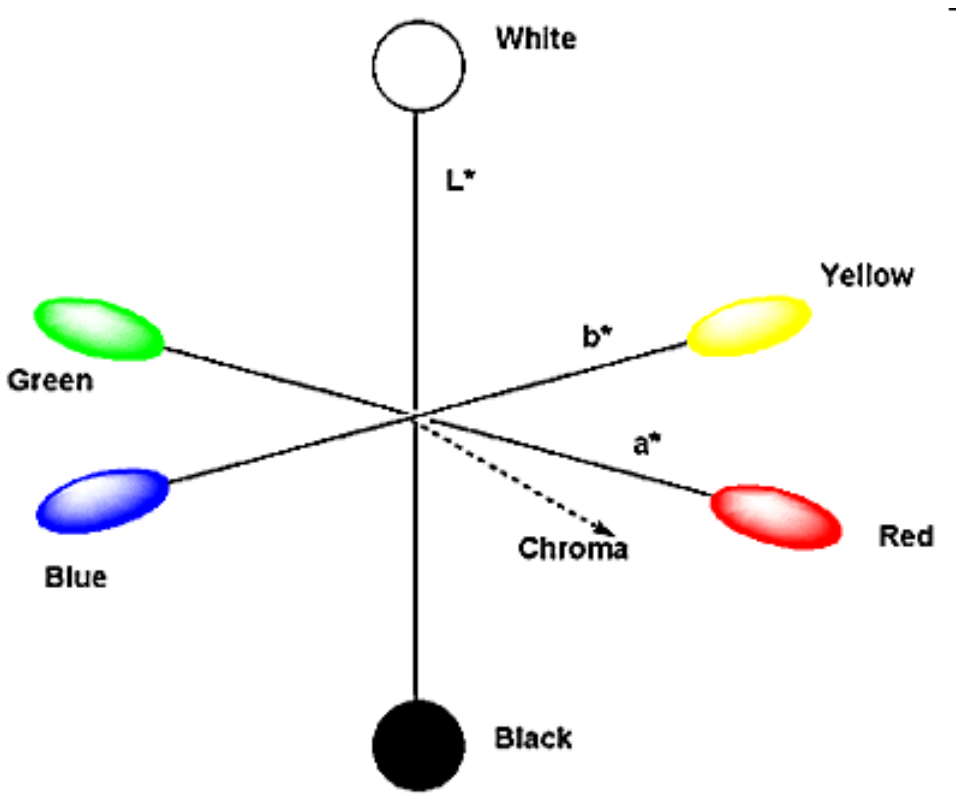


人类视觉



算法概览

·将图像从RGB空间转为Lab空间



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

– 计算亮度距离
– 计算色度距离

} δ_{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

参数设置

可以通过调节三个参数，得到不同的优化效果

μ:决定每个像素的邻域半径

α:用于控制色度差异的范围

θ:决定色度差异对亮度的影响，是增强还是减弱

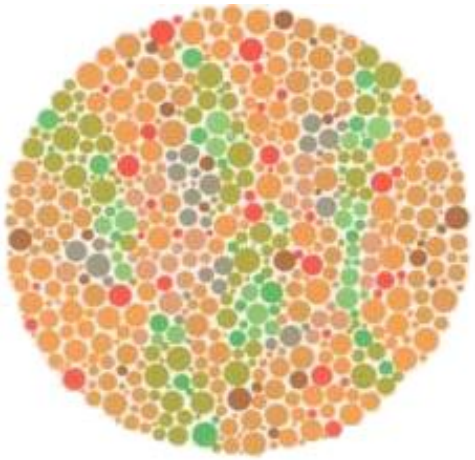
μ：邻域半径

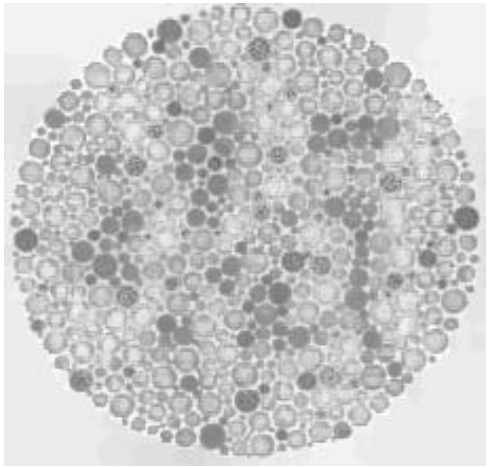
μ＝2

μ＝16

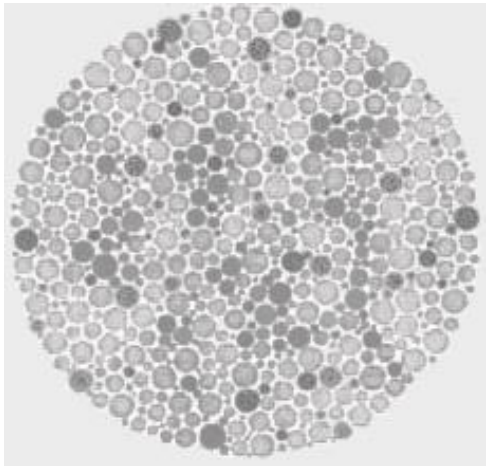
μ＝整幅图像

μ：邻域半径





μ＝16



μ＝整幅图像

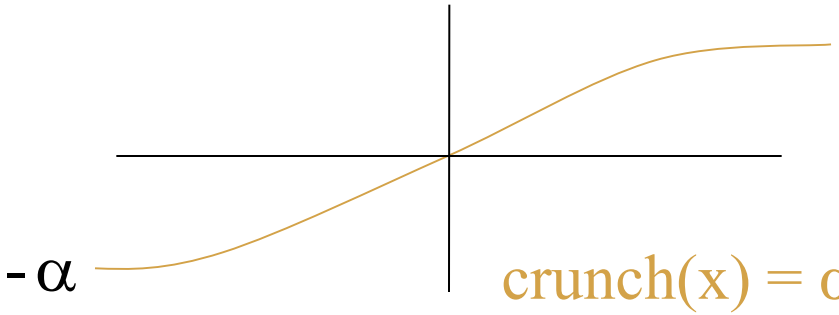
α:色度变化映射到亮度变化

·Lab空间内，L分量范围 [0, 100]

·a分量范围 [-500, 500], b分量范围[-200, 200]

