PosterChild: Blend-Aware Artistic Posterization

Cheng-Kang Ted Chao George Mason University
Karan Singh University of Toronto
Yotam Gingold George Mason University

GEORGE MASON UNIVERSITY CraGL Creativity and Graphics Lab

UNIVERSITY OF TORONTO

Artistic Posterization

Before going deep into our approach, let me give you a small introduction on what is artistic posterization…..

‘”

Posterization is an effect in which an image with continuous colors converted into an image consisting of smooth regions of constant colors.

”’

These are manually-created and artists like to exaggerate or recolor regions in their posters

Artistic Posterization

Artistic Posterization

Artistic Posterization

Artistic Posterization

Saarland University

Here is a small example of the teaser poster from EGSR webpage created by our algorithm.

People haven’t directly dealt with this artistic posterization problem before except for [Afifi 2018]. Other previous work such as Xu and Kaplan, they tackled the similar problem but only in black-and-white images. Gerstner et al. 2013 has intermediate results of posterization in his pixelated images paper.

Saarland University

Saarland University

Saarland University

Previous Work: [Xu and Kaplan 2008], [Gerstner et al. 2013], [Afifi 2018]

Problem Statement

Automatic posterization tools such as those in Photoshop or Illustrator generate artifacts such as noise along color boundaries and color saturation if small number of layers are desired.

So, artists tend to create posterization effect by their own workflow. If you tried to google ‘posterization’, you would be able to see many kinds of tutorials. For most of the time (in the tutorials), artists will be tired of tracing enormous numbers of color boundaries and applying filtering techniques in different color regions.

This is an extremely tedious process. And if artists want to transfer this kinds of posterization style into different images, they would need to do it again….

So, a faithful and automatic posterization tool is demanding to artists.

Problem Statement

- Create a posterized image using a discrete set of colors representing the input image and also provided convenient handles for recoloring.
  - Existing automatic posterization tools produce output quite different from artists.
  - Manual approaches are time-consuming.

Problem Statement

- Create a posterized image using a discrete set of colors representing the input image and also provided convenient handles for recoloring.
  - Existing automatic posterization tools produce output quite different from artists.
  - Manual approaches are time-consuming.

Problem Statement

- Create a posterized image using a discrete set of colors representing the input image and also provided convenient handles for recoloring.
  - Existing automatic posterization tools produce output quite different from artists.
  - Manual approaches are time-consuming.

Photoshop's Posterization Filter

Problem Statement

- Create a posterized image using a discrete set of colors representing the input image and also provided convenient handles for recoloring.
  - Existing automatic posterization tools produce output quite different from artists.
  - Manual approaches are time-consuming.

Photoshop's Posterization Filter

Problem Statement

- Create a posterized image using a discrete set of colors representing the input image and also provided convenient handles for recoloring.
  - Existing automatic posterization tools produce output quite different from artists.
  - Manual approaches are time-consuming.

Photoshop's Posterization Filter

Artist's Creation

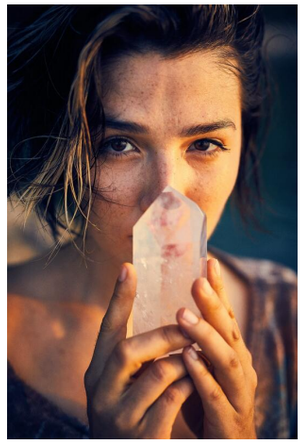
Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

Our algorithm is based on principled steps carefully designed to capture artistic design requirements.

Our Approach

- Step 1: Convex-hull based palette extraction
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries



Our algorithm is based on principled steps carefully designed to capture artistic design requirements.

Our Approach

- Step 1: Convex-hull based palette extraction
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

Our Approach

- Step 1: Convex-hull based palette extraction
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

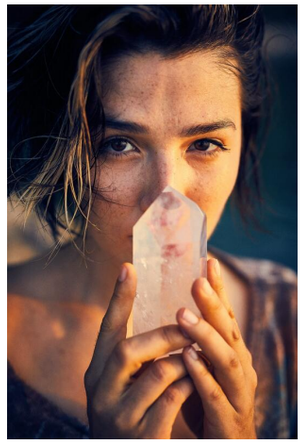
Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

Our algorithm is based on principled steps carefully designed to capture artistic design requirements.

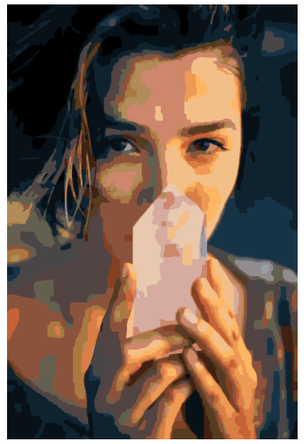
Our Approach

- Step 1: Choose a color palette
- Step 2: Rough region and color assignment
- Step 3: Improve region color blends
- Step 4: Improve region boundaries



Our Approach

- Step 1: Choose a color palette
- Step 2: Rough region and color assignment
- Step 3: Improve region color blends
- Step 4: Improve region boundaries



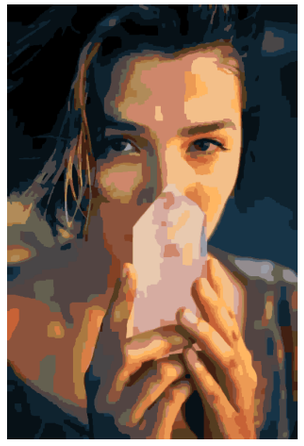
Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

Our algorithm is based on principled steps carefully designed to capture artistic design requirements.

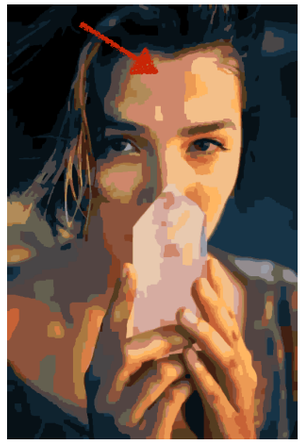
Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Blend refinement
- Step 4: Improve region boundaries



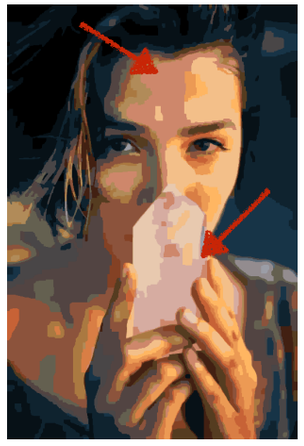
Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Blend refinement
- Step 4: Improve region boundaries



Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Blend refinement
- Step 4: Improve region boundaries



Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Blend refinement
- Step 4: Improve region boundaries

Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Blend refinement
- Step 4: Improve region boundaries

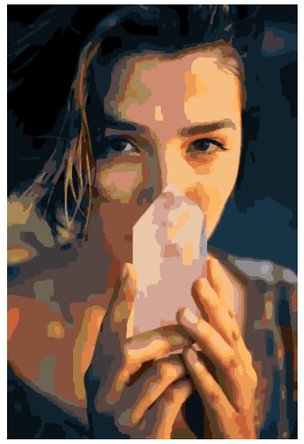
Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

Our algorithm is based on principled steps carefully designed to capture artistic design requirements.

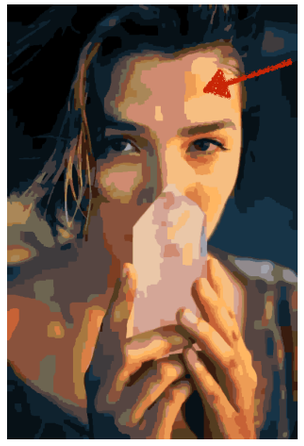
Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Region boundary smoothing



Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Region boundary smoothing



Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Region boundary smoothing

Our Approach

- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Region boundary smoothing

Step 1: Convex-hull based palette extraction

An important property of posterization is that it creates constant color regions and involves gradation of tones in continuous regions such as skin, sky, and sunsets.

