Style-aware robust color transfer

H. Hristova O. Le Meur R. Cozot K. Bouatouch

IRISA - University of Rennes 1

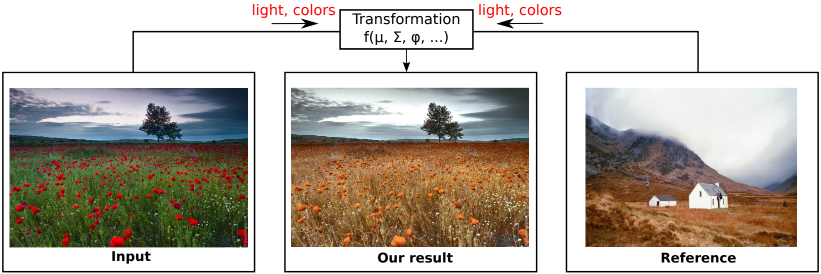
Outline

Outline

Color transfer domain

Objective of color transfer

Color transfer aims at modifying the look of an input image considering the illumination and the color palette of a reference image (e.g., considering statistical characteristics of the light and color distributions, histograms, gradients, etc.).



State-of-the-art global methods

- [Reinhard et al., 2001]
  - Linear mapping
  - L$\alpha\beta$
  - Diagonal covariance matrix
  - Gaussian distribution
- [Pitié et al., 2007]
  - Linear mapping as a solution to Monge-Kantorovich optimization problem
  - CIE Lab
  - Non-diagonal covariance matrix
  - Gaussian distribution

State-of-the-art local methods

- [Tai et al., 2005]
  - Linear mapping: [Reinhard et al., 2001]
  - L$\alpha\beta$
  - 3D image clustering
  - Cluster mapping: maps the closest clusters in terms of their luminance channel values
  - Gaussian distribution (cluster-wise)
- [Bonneel et al., 2013]
  - Linear mapping: [Pitié et al., 2007]
  - CIE Lab
  - Image clustering in terms of the luminance channel
  - Cluster mapping: shadows to shadows, midtones to midtones, highlights to highlights
  - Gaussian distribution (cluster-wise)

Limitations

Limitations

Style inconsistency

Limitations

Style inconsistency

Limitations

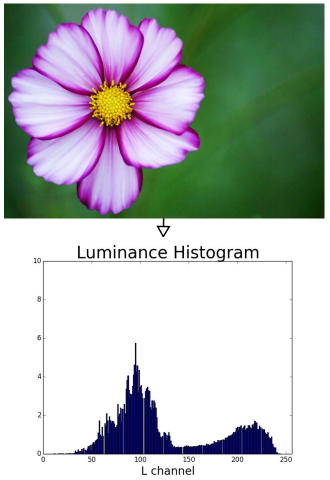
Style inconsistency

Gaussian distribution

Limitations

Style inconsistency

Gaussian distribution



Limitations

Style inconsistency

Gaussian distribution

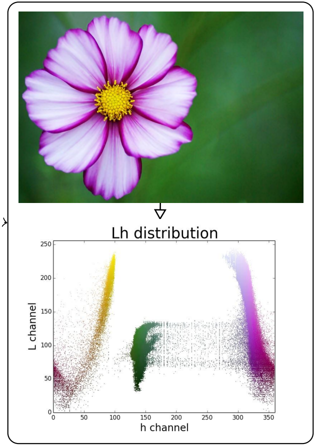
Clustering on the same color space components for input/reference images

Limitations

Style inconsistency

Gaussian distribution

Clustering on the same color space components for input/reference images



Limitations

Style inconsistency

Gaussian distribution

Clustering on the same color space components for input/reference images

Over-saturation or lack of contrast

Limitations

Style inconsistency

Gaussian distribution

Clustering on the same color space components for input/reference images

Over-saturation or lack of contrast

[Pitie et al., 2007]

[Reinhard et al., 2001]

Limitations

- Style inconsistency
- Gaussian distribution
- Clustering on the same color space components for input/reference images
- Over-saturation or lack of contrast

Outline

Our objective

Transferring the light and color features of a reference image to an input image with respect to the reference style features. Obtaining a naturally looking image which style features are as close as possible to those of the reference style.