·需要将色度差异映射到 [-α，α] 供之后求解用

α : 色度变化映射到亮度变化



$crunch(x) = \alpha * \tanh(x/\alpha)$

$\alpha = 5$

$\alpha = 10$

$\alpha = 25$

α的效果

$\alpha = 5$

$\alpha = 15$

$\alpha = 25$

$\alpha = 35$

$\alpha = 45$

$\alpha = 55$

$\alpha = 65$

$\alpha = 75$

$\alpha = 85$

$\alpha = 95$

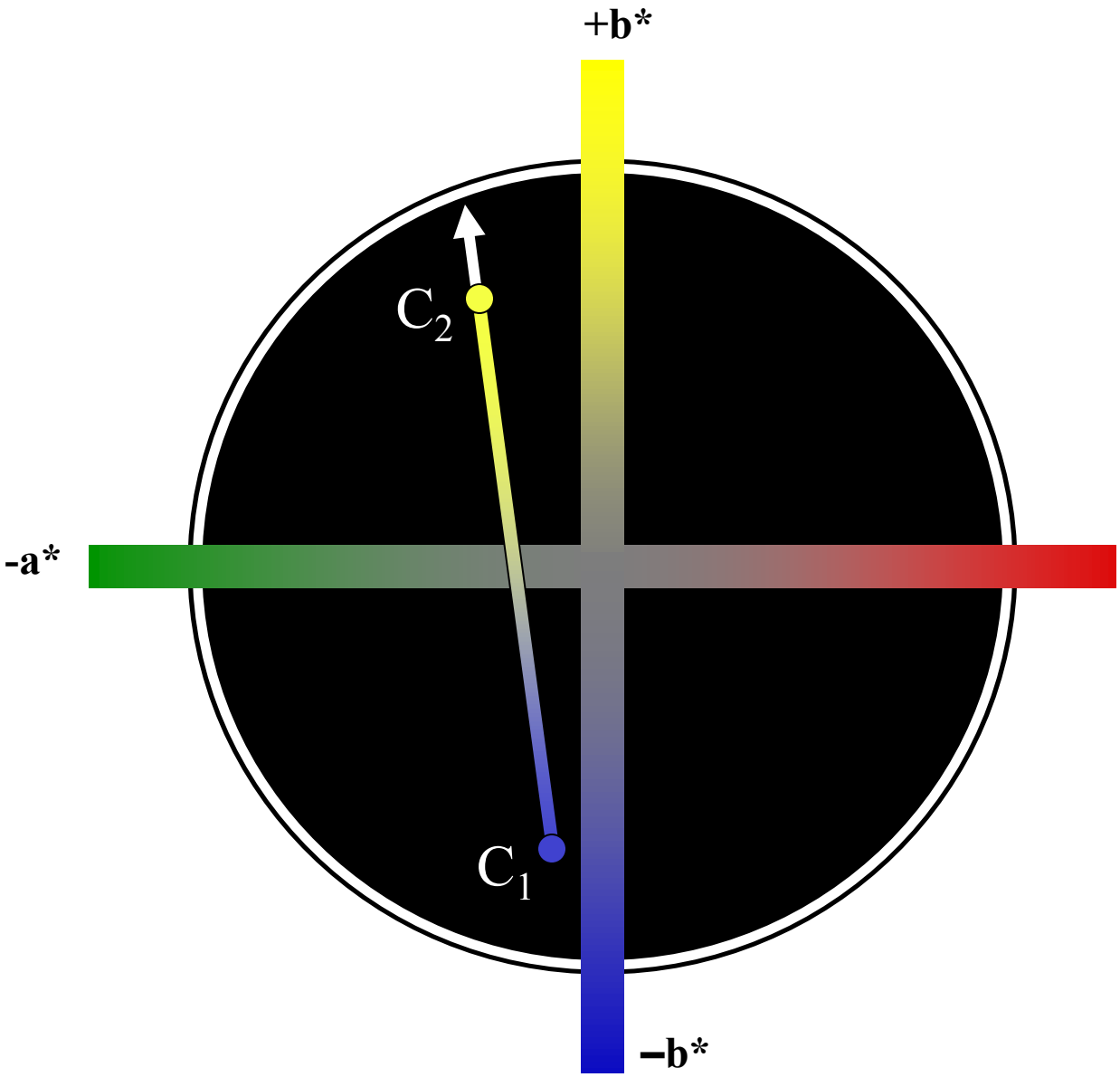
θ:决定色度差异对亮度的影响

亮度差异：$\Delta L_{ij} = L_i - L_j$

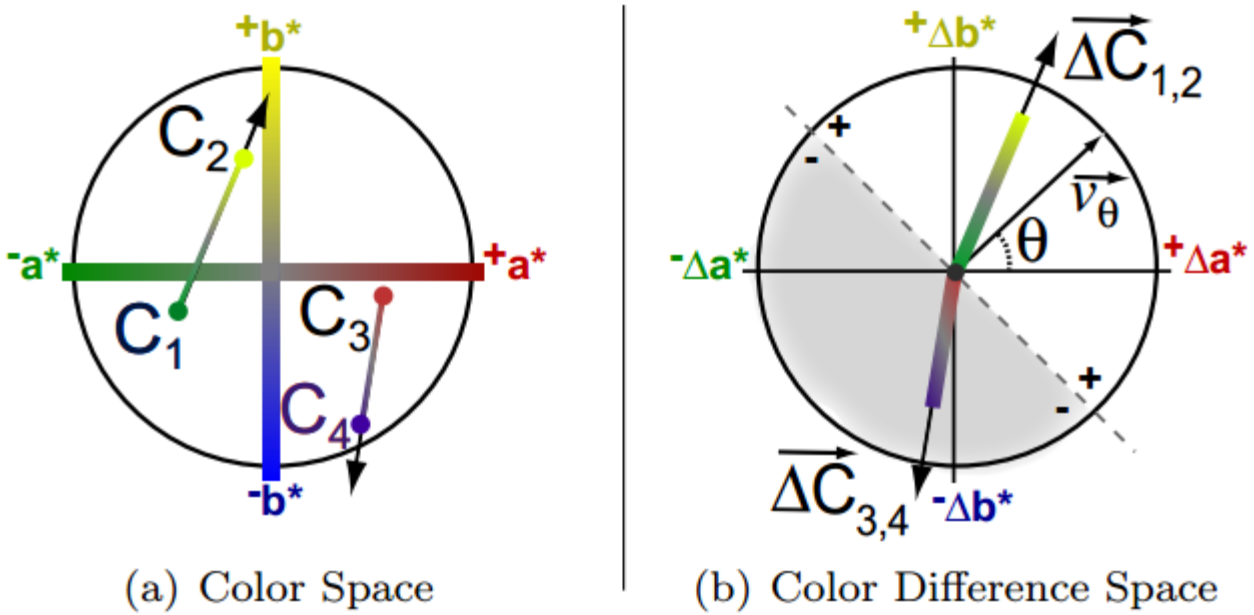
色度差异：$\| \Delta \vec{C}_{ij} \|$

$\Delta \vec{C}_{ij} = \left(\Delta A_{ij}, \Delta B_{ij} \right) \quad \Delta A_{ij} = A_i - A_j$