This motivates us to think about finding representing colors and blending those colors to achieve the gradation of tones.

We adopt [Tan et al. 2016] convex hull simplification method to find palette colors and generate blends from the extracted palette in our next step.

Here is one of the examples showing what exactly the gradation of tones is. You can see ……

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.
- Improvement: We use K-means to reduce outlier sensitivity. See paper for details.

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.
- Improvement: We use K-means to reduce outlier sensitivity. See paper for details.

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.
- Improvement: We use K-means to reduce outlier sensitivity. See paper for details.

Step 1: Convex-hull based palette extraction

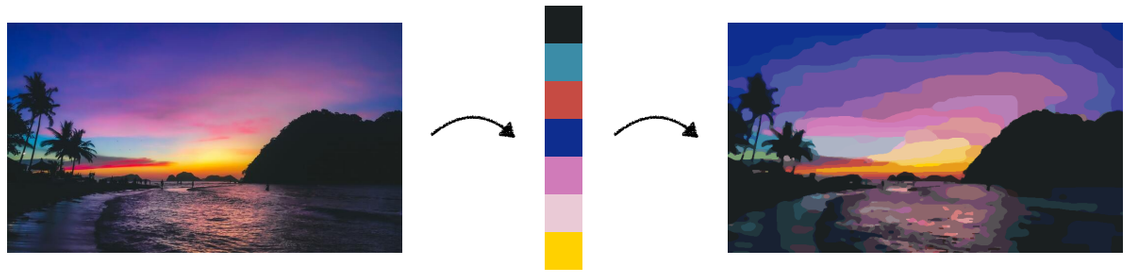
- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.
- Improvement: We use K-means to reduce outlier sensitivity. See paper for details.

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.
- Improvement: We use K-means to reduce outlier sensitivity. See paper for details.

Step 1: Convex-hull based palette extraction

- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.
- Improvement: We use K-means to reduce outlier sensitivity. See paper for details.



Step 2: Rough region and color assignment

If you look at a posterized image example in the previous slides, you could see colors are being placed in different certain regions.

So, our goal is to partition image into colored regions and at the same time the color in each region does not deviate too much from the original image and also has spatial consistency.

This problem could be solved by multi-label optimization.

There are two terms in our formulation:
The first term…… For example, if a color at a pixel in the original image is black, but you assign white in that pixel, then the cost will be square root of 3.
The second term…

You could imagine if you have 3 color labels and you would like to place those 3 color labels onto the image. Your placement is actually a labeling function and your current placement could be measured as an energy by this formulation. Our goal is to minimize this placement.

<talk a bit more about the lambda>

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

$$E_{pairwise} = \sum_{p,q \in N} \|L_p - L_q\|_2$$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

$$E_{pairwise} = \sum_{p,q \in N} \|L_p - L_q\|_2$$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

$$E_{pairwise} = \sum_{p,q \in N} \|L_p - L_q\|_2$$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

$$E_{pairwise} = \sum_{p,q \in N} \|L_p - L_q\|_2$$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

$$E_{pairwise} = \sum_{p,q \in N} \|L_p - L_q\|_2$$

- Overall objective function: $E(f) = E_{data}(f) + \lambda E_{pairwise}(f)$

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

$$E_{pairwise} = \sum_{p,q \in N} \|L_p - L_q\|_2$$

- Overall objective function: $E(f) = E_{data}(f) + \lambda E_{pairwise}(f)$
  - λ controls the clumpiness of the regions in the output.

Step 2: Rough region and color assignment

- Goal: Partition the image into regions that approximate the input (E_{data}) with spatial consistency ($E_{pairwise}$).
  - E_{data} penalizes the difference between a pixel's input color I_p and its region color R_p.

$$E_{data} = \sum_{p \in I} \|R_p - I_p\|_2$$

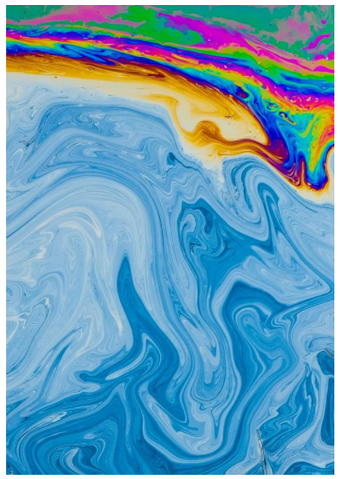
  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

$$E_{pairwise} = \sum_{p,q \in N} \|L_p - L_q\|_2$$

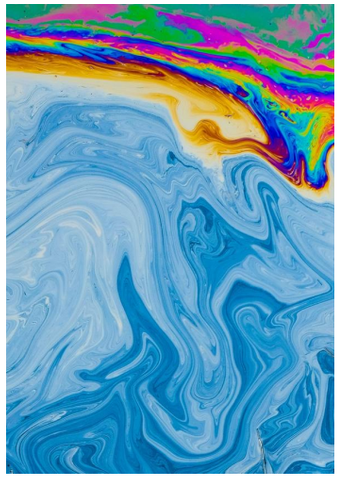
- Overall objective function: $E(f) = E_{data}(f) + \lambda E_{pairwise}(f)$
  - λ controls the clumpiness of the regions in the output.
  - We solve this problem with multi-label optimization [Boykov and Kolmogorov 2001].

Step 2: Rough region and color assignment

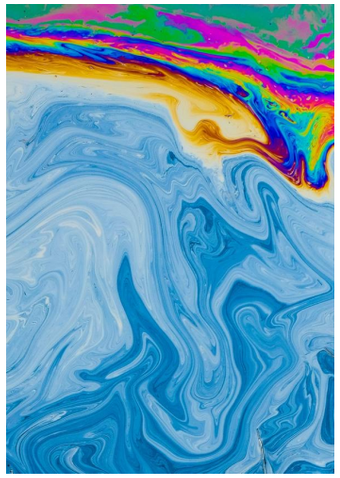
Step 2: Rough region and color assignment