Style notion in our context

In our method, style in images is characterized by two features: color and light.

Example

Input

Reference

Example

[Pitie et al., 2007]

Input

Reference

Example

[Pitie et al., 2007]

Input

Reference

[Bonneel et al., 2013]

Example

[Pitie et al., 2007]

Input

Reference

[Bonneel et al., 2013]

Outline

- Classification of the input and reference images
  - Light-based style images
  - Colors-based style images

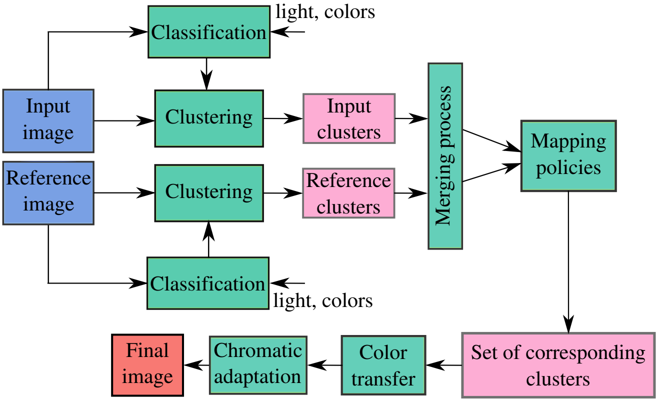
- Classification of the input and reference images
    - Light-based style images
    - Colors-based style images
- Two ways of clustering:
    - Luminance channel
    - Luminance-Hue distributions

- Classification of the input and reference images
    - Light-based style images
    - Colors-based style images
- Two ways of clustering:
    - Luminance channel
    - Luminance-Hue distributions
- Four mapping policies: Light to Light, Light to Colors, Colors to Light, Colors to Colors

- Classification of the input and reference images
    - Light-based style images
    - Colors-based style images
- Two ways of clustering:
    - Luminance channel
    - Luminance-Hue distributions
- Four mapping policies: Light to Light, Light to Colors, Colors to Light, Colors to Colors
- Evaluation of the results
    - Subjective user study
    - Objective metrics

Outline

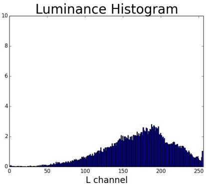
Framework

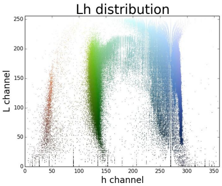

Classification
light, colors
Input image
Clustering
Input clusters
Merging process
Mapping policies
Reference image
Clustering
Reference clusters
Classification
light, colors
Final image
Chromatic adaptation
Color transfer
Set of corresponding clusters

Automatic image classification system

Colors-based style images

Images which color information is sufficient enough to well-define at least two different and significant colors.

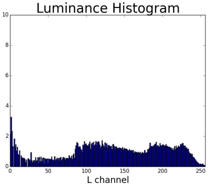


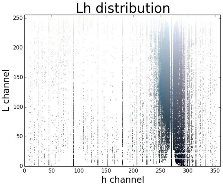


Automatic image classification system

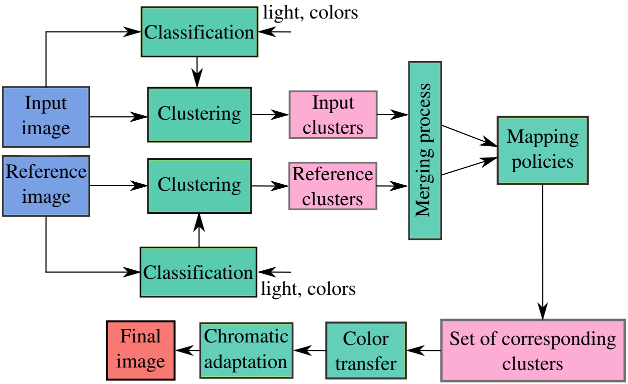
Light-based style images

Images which are not classified as colors-based style images are classified as light-based style images. Their light features are more meaningful than their color features.