问题：色度差异值是无符号的


+b*
-a*
C2
C1
−b*

θ：决定色度差异对亮度的影响

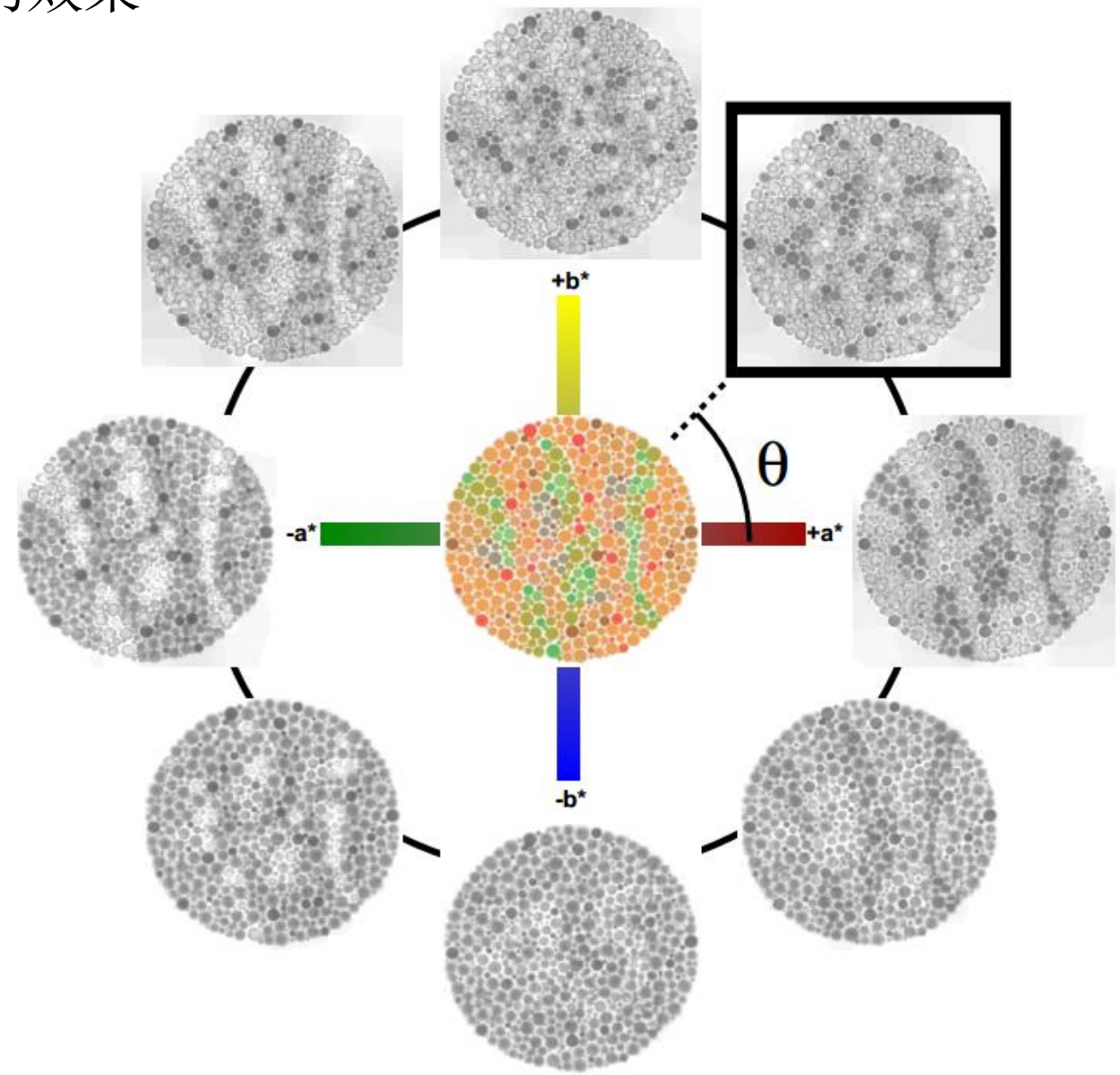


(a) Color Space　　(b) Color Difference Space

确定符号：$\overrightarrow{\Delta C_{ij}} \cdot \overrightarrow{\nu_\theta}$

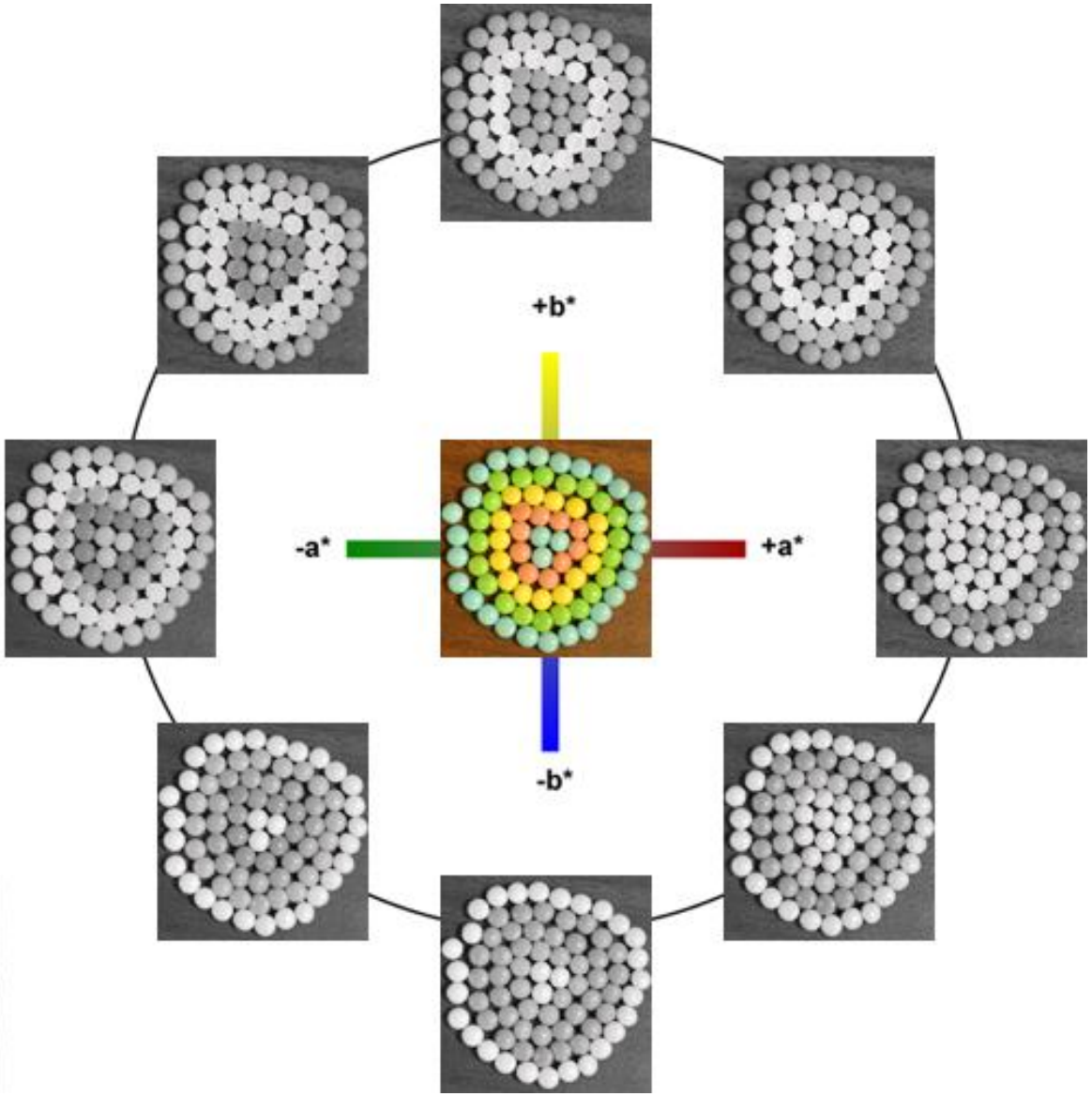
$$\overrightarrow{\nu_\theta} = (\cos\theta, \sin\theta)$$

不同θ的效果

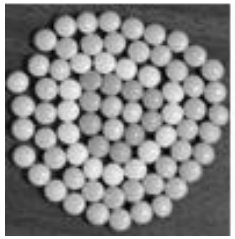


不同θ的效果

Rasche et al.