Step 2: Rough region and color assignment

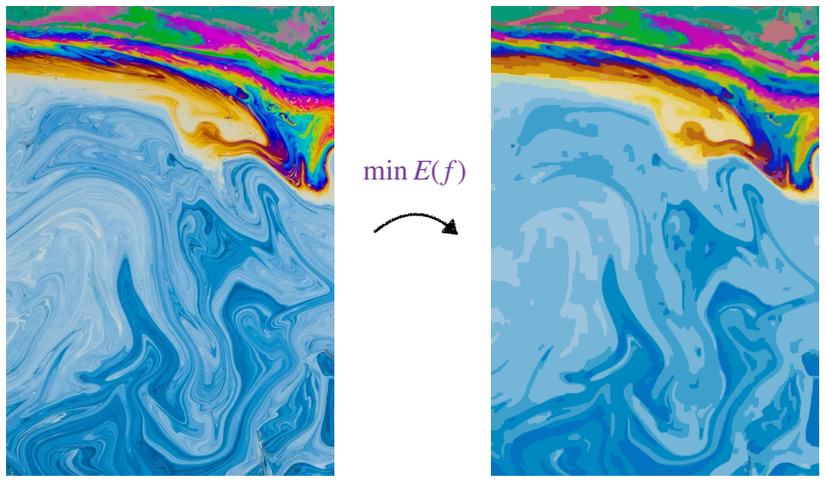


Step 2: Rough region and color assignment

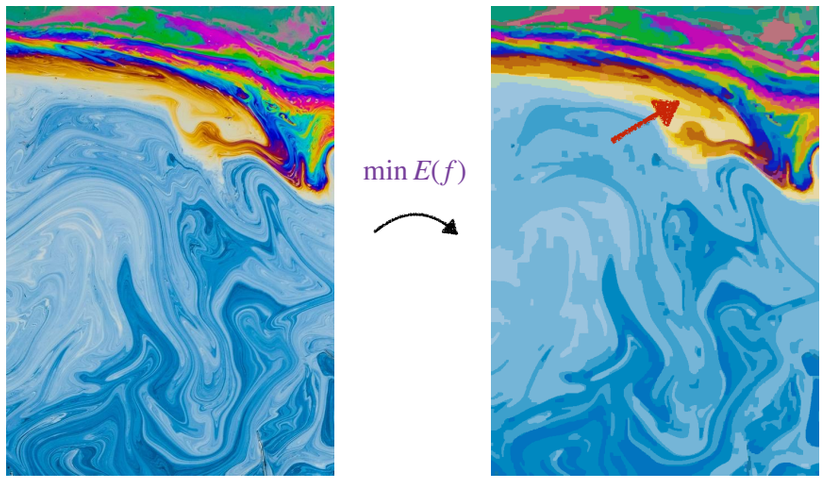


$\min E(f)$

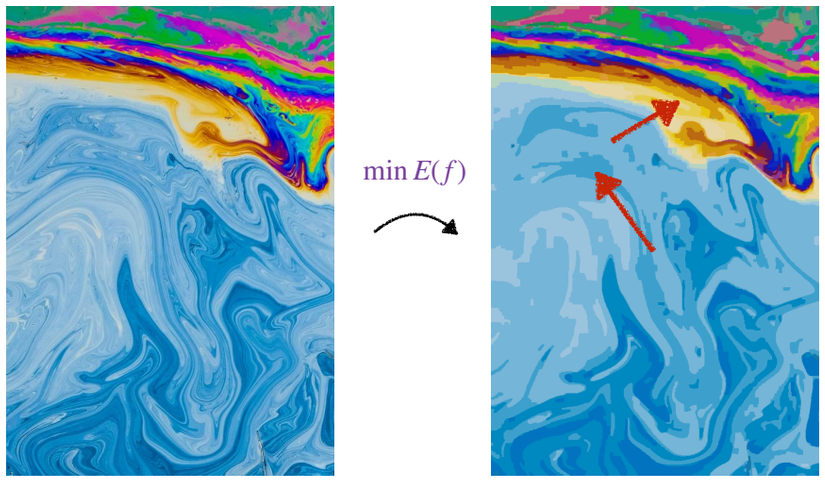
Step 2: Rough region and color assignment



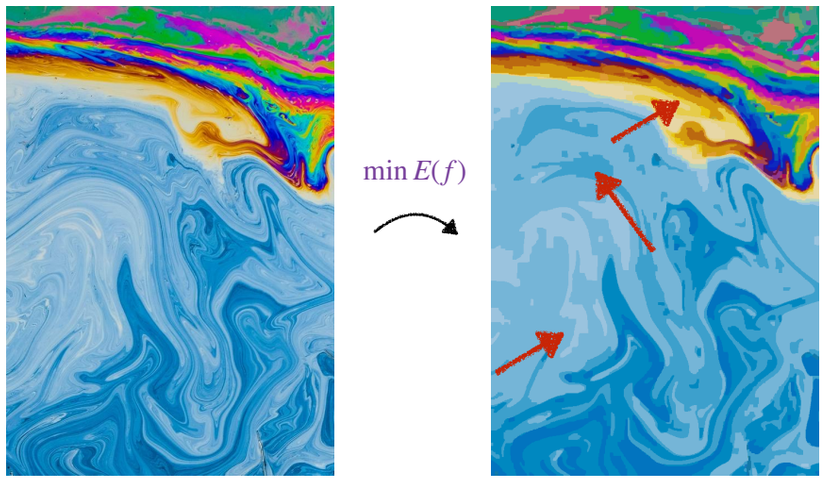
Step 2: Rough region and color assignment



Step 2: Rough region and color assignment



Step 2: Rough region and color assignment



Step 2: Rough region and color assignment

So previously we talked about labels right?

How do we create labels?

Step 2: Rough region and color assignment

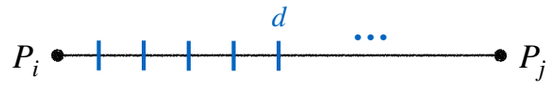
- Palette colors P from palette extraction in step 1.

Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

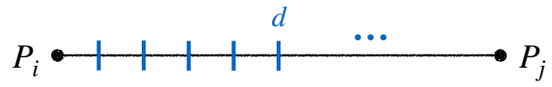
Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.



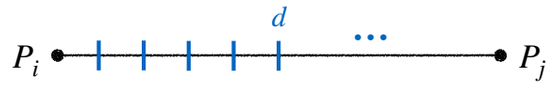
Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.



Step 2: Rough region and color assignment

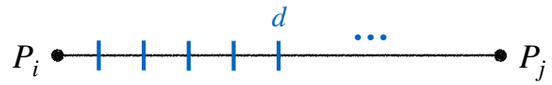
- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

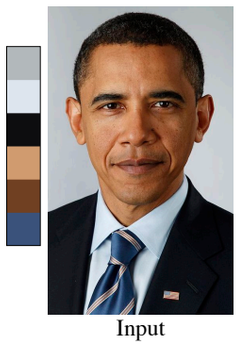


Input

Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

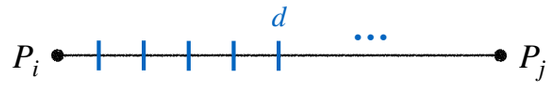


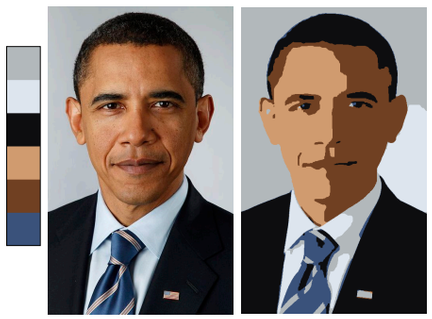


Input

Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

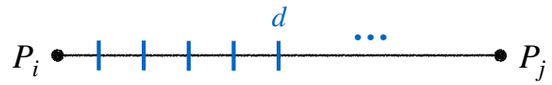


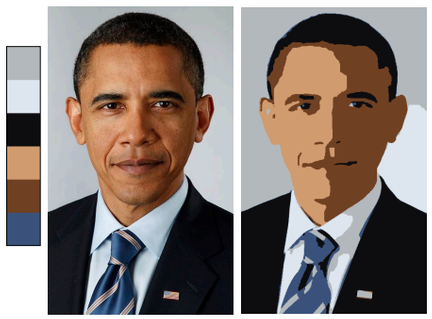


Input

Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

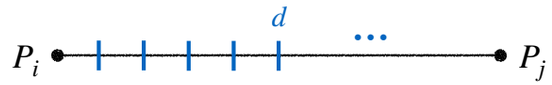


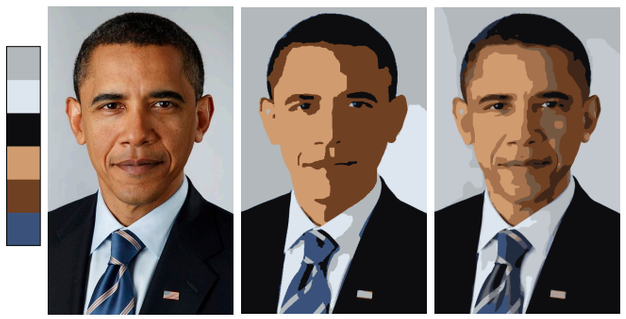


Input $d = 0$

Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

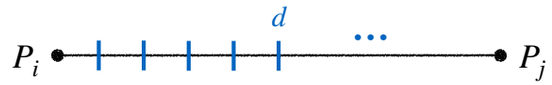


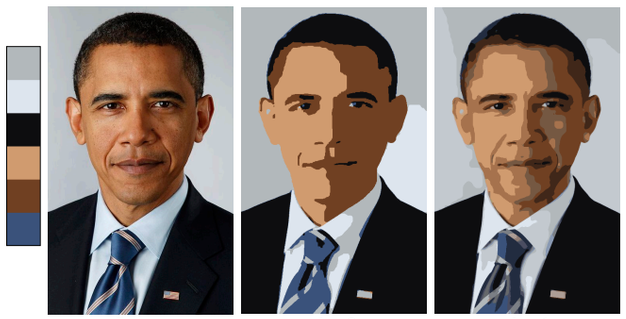


Input $d = 0$

Step 2: Rough region and color assignment

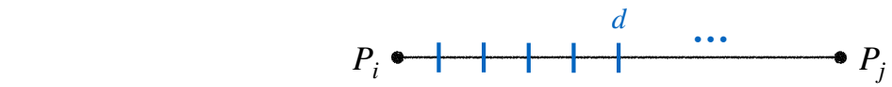
- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