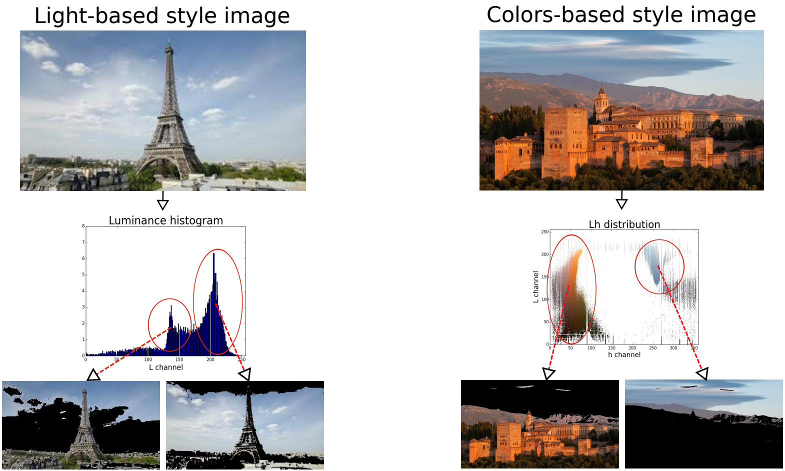




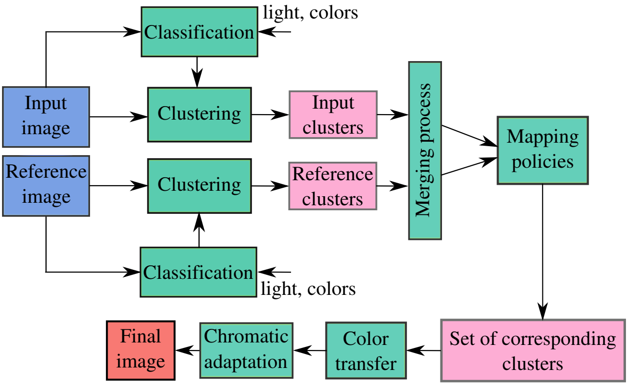
Framework


Classification
light, colors
Input image
Clustering
Input clusters
Merging process
Mapping policies
Reference image
Clustering
Reference clusters
Classification
light, colors
Final image
Chromatic adaptation
Color transfer
Set of corresponding clusters

Clustering: Gaussian mixture models



Framework


Classification
light, colors
Input image
Clustering
Input clusters
Merging process
Mapping policies
Reference image
Clustering
Reference clusters
Classification
light, colors
Final image
Chromatic adaptation
Color transfer
Set of corresponding clusters

Pseudo code algorithm

for $k = 1, \ldots, N$ **do**
  if Light to Colors **then**
    I_{Lab} := FindDarkestCluster()
    J_{Lab} := FindColdestCluster()
  end if
  if Colors to Light **then**
    I_{Lab} := FindColdestCluster()
    J_{Lab} := FindDarkestCluster()
  end if
  if Light to Light **then**
    I_{Lab} := FindDarkestCluster()
    J_{Lab} := FindDarkestCluster()
  end if
  if Colors to Colors **then**
    $[I_{Lab}, J_{Lab}]$:= FindMinDistPair()
  end if
  O_{Lab} := PerformTranspOnAB(I_{Lab}, J_{Lab})
  Exclude I_{Lch}
  Exclude J_{Lch}
end for
O_{rgb}^{final} := CATLocal(O_{rgb}, J_{rgb})

- Darkest cluster - the cluster which centroid has the minimum luminance value
- Coldest cluster - the cluster which centroid has the maximum hue value

Mapping policies

```
for k = 1, ..., N do
    if Light to Colors then
        I_Lab := FindDarkestCluster()
        J_Lab := FindColdestCluster()
    end if
    if Colors to Light then
        I_Lab := FindColdestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Light to Light then
        I_Lab := FindDarkestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Colors to Colors then
        [I_Lab, J_Lab] := FindMinDistPair()
    end if
    O_Lab := PerformTranspOnAB(I_Lab, J_Lab)
    Exclude I_Lch
    Exclude J_Lch
end for
O_rgb^final := CATLocal(O_rgb, J_rgb)
```