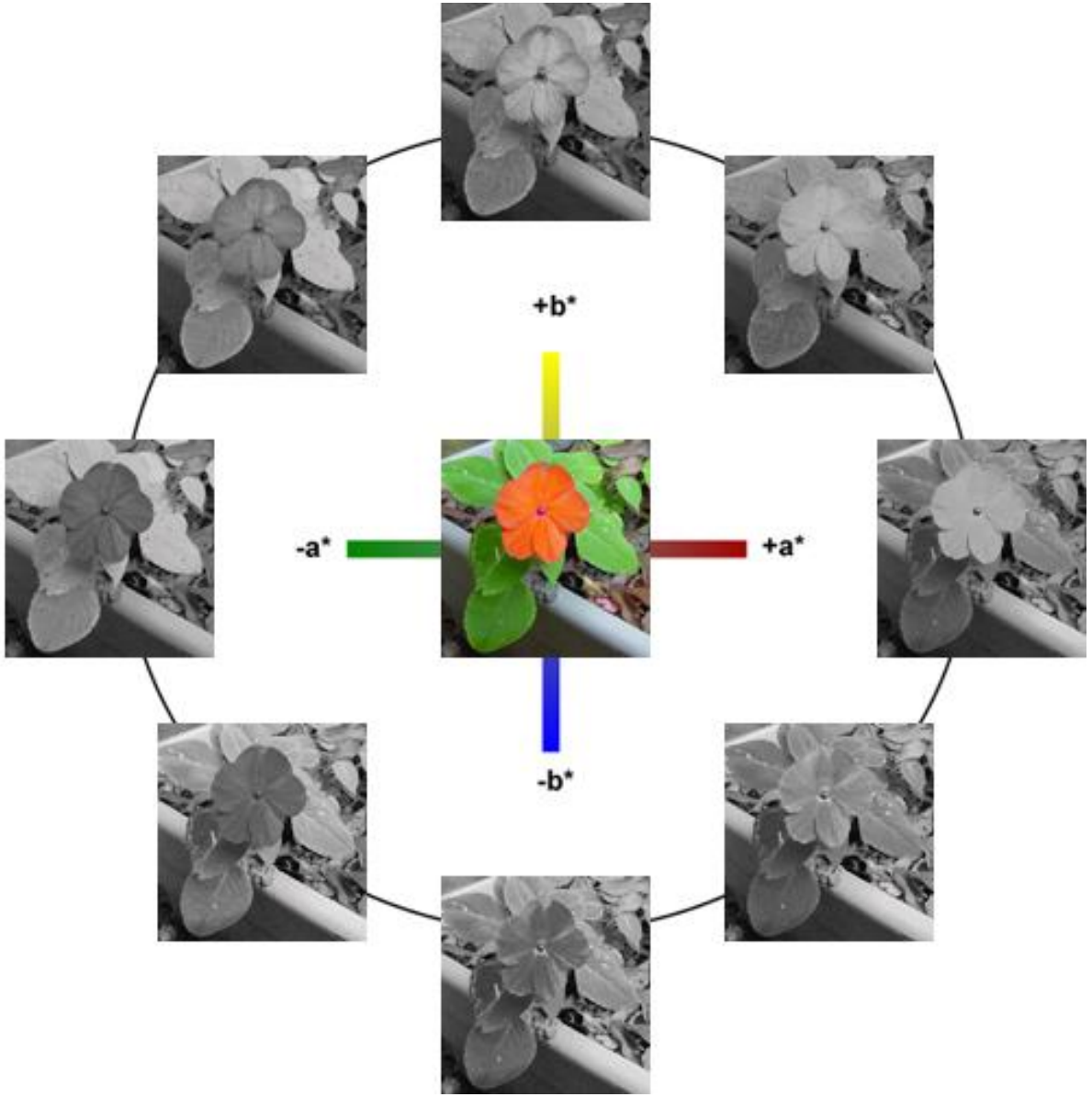


Photoshop



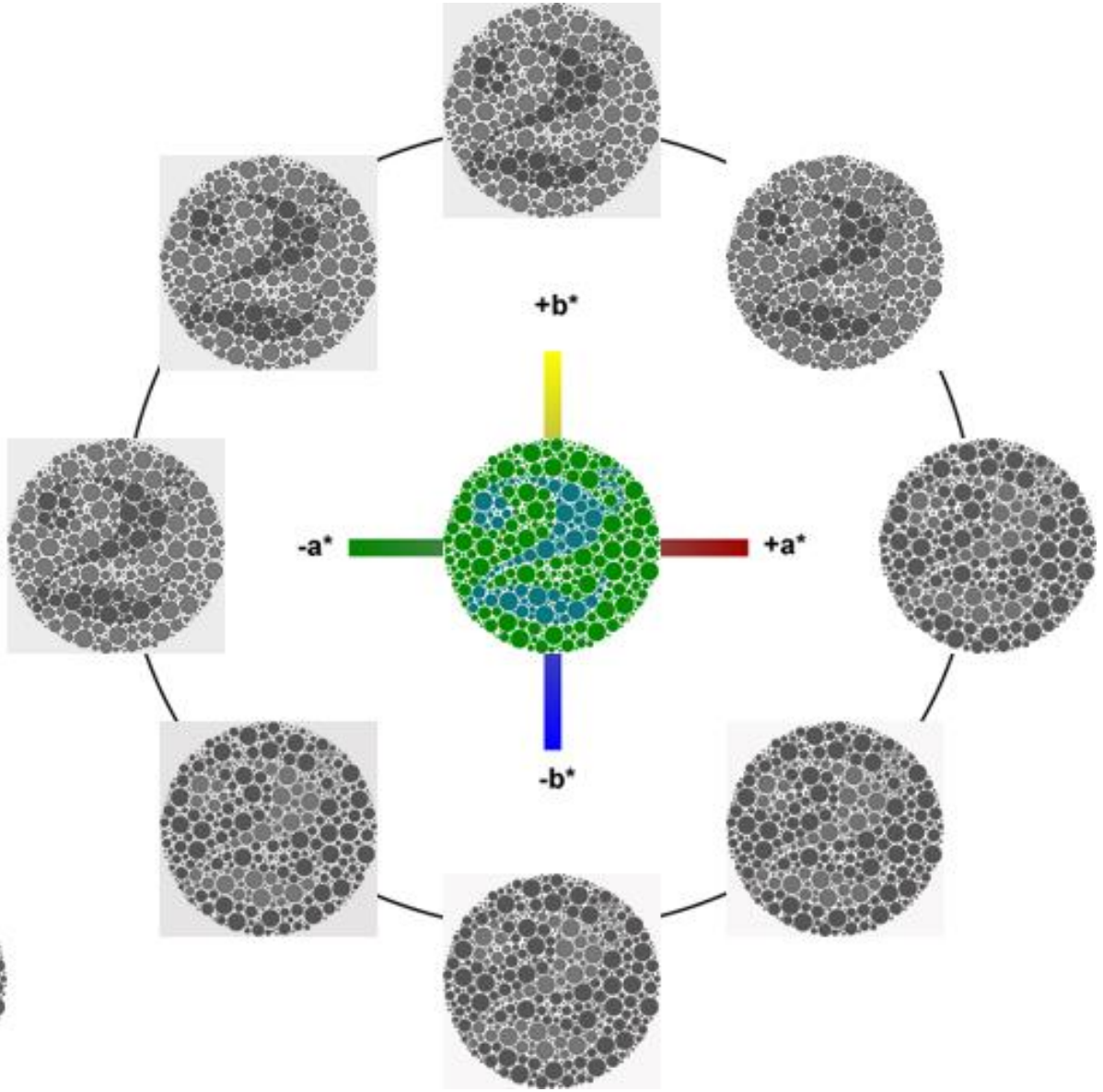
不同θ的效果

Rasche et al.

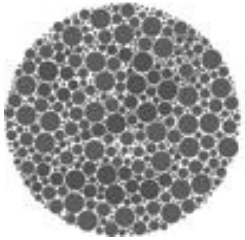


Photoshop

不同θ的效果



Photoshop



最终方程

Given:

$$\begin{aligned} \text{crunch}(x) &= \alpha * \tanh(x/\alpha) \\ \overrightarrow{v_\theta} &= (\cos\theta, \sin\theta) \end{aligned}$$

then:

$$\delta(\alpha,\theta)_{ij} = \begin{cases} \Delta L_{ij} & \text{if } |\Delta L_{ij}| > \text{crunch}(||\overrightarrow{\Delta C_{ij}}||) \\ \text{crunch}(||\overrightarrow{\Delta C_{ij}}||) & \text{if } \overrightarrow{\Delta C_{ij}} \cdot \overrightarrow{v_\theta} \geq 0 \\ \text{crunch}(-||\overrightarrow{\Delta C_{ij}}||) & \text{otherwise.} \end{cases}$$

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

效率

图像大小	Athlon 64 3200 CPU	NVIDIA GeForce GT6800
100 * 100	12.7 s	2.8 s
150 * 150	65.6 s	9.7 s
200 * 200	204.0 s	25.7 s