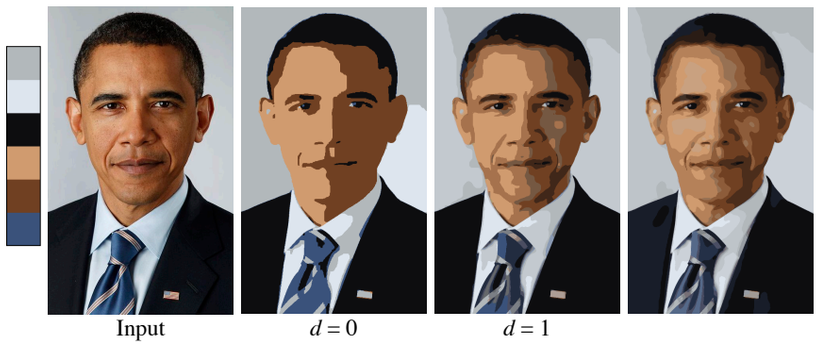




Step 2: Rough region and color assignment

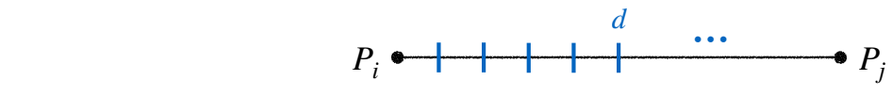
- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

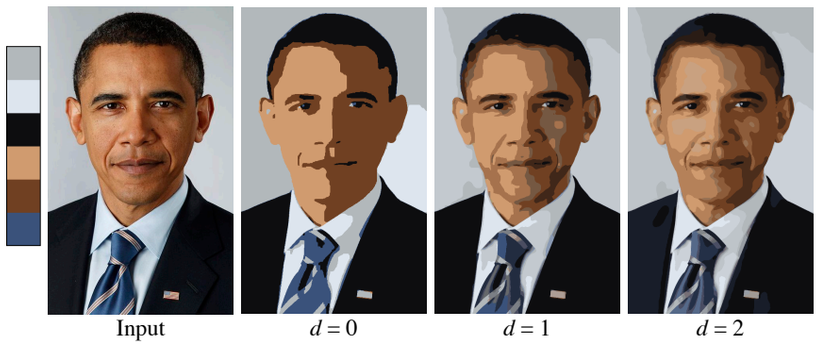




Step 2: Rough region and color assignment

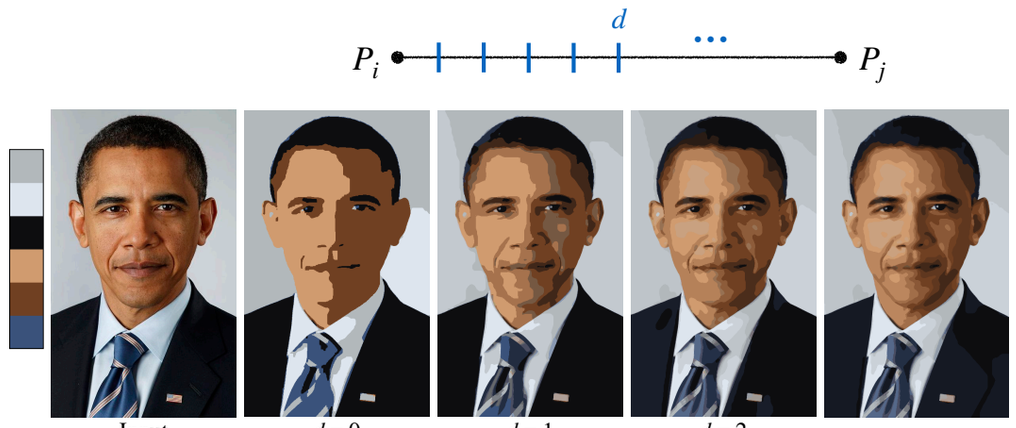
- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.





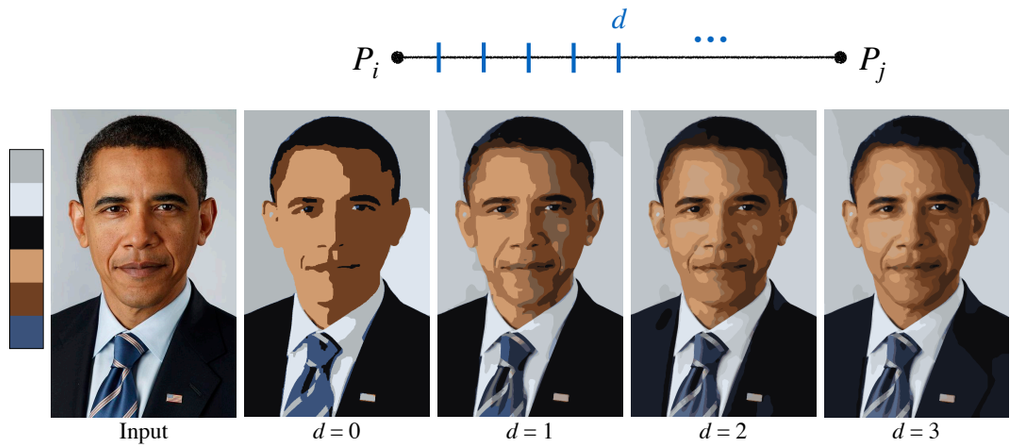
Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.



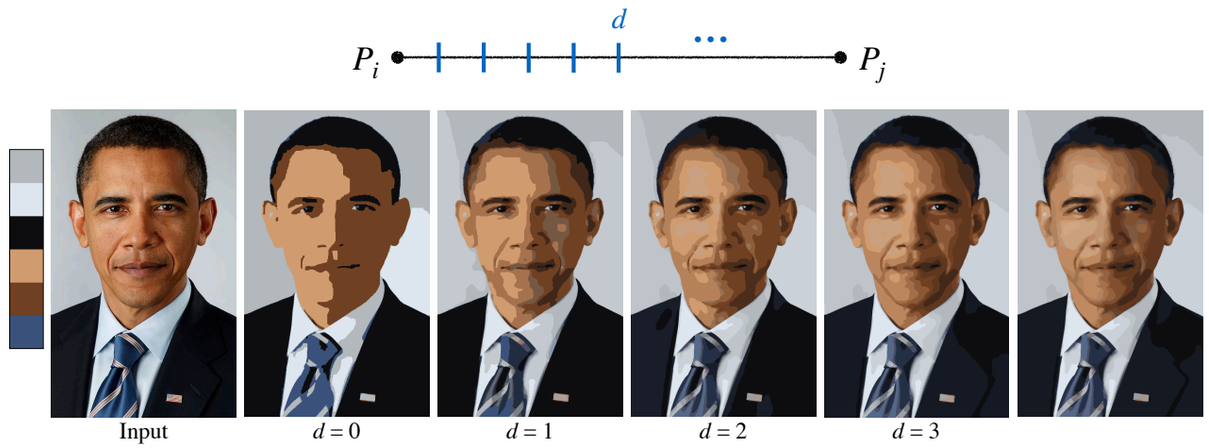
Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.