- Darkest cluster - the cluster which centroid has the minimum luminance value
- Coldest cluster - the cluster which centroid has the maximum hue value
- Light to Colors
    - Input light-based style image
    - Reference colors-based style image

Mapping policies

```
for k = 1, ..., N do
    if Light to Colors then
        I_Lab := FindDarkestCluster()
        J_Lab := FindColdestCluster()
    end if
    if Colors to Light then
        I_Lab := FindColdestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Light to Light then
        I_Lab := FindDarkestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Colors to Colors then
        [I_Lab, J_Lab] := FindMinDistPair()
    end if
    O_Lab := PerformTranspOnAB(I_Lab, J_Lab)
    Exclude I_Lch
    Exclude J_Lch
end for
O_rgb^final := CATLocal(O_rgb, J_rgb)
```

- Darkest cluster - the cluster which centroid has the minimum luminance value
- Coldest cluster - the cluster which centroid has the maximum hue value
- Colors to Light
    - Input colors-based style image
    - Reference light-based style image

Mapping policies

```
for k = 1, ..., N do
    if Light to Colors then
        I_Lab := FindDarkestCluster()
        J_Lab := FindColdestCluster()
    end if
    if Colors to Light then
        I_Lab := FindColdestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Light to Light then
        I_Lab := FindDarkestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Colors to Colors then
        [I_Lab, J_Lab] := FindMinDistPair()
    end if
    O_Lab := PerformTranspOnAB(I_Lab, J_Lab)
    Exclude I_Lch
    Exclude J_Lch
end for
O_rgb^final := CATLocal(O_rgb, J_rgb)
```

- Darkest cluster - the cluster which centroid has the minimum luminance value
- Coldest cluster - the cluster which centroid has the maximum hue value
- Light to Light
    - Input light-based style image
    - Reference light-based style image

Mapping policies

```
for k = 1, ..., N do
    if Light to Colors then
        I_Lab := FindDarkestCluster()
        J_Lab := FindColdestCluster()
    end if
    if Colors to Light then
        I_Lab := FindColdestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Light to Light then
        I_Lab := FindDarkestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Colors to Colors then
        [I_Lab, J_Lab] := FindMinDistPair()
    end if
    O_Lab = PerformTranspOnAB(I_Lab, J_Lab)
    Exclude I_Lch
    Exclude J_Lch
end for
O_rgb^final := CATLocal(O_rgb, J_rgb)
```

- Darkest cluster - the cluster which centroid has the minimum luminance value
- Coldest cluster - the cluster which centroid has the maximum hue value
- Colors to Colors
  - Input colors-based style image
  - Reference colors-based style image

Color transformation

```
for k = 1, ..., N do
    if Light to Colors then
        I_Lab := FindDarkestCluster()
        J_Lab := FindColdestCluster()
    end if
    if Colors to Light then
        I_Lab := FindColdestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Light to Light then
        I_Lab := FindDarkestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Colors to Colors then
        [I_Lab, J_Lab] := FindMinDistPair()
    end if
    O_Lab := PerformTranspOnAB(I_Lab, J_Lab)
    Exclude I_Lch
    Exclude J_Lch
end for
O_rgb^final := CATLocal(O_rgb, J_rgb)
```

- Color transformation between each two corresponding clusters
- Carried out on the *a* and *b* channels of CIE Lab
- Closed-form solution: solution to Monge-Kantorovich's optimization problem
- Non-diagonal covariance matrix between the channels of the color space