结果

Photoshop Gray

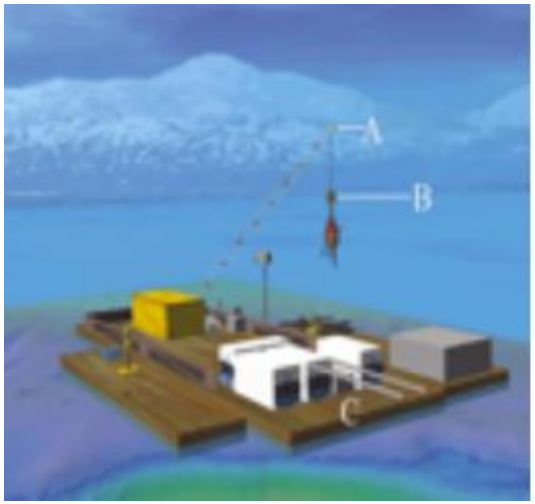
Color2Gray

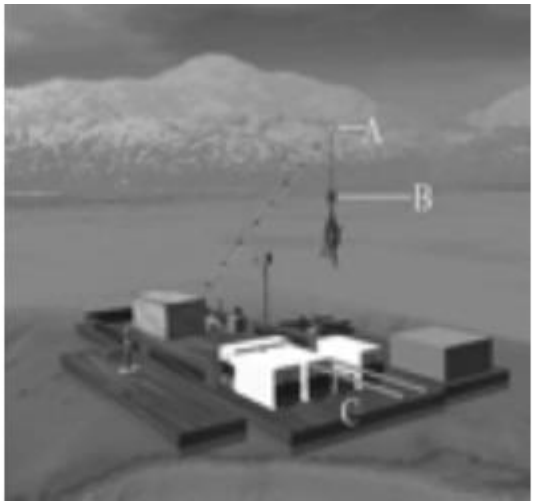
结果

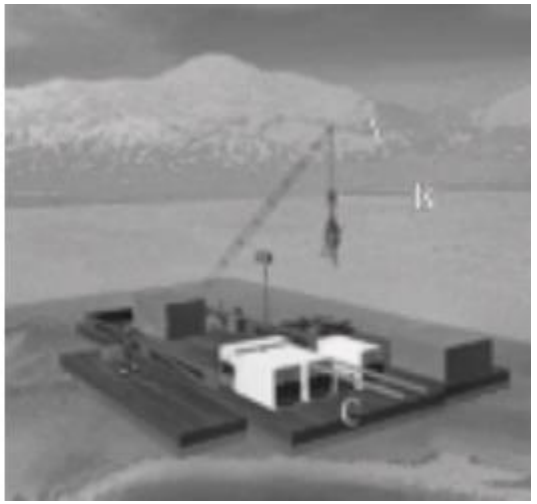
Photoshop Gray

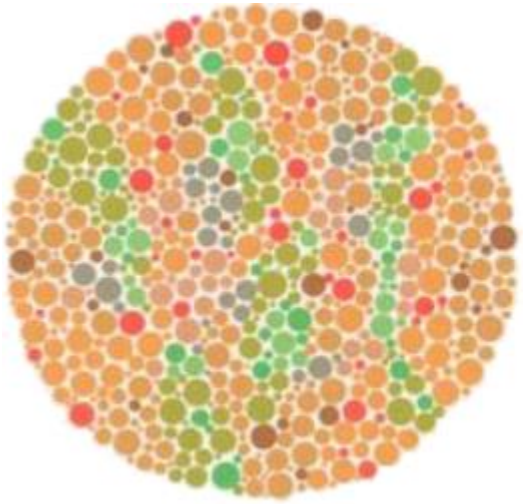
Color2Gray

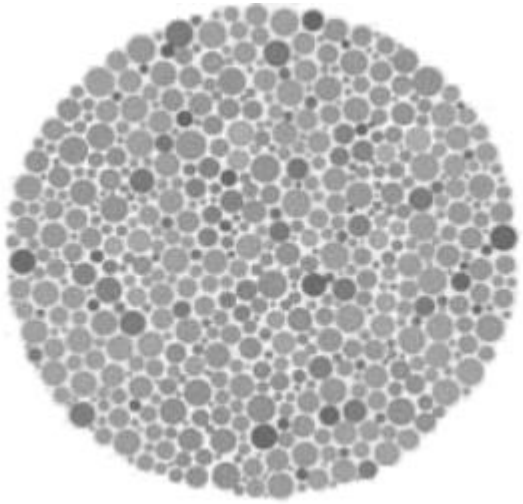
结果



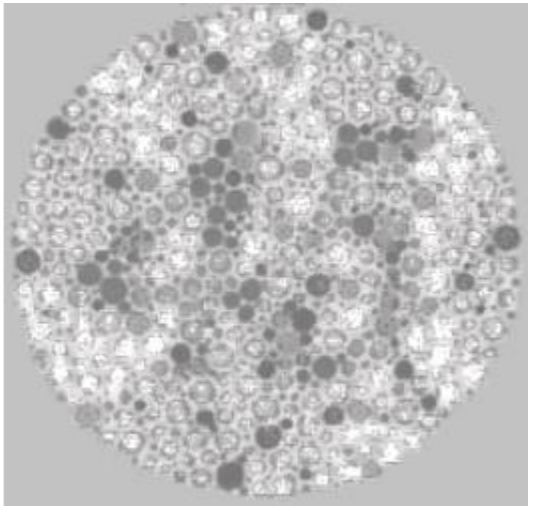






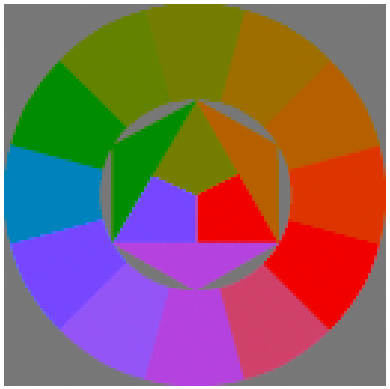


Photoshop Gray

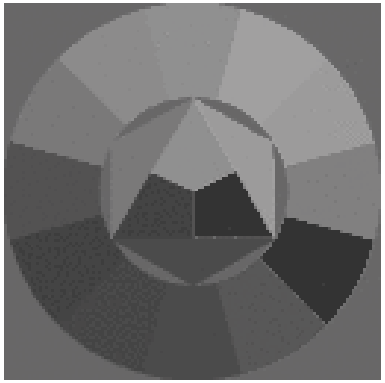


Color2Gray

结果



Photoshop Gray



Color2Gray

结果

Color2Gray

Color2Gray+Color

结果

原图

Photoshop

Color2Gray

结果

原图

Photoshop

Color2Gray+Color

结果

原图

Photoshop

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。

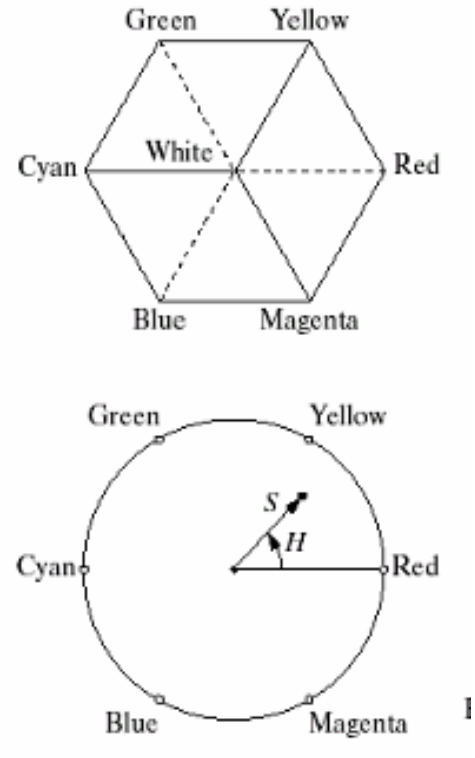
有经验的艺术家往往凭借经验或直觉来选择合适的颜色集，然后借助某些工具手动交互的调整图片的颜色搭配。