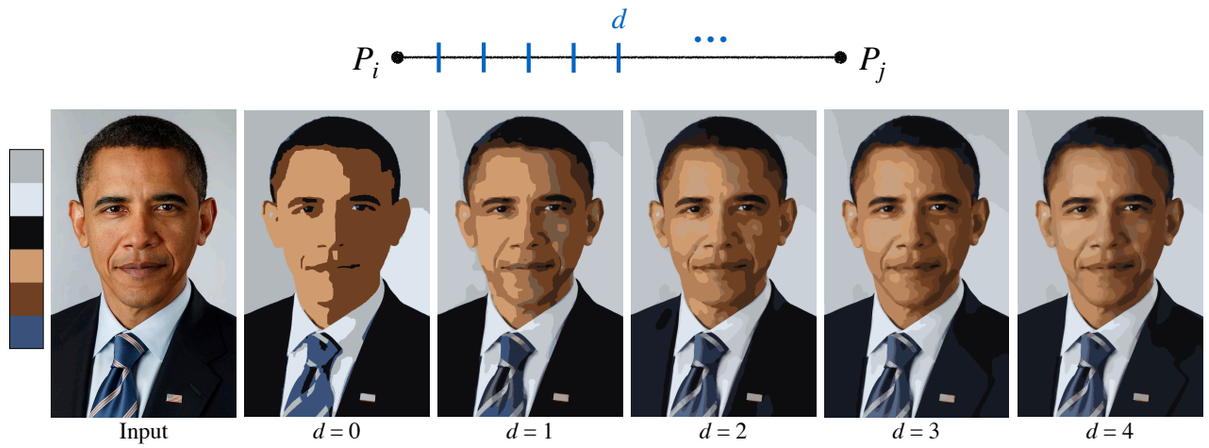
Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.



Step 2: Rough region and color assignment

- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.



Step 3&4: Region Refinement

- Step 3: Assign each region a continuous rather than discrete blend.
- Step 4: Smooth region boundaries with a frequency-guided median filter.
- See our paper for details.

Step 3&4: Region Refinement

- Step 3: Assign each region a continuous rather than discrete blend.
- Step 4: Smooth region boundaries with a frequency-guided median filter.
- See our paper for details.

Step 3&4: Region Refinement

- Step 3: Assign each region a continuous rather than discrete blend.
- Step 4: Smooth region boundaries with a frequency-guided median filter.
- See our paper for details.

Step 3&4: Region Refinement

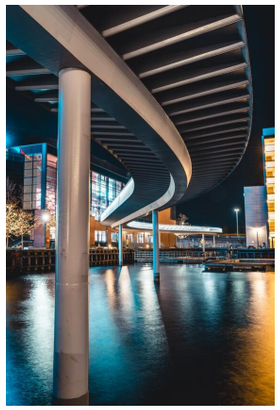
- Step 3: Assign each region a continuous rather than discrete blend.
- Step 4: Smooth region boundaries with a frequency-guided median filter.
- See our paper for details.

Step 3&4: Region Refinement

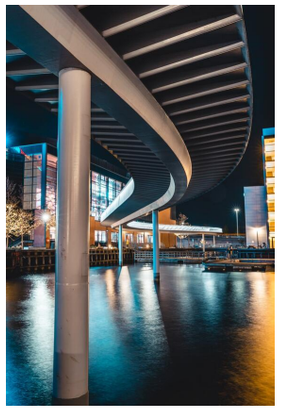
- Step 3: Assign each region a continuous rather than discrete blend.
- Step 4: Smooth region boundaries with a frequency-guided median filter.
- See our paper for details.

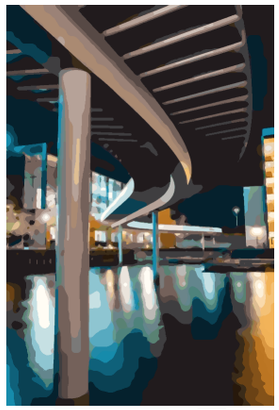
Results

Results

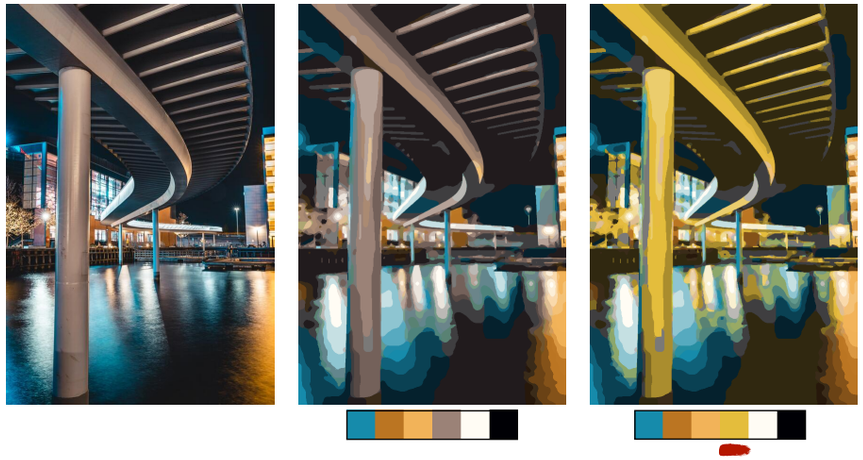


Results

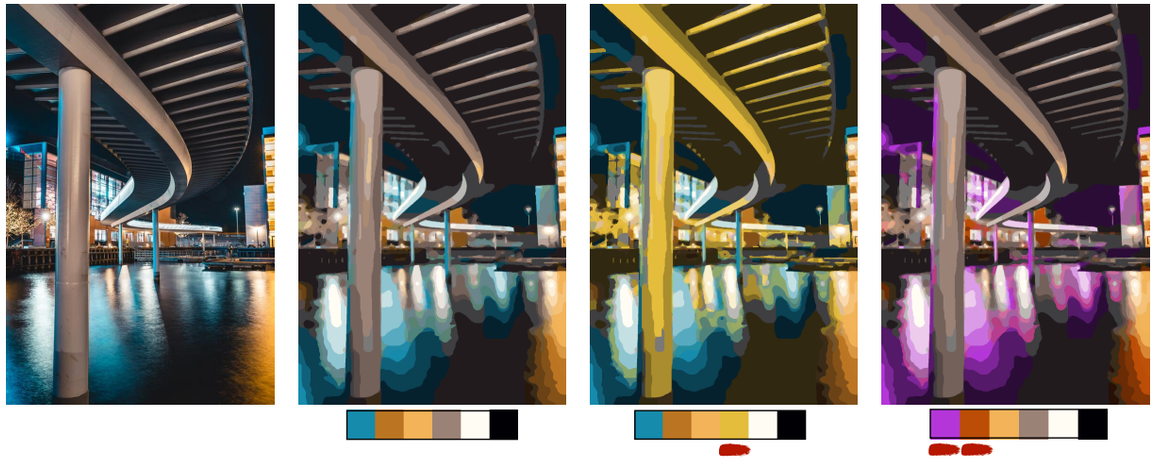


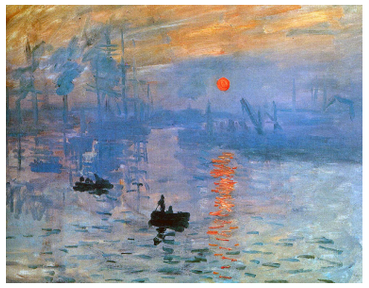


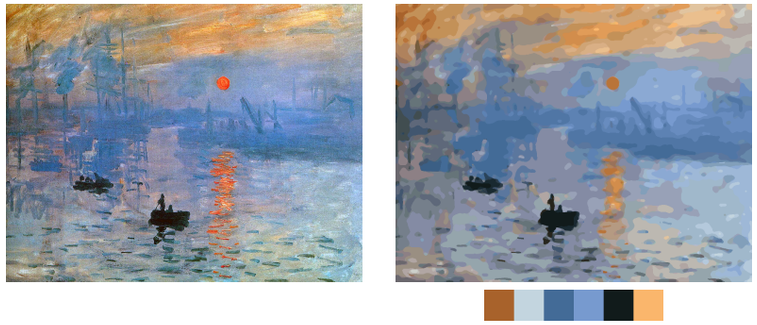
Results

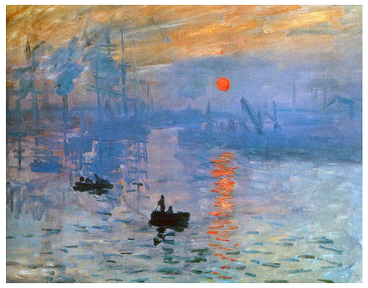


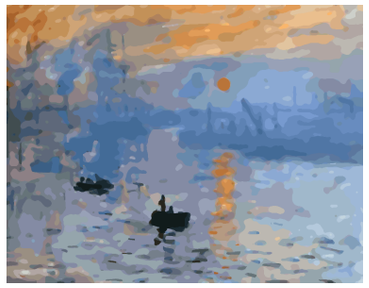
Results











Evaluation

Evaluation

- Comparison to related approaches.