Color transformation

```
for k = 1, . . . , N do
    if Light to Colors then
        I_Lab := FindDarkestCluster()
        J_Lab := FindColdestCluster()
    end if
    if Colors to Light then
        I_Lab := FindColdestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Light to Light then
        I_Lab := FindDarkestCluster()
        J_Lab := FindDarkestCluster()
    end if
    if Colors to Colors then
        [I_Lab, J_Lab] := FindMinDistPair()
    end if
    O_Lab := PerformTranspOnAB(I_Lab, J_Lab)
    Exclude I_Lch
    Exclude J_Lch
end for
O_rgb^final := CATLocal(O_rgb, J_rgb)
```

- Color transformation between each two corresponding clusters
- Carried out on the *a* and *b* channels of CIE Lab
- Closed-form solution: solution to Monge-Kantorovich optimization problem
- Non-diagonal covariance matrix between the channels of the color space

Chromatic adaptation

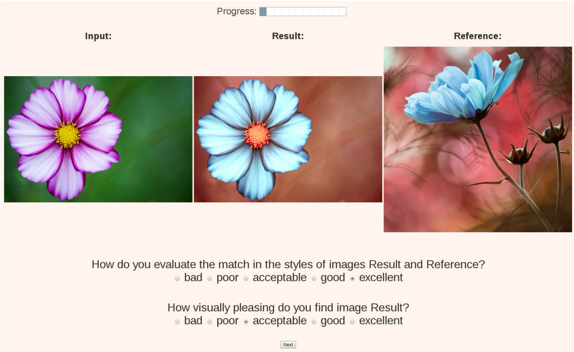
for $k = 1, \ldots, N$ **do**
    if Light to Colors **then**
        I_{Lab} := FindDarkestCluster()
        J_{Lab} := FindColdestCluster()
    end if
    if Colors to Light **then**
        I_{Lab} := FindColdestCluster()
        J_{Lab} := FindDarkestCluster()
    end if
    if Light to Light **then**
        I_{Lab} := FindDarkestCluster()
        J_{Lab} := FindDarkestCluster()
    end if
    if Colors to Colors **then**
        $[I_{Lab}, J_{Lab}]$:= FindMinDistPair()
    end if
    O_{Lab} := PerformTranspOnAB(I_{Lab}, J_{Lab})
    Exclude I_{Lch}
    Exclude J_{Lch}
end for
O_{rgb}^{final} := CATLocal(O_{rgb}, J_{rgb})

- Adapting pixel-wisely the colors of the output to the reference illuminant
- Input illuminant: "white image" obtained by performing Gaussian low-pass filter
- Reference illuminant estimation: Gray World assumption [FSDP14, HCWxW06]
- Local CAT algorithm [KJF07]
- Avoids undesired color saturation, prevents the result from becoming flat and reduces false colors in the result

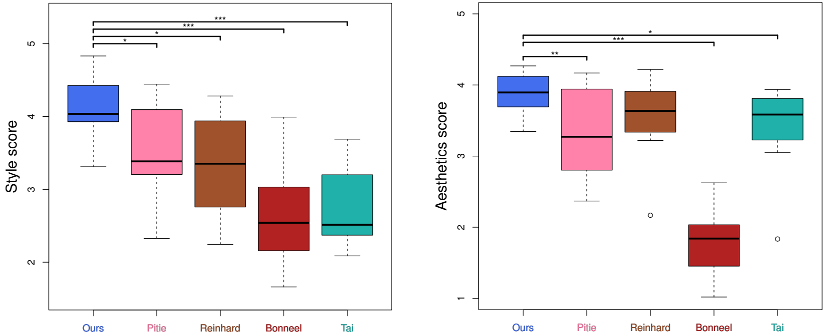
Outline

Subjective evaluation

- 15 users
- Results from 4 state-of-the-arts
- Results from our method
- Style match evaluation
- Aesthetic pleasingness evaluation
- 5-point scale



Subjective evaluation: the analysis



- Box-and-Whisker plots
- Paired T-tests
- Correlation between the two scores

Objective evaluation

SSIM

It measures the degree of artifacts in the result. It is applied between the input image and the result.