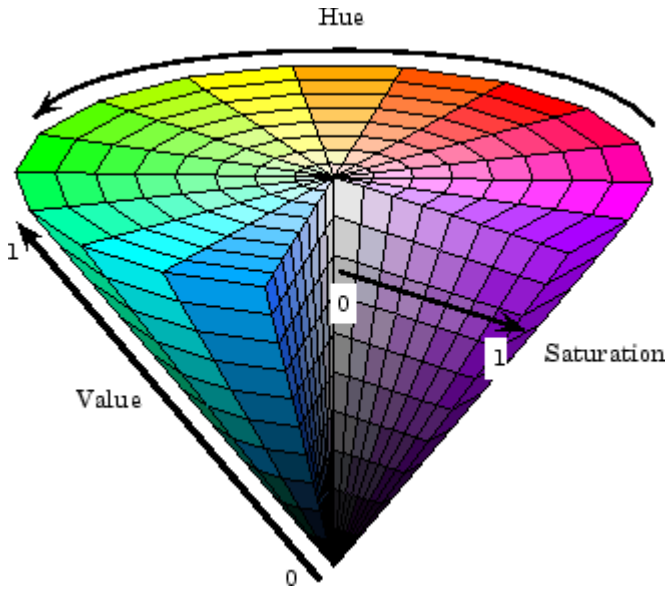
Color Harmony ?