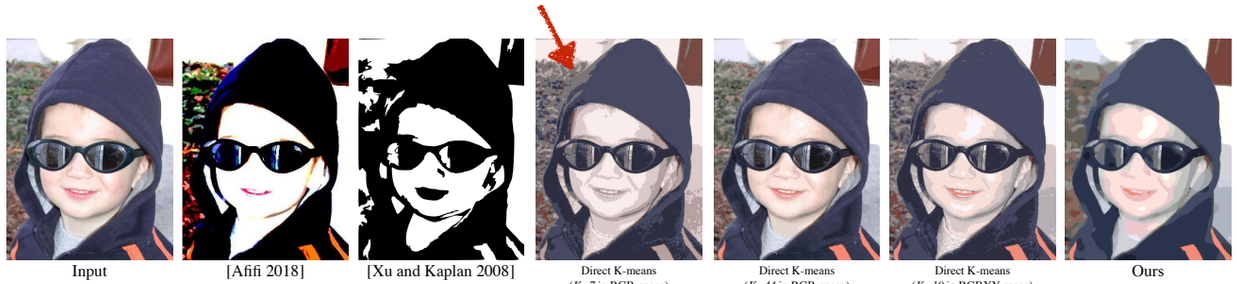
Evaluation

- Comparison to related approaches.

Input | [Afifi 2018] | [Xu and Kaplan 2008] | Direct K-means (*K=7* in RGB-space) | Direct K-means (*K=44* in RGB-space) | Direct K-means (*K=10* in RGBXY-space) | Ours

Evaluation

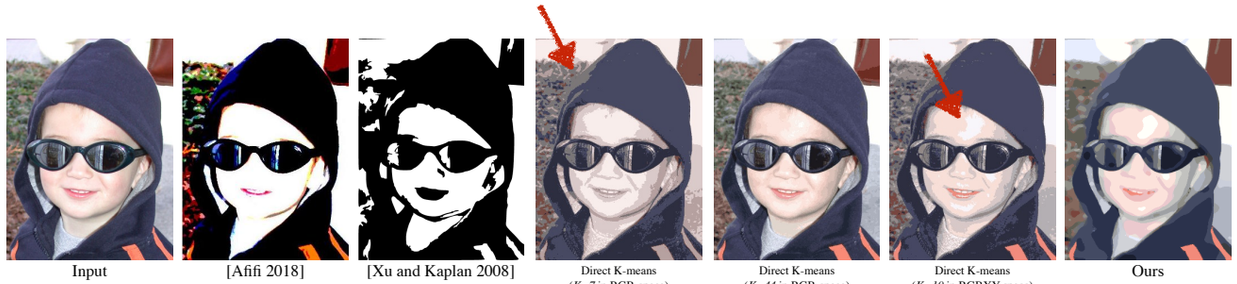
- Comparison to related approaches.



Input | [Afifi 2018] | [Xu and Kaplan 2008] | Direct K-means (*K=7* in RGB-space) | Direct K-means (*K=44* in RGB-space) | Direct K-means (*K=10* in RGBXY-space) | Ours

Evaluation

- Comparison to related approaches.



Input | [Afifi 2018] | [Xu and Kaplan 2008] | Direct K-means (*K=7* in RGB-space) | Direct K-means (*K=44* in RGB-space) | Direct K-means (*K=10* in RGBXY-space) | Ours

Evaluation

- Comparison to related approaches.

Input | [Afifi 2018] | [Xu and Kaplan 2008] | Direct K-means (*K=7* in RGB-space) | Direct K-means (*K=44* in RGB-space) | Direct K-means (*K=10* in RGBXY-space) | Ours

Evaluation

- Comparison to related approaches.

Evaluation

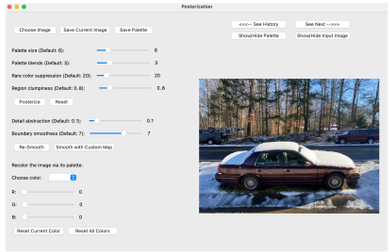
- Comparison to related approaches.

Evaluation

- Comparison to related approaches.
- Expert study with professional artists.

Evaluation

- Comparison to related approaches.
- Expert study with professional artists.

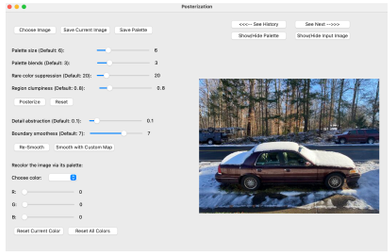


(44 colors)

Input | [Afifi 2018] | [Xu and Kaplan 2008] | Direct K-means (*K=7* in RGB-space) | Direct K-means (*K=44* in RGB-space) | Direct K-means (*K=10* in RGBXY-space) | Ours

Evaluation

- Comparison to related approaches.
- Expert study with professional artists.
- See paper for the details.



(44 colors)

Input | [Afifi 2018] | [Xu and Kaplan 2008] | Direct K-means (*K=7* in RGB-space) | Direct K-means (*K=44* in RGB-space) | Direct K-means (*K=10* in RGBXY-space) | Ours

Conclusion

Conclusion

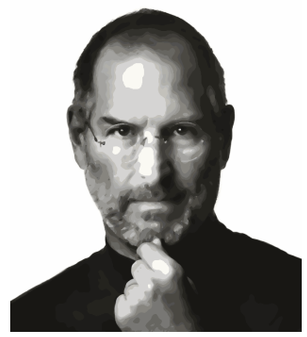
- PosterChild shows:

Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.

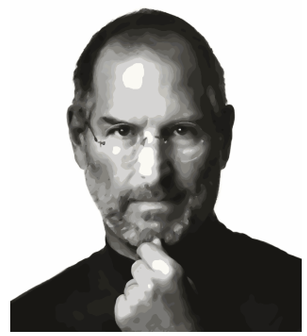
Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.



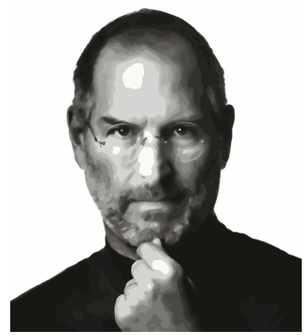
Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.
  - Easy to do palette-based recoloring on posters in real-time.



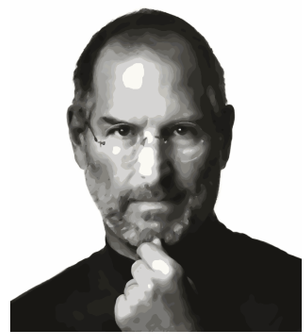
Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.
  - Easy to do palette-based recoloring on posters in real-time.
  - Aesthetically outperform state-of-the-art automatic posterization tools.