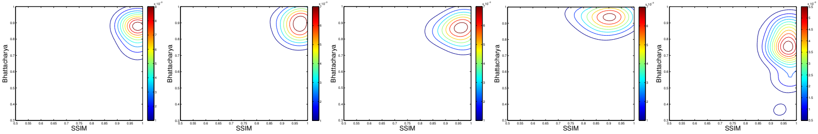
$$S(x, y) = f(c(x, y), s(x, y)) \tag{1}$$

The luminance component is removed from the computation of the metrics leaving the structural and contrast components.

Bhattacharya coefficient

Measures the distance between the result/reference histograms [ATR98] of the components of CIE Lab color space.

Objective evaluation: the analysis

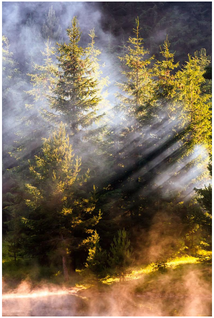


(a) Our method (b) Pitié et al. (d) Reinhard et al. (c) Bonneel at al. (e) Tai et al.

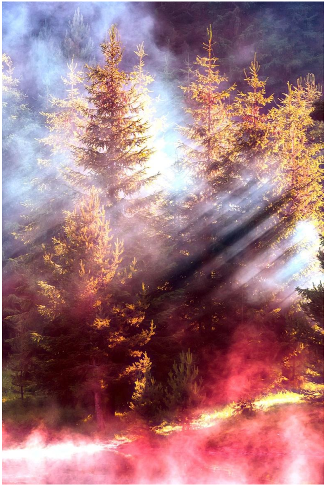
- Joint distribution of SSIM and Bhattacharya coefficient
- Paired T-tests
- (SSIM, Bhattacharya) = (1, 1): optimal for the pair

Outline

Results: Light to Colors


Input image


Reference image


Result

Results: Colors to Light


Input image


Reference image


Result

Results: Light to Light

Input image

Reference image

Result

Results: Colors to Colors


Input image


Reference image


Result

Results: classical example


Input image


Reference image


Result

Summary

- New method for style transfer for a wide class of image pairs
- Automatic image classification
- Four mappings policies
- Subjective and objective evaluations

- Future work
  - Enriching the image feature space
  - Finding a connection between the subjective user study and the objective evaluation

Thank you for your attention!

More results and information on

http://people.irisa.fr/Hristina.Hristova

References I

AHERNE F. J., THACKER N. A., ROCKETT P. I.:
The bhattacharyya metric as an absolute similarity measure for frequency coded data.
Kybernetika 34, 4 (1998), 363–368.

BONNEEL N., SUNKAVALLI K., PARIS S., PFISTER H.:
Example-based video color grading.
ACM Transactions on Graphics (Proceedings of SIGGRAPH 2013) 32, 4 (2013), 2.

FRIGO O., SABATER N., DEMOULIN V., PIERRE H.:
Optimal transportation for example-guided color transfer.
12th Asian Conference on Computer Vision (ACCV) (2014).

References II

Huo J., Chang Y., Wang J., xia Wei X.:
Robust automatic white balance algorithm using gray color points in images.
IEEE Trans. Consumer Electronics 52, 2 (2006), 541–546.
URL: http://dblp.uni-trier.de/db/journals/tce/tce52.html#HuoCWW06.

Kuang J., Johnson G. M., Fairchild M. D.:
icam06: A refined image appearance model for hdr image rendering.
Journal of Visual Communication and Image Representation 18, 5 (2007), 406–414.

References III

PITIÉ F., KOKARAM A. C., DAHYOT R.:
Automated colour grading using colour distribution transfer.
Computer Vision and Image Understanding 107, 1 (2007), 123–137.

REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.:
Color transfer between images.
Computer Graphics and Applications, IEEE 21, 5 (2001), 34–41.

SHIH Y., PARIS S., DURAND F., FREEMAN W. T.:
Data-driven hallucination of different times of day from a single outdoor photo.
ACM Transactions on Graphics (TOG) 32, 6 (2013), 200.

References IV

Tai Y.-W., Jia J., Tang C.-K.:
Local color transfer via probabilistic segmentation by expectation-maximization.
In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 1, IEEE, pp. 747–754.