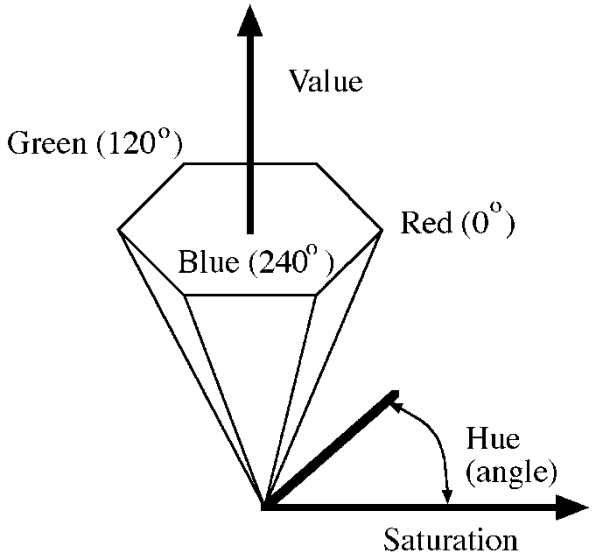
original image

our method（automatically）

HSV

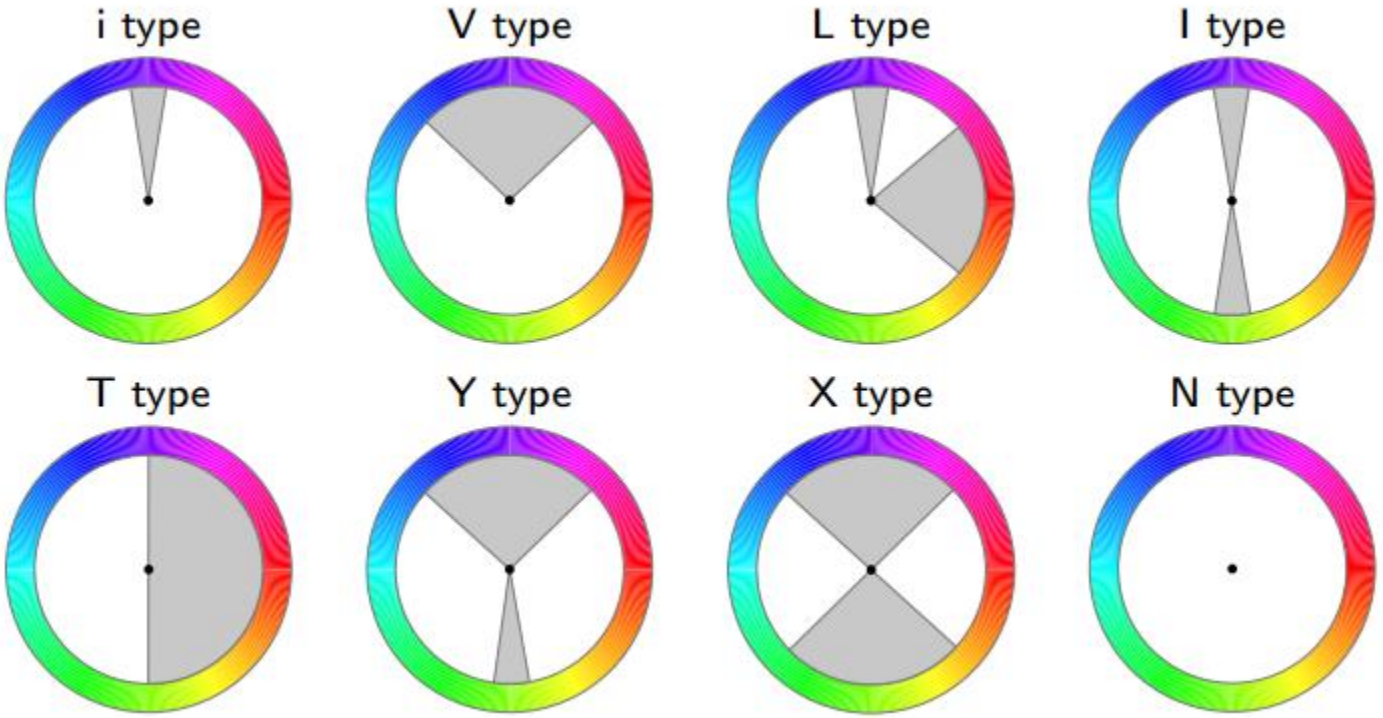






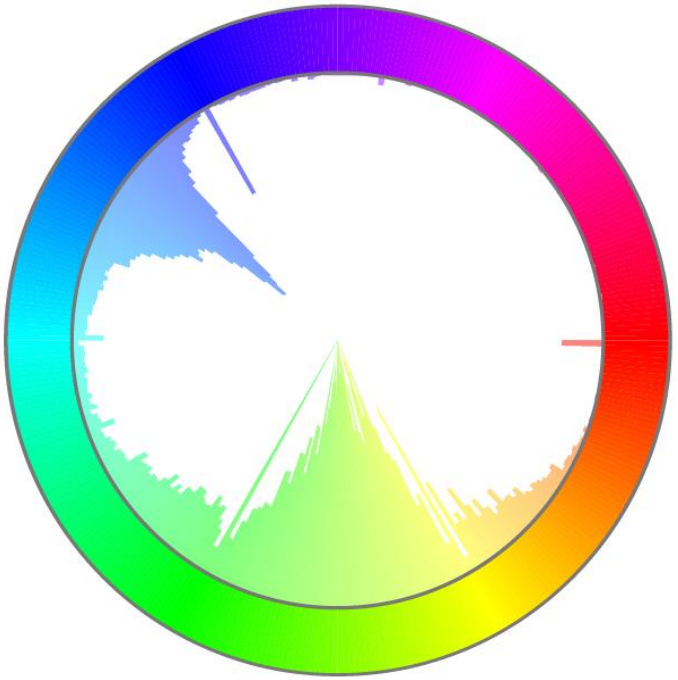
Hue, Saturation, Value

HSV色轮



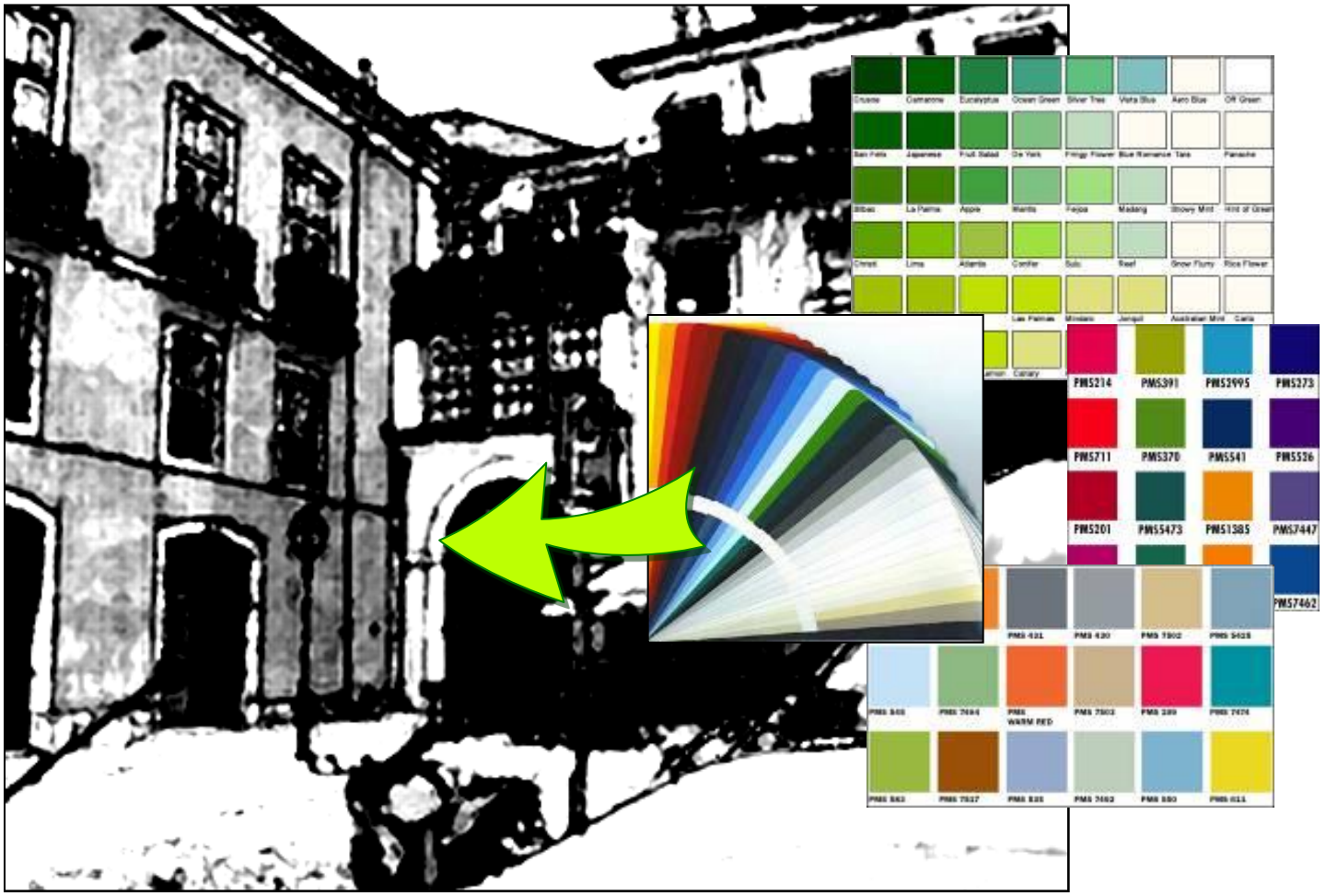
根据经验得出的符合Harmony的颜色分布

HSV色轮

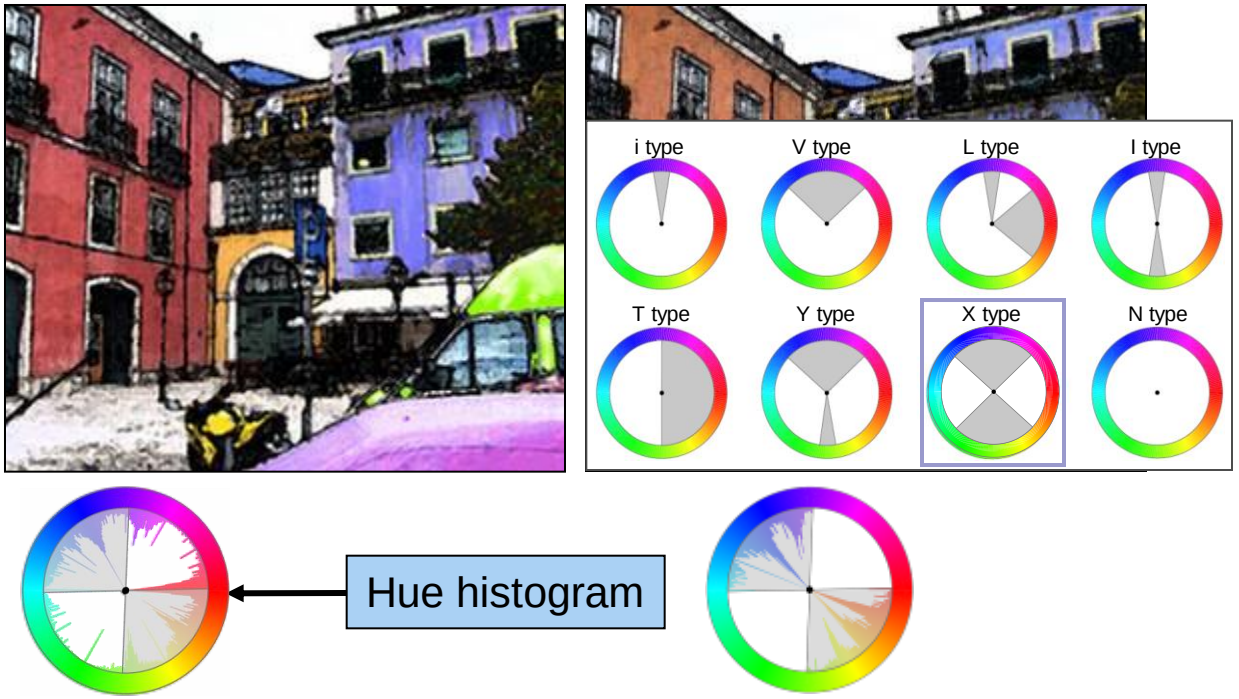


H + S

上色/重上色


PMS214
PMS391
PMS3995
PMS373
PMS711
PMS370
PMS541
PMS526
PMS201
PMS5473
PMS1385
PMS7447
PMS7462

Harmonization

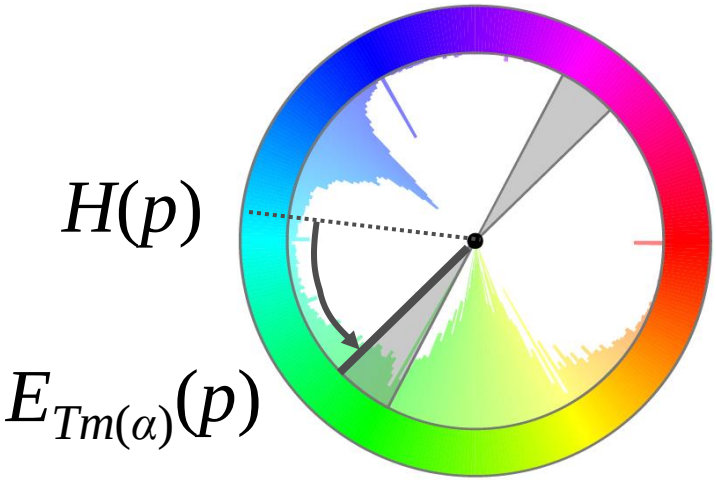


Harmony 方程

弧长

像素p色度　最近扇区边缘　像素p饱和度

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

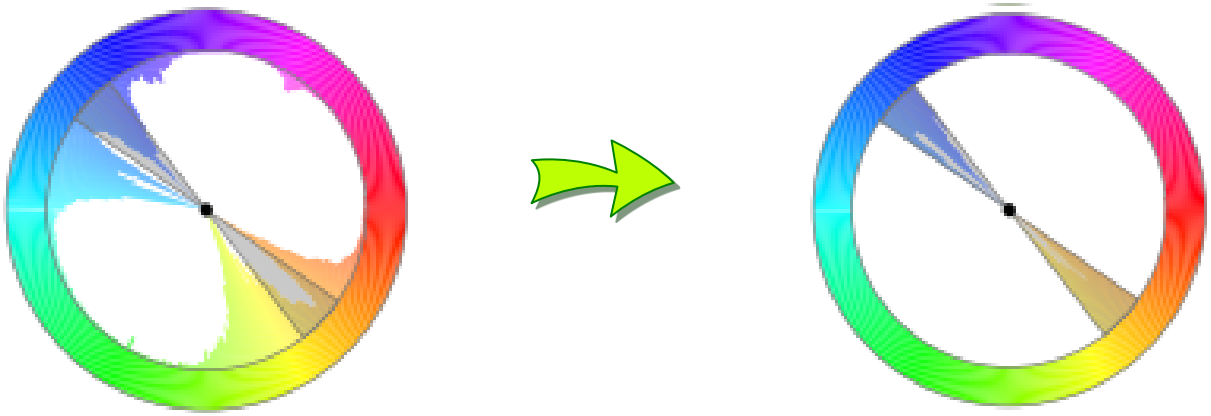


最佳harmony模板

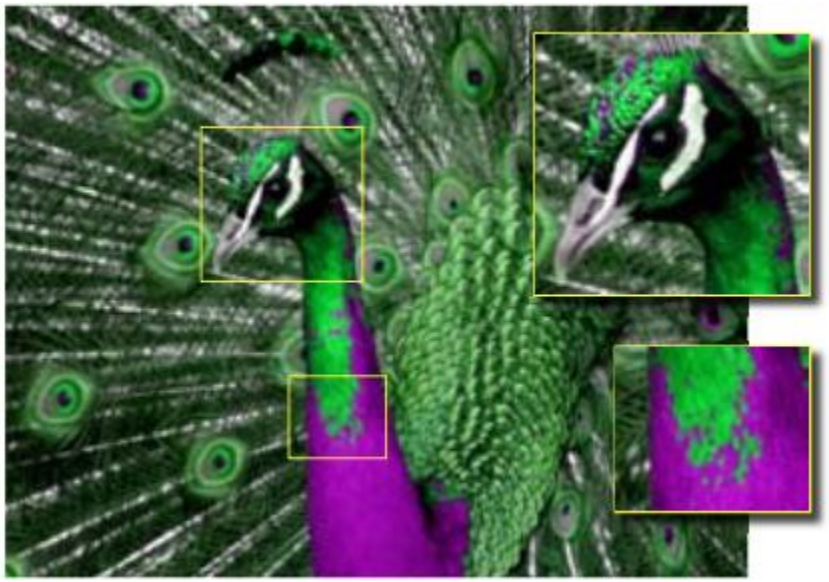
对每个模板 T_m，根据 $F\left(X,\left(T_m,\alpha\right)\right)$ 计算最佳角度 α，寻找最佳匹配，之后重新上色。

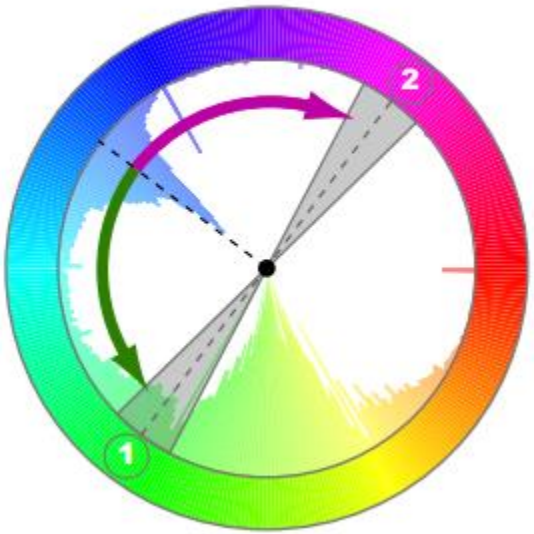