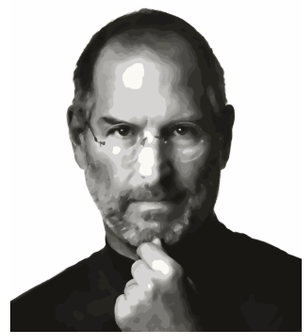
Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.
  - Easy to do palette-based recoloring on posters in real-time.
  - Aesthetically outperform state-of-the-art automatic posterization tools.
- Limitations:



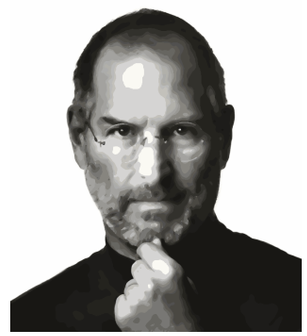
Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.
  - Easy to do palette-based recoloring on posters in real-time.
  - Aesthetically outperform state-of-the-art automatic posterization tools.
- Limitations:
  - Only allows real-time recoloring.



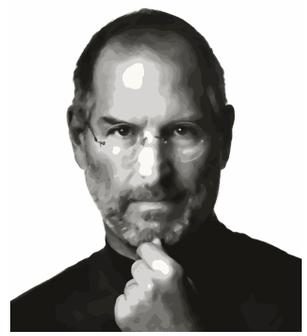
Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.
  - Easy to do palette-based recoloring on posters in real-time.
  - Aesthetically outperform state-of-the-art automatic posterization tools.
- Limitations:
  - Only allows real-time recoloring.
  - Slow performance on outlier removal.



Conclusion

- PosterChild shows:
  - Qualitatively similar to those created by artists in a time-consuming manner.
  - Easy to do palette-based recoloring on posters in real-time.
  - Aesthetically outperform state-of-the-art automatic posterization tools.
- Limitations:
  - Only allows real-time recoloring.
  - Slow performance on outlier removal.
  - Does not recognize the semantics of input images.



Thank You

- Code and GUI will be available at: https://cragl.cs.gmu.edu/
- Financial support
  - NSERC

GEORGE MASON UNIVERSITY CraGL Creativity and Graphics Lab

UNIVERSITY OF TORONTO

Thank You

- Code and GUI will be available at: https://cragl.cs.gmu.edu/
- Financial support
  - NSERC

GEORGE MASON UNIVERSITY CraGL Creativity and Graphics Lab

UNIVERSITY OF TORONTO

Step 1: Convex-hull based palette extraction

However, unfortunately, convex-hull based approach is sensitive to outliers. Its extracted palette is not compact and sometimes is not representative enough to human perceptual because of some outliers.

[Wang et al.] addressed this problem by formulating an optimization problem.

Our goal is not to address this sensitivity problem in convex-hull based approach. Our algorithm to extract palette is built on [Tan et al. 2016] and motivation from [Wang et al. 2019].

We found a simple way which could remove the outliers for simplified convex hull: applying K-means clustering on the input RGB colors before performing [Tan et al. 2016]’s convex hull simplification.

Step 1: Convex-hull based palette extraction

- [Wang et al. 2019] observed that convex-hull based palettes are sensitive to outliers.

Step 1: Convex-hull based palette extraction

- [Wang et al. 2019] observed that convex-hull based palettes are sensitive to outliers.
- K-means as a relaxation on the input RGB colors.

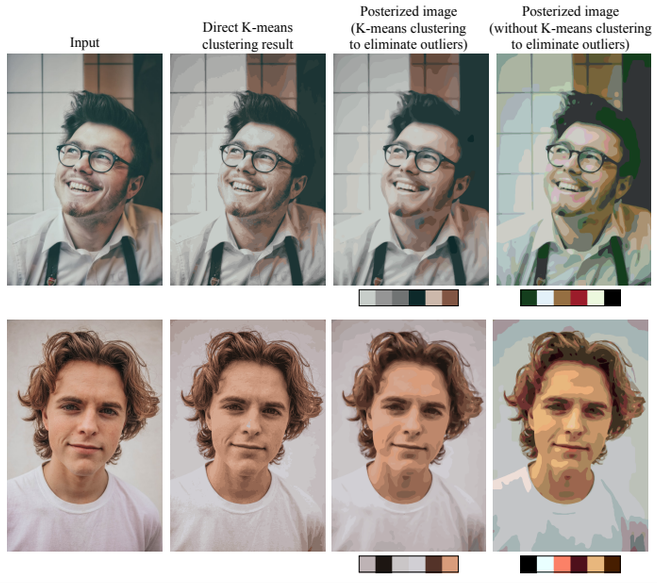
Step 1: Convex-hull based palette extraction

Step 1: Convex-hull based palette extraction

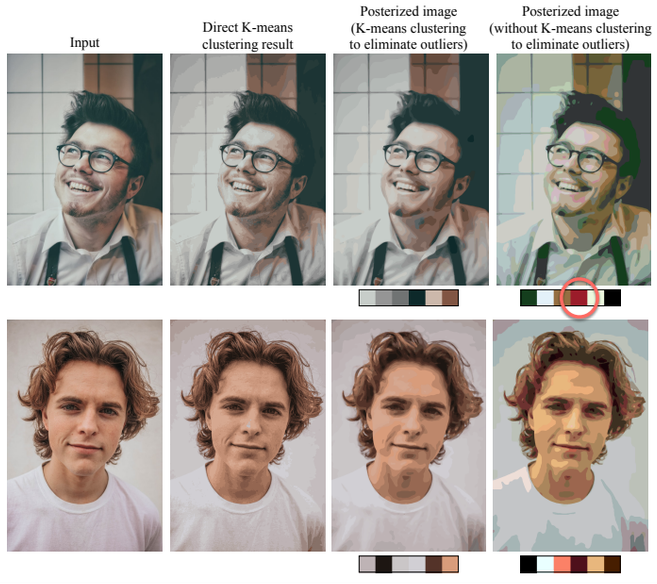
If you look at the example in the first row, you would not see a dark-red-like color appearing in the input, but dark red could be extracted by the [Tan et al.] convex hull simplification without using K-means as a relaxation.

Similar to the example in the second row, orange should not be appeared in the palette. For human perception, what colors we could see from this input are just skin color, brown, black and white, (gray which is mix of black and white).

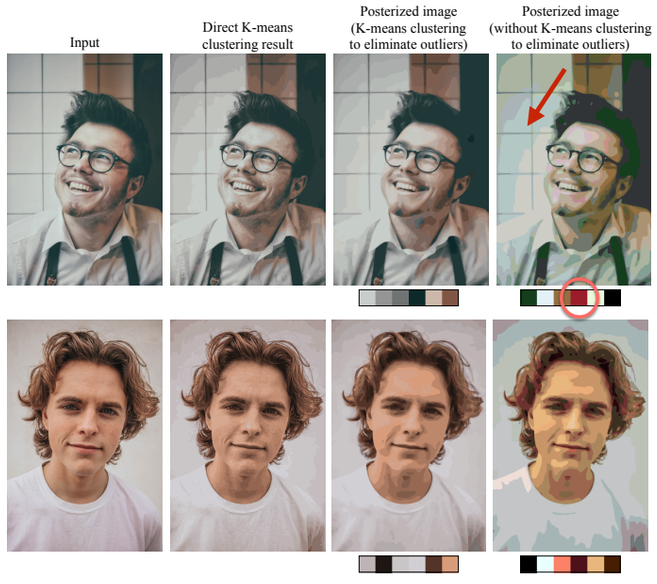
Step 1: Convex-hull based palette extraction



