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

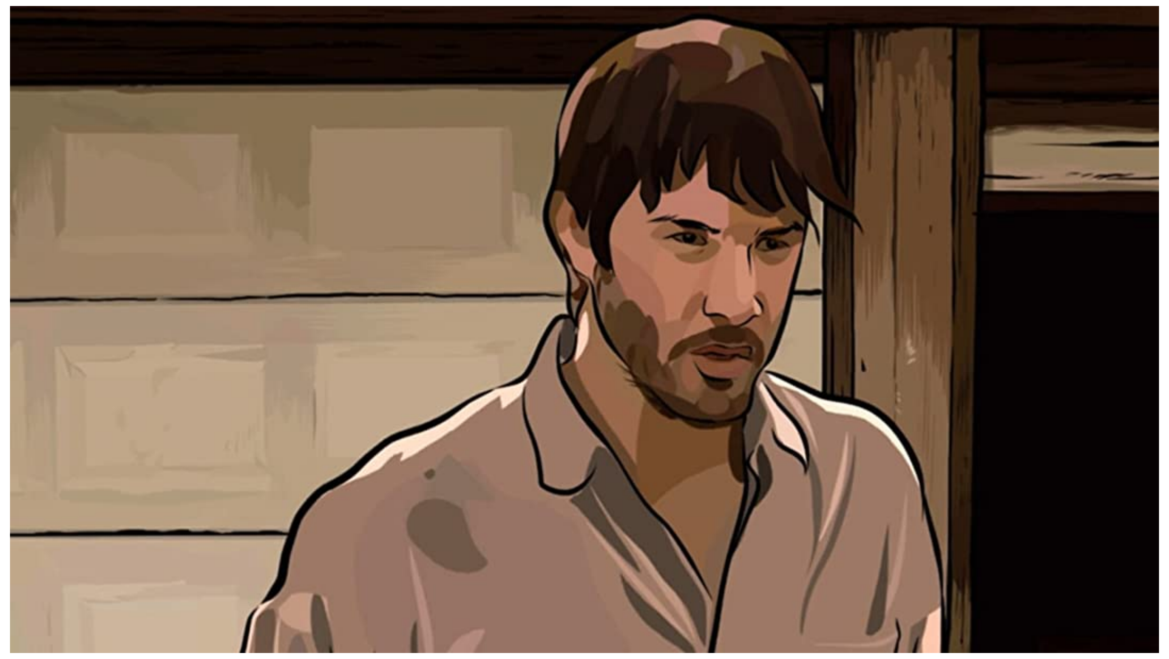
University of Toronto

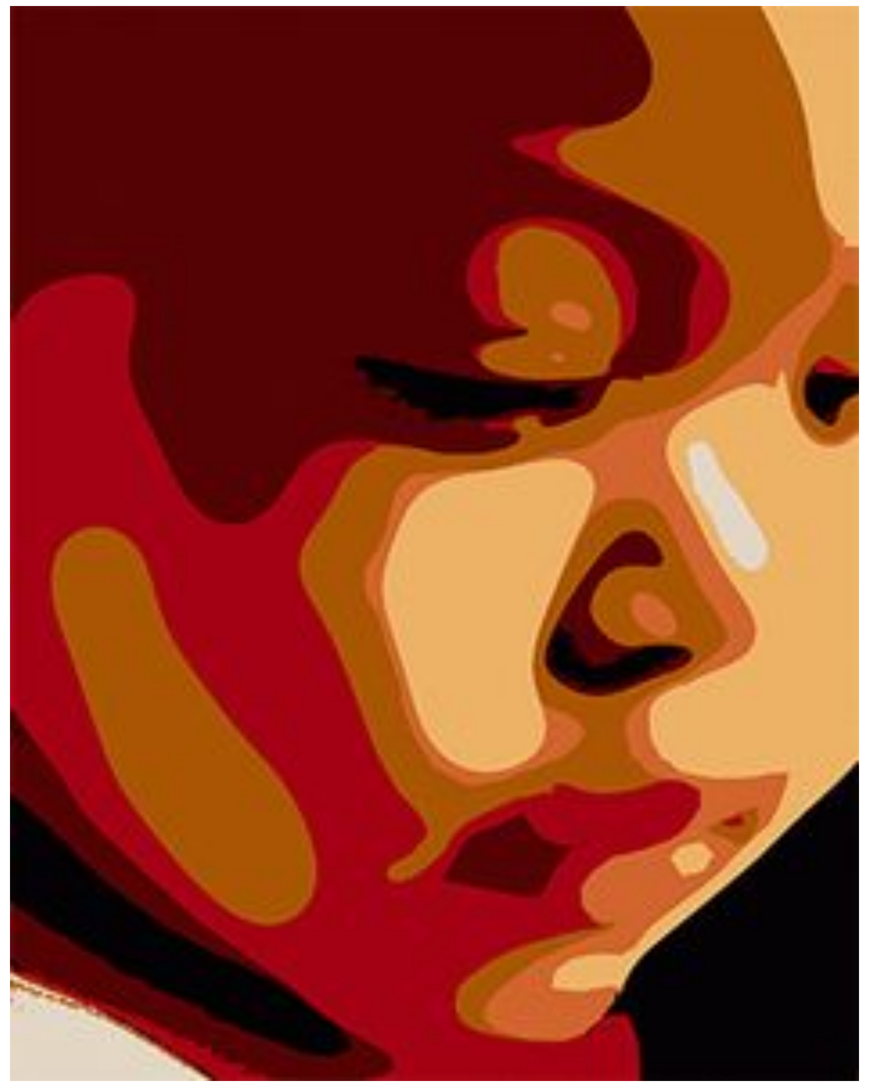
Artistic Posterization