$$\left(T_{m_0},\alpha_0\right)=\arg\min_{(m,\alpha)} F\left(X,\left(T_m,\alpha\right)\right)$$

结果



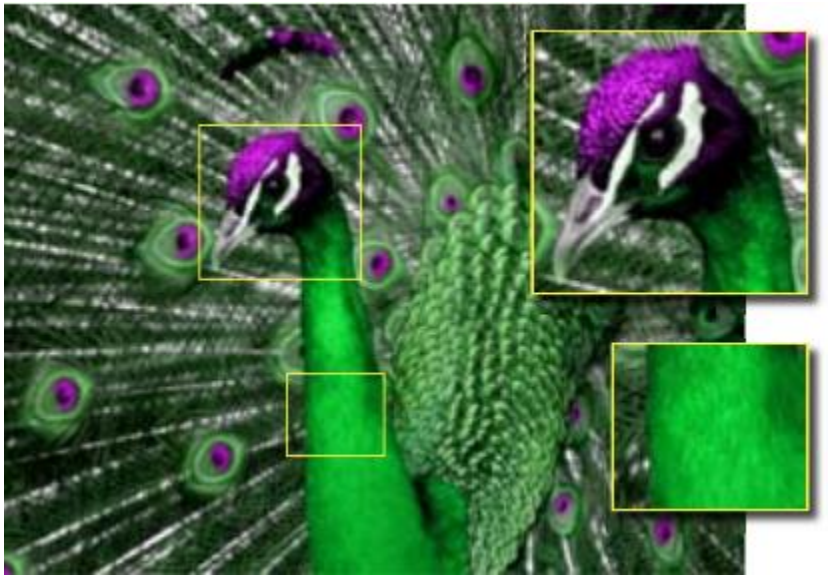
问题





不连续，蓝色区域的像素有的shift到1，有的shift到 2

Graph-cut

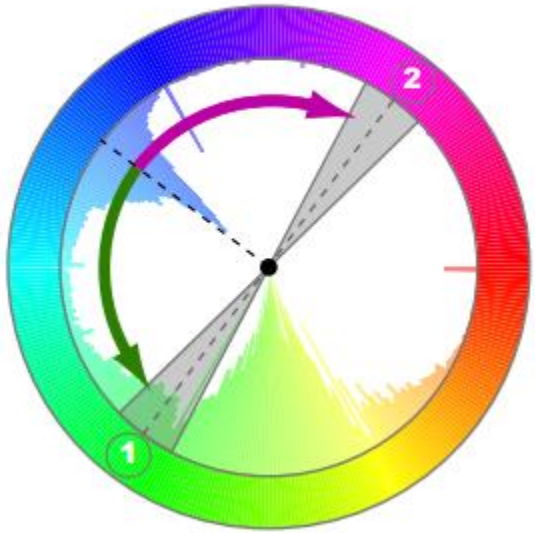


Graph-cut 优化

Graph-cut

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

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



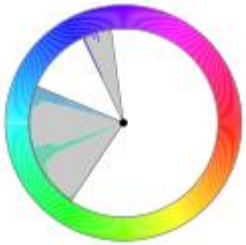
Graph-cut

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

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

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

结果



原图

选择不同模板进行重上色

结果












原图　　好模板　　差模板

结果

原背景

根据前景调整背景

根据国旗的配色类型，重上色图片


interaction
face
2B0ST0N6

原图


interaction
face
2B0ST0N6

反L型


interaction
face
2B0ST0N6

V型


interaction
face
2B0ST0N6

Y型

Thank you!