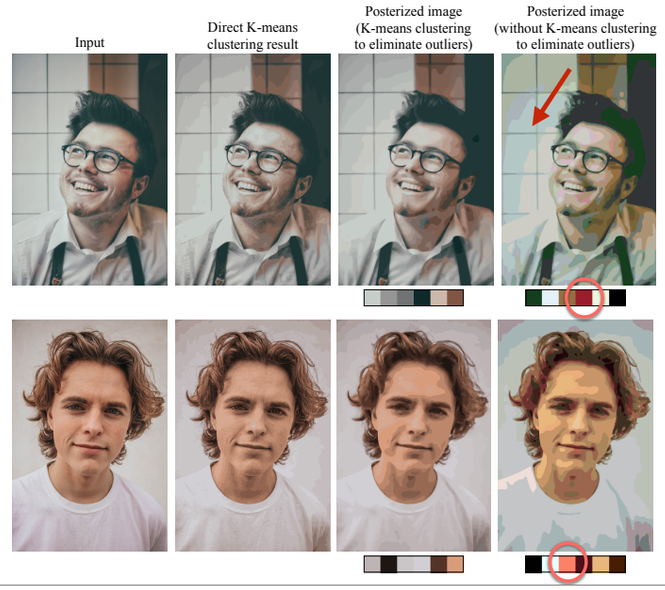
Step 1: Convex-hull based palette extraction



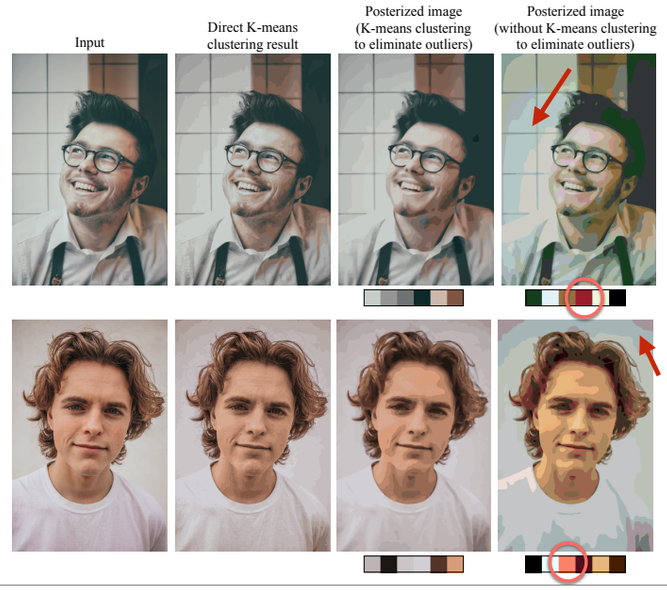
Step 1: Convex-hull based palette extraction



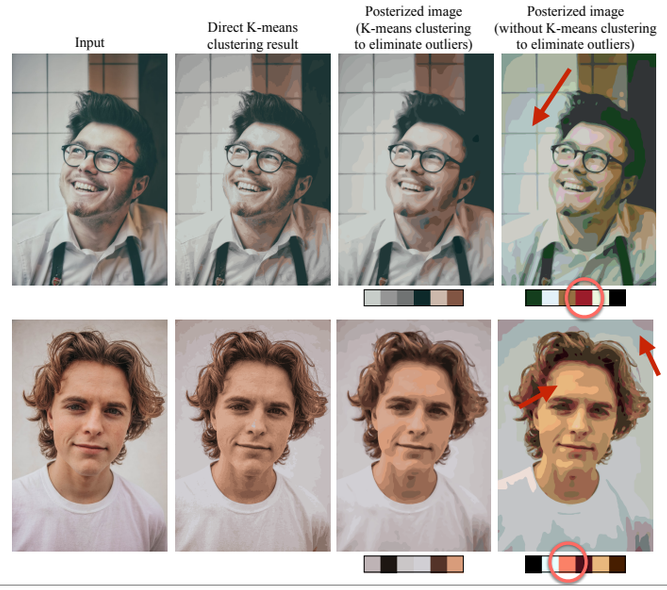
Step 1: Convex-hull based palette extraction



Step 1: Convex-hull based palette extraction

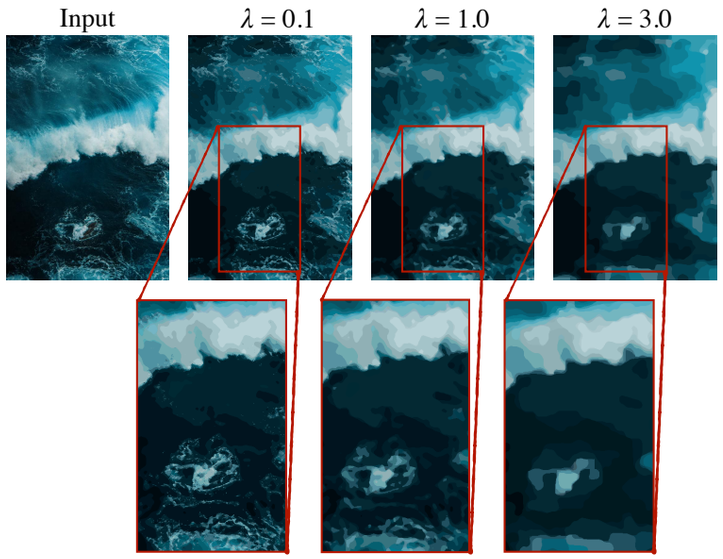


Step 1: Convex-hull based palette extraction

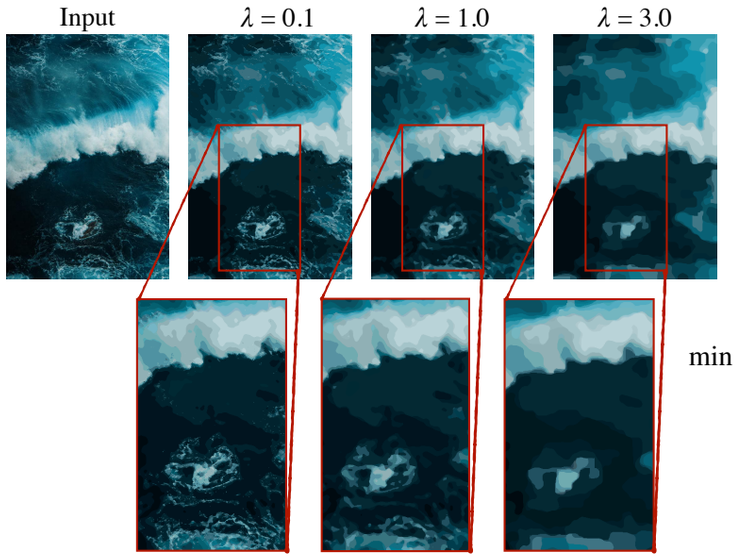


Step 2: Rough region and color assignment

Step 2: Rough region and color assignment

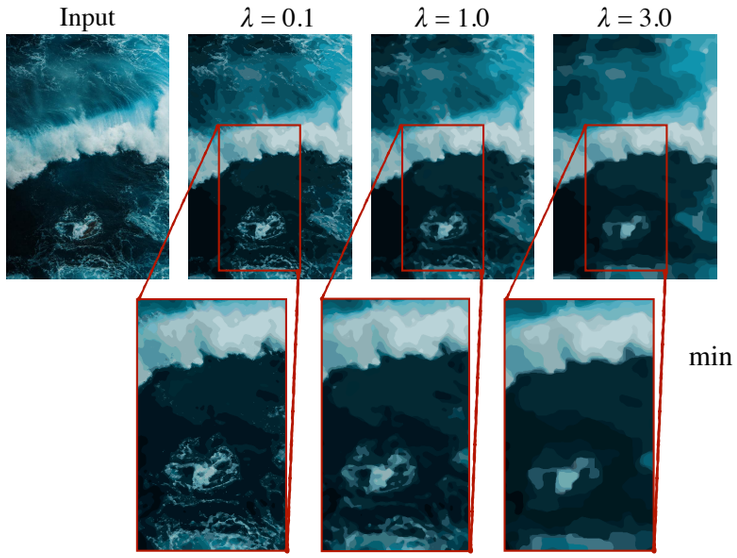


Step 2: Rough region and color assignment



$$\min \sum_{p \in I} ||f_p - I_p||_2 + \lambda \sum_{p,q \in N} ||L_p - L_q||_2$$

Step 2: Rough region and color assignment



$$\min \sum_{p \in I} ||f_p - I_p||_2 + \lambda \sum_{p,q \in N} ||L_p - L_q||_2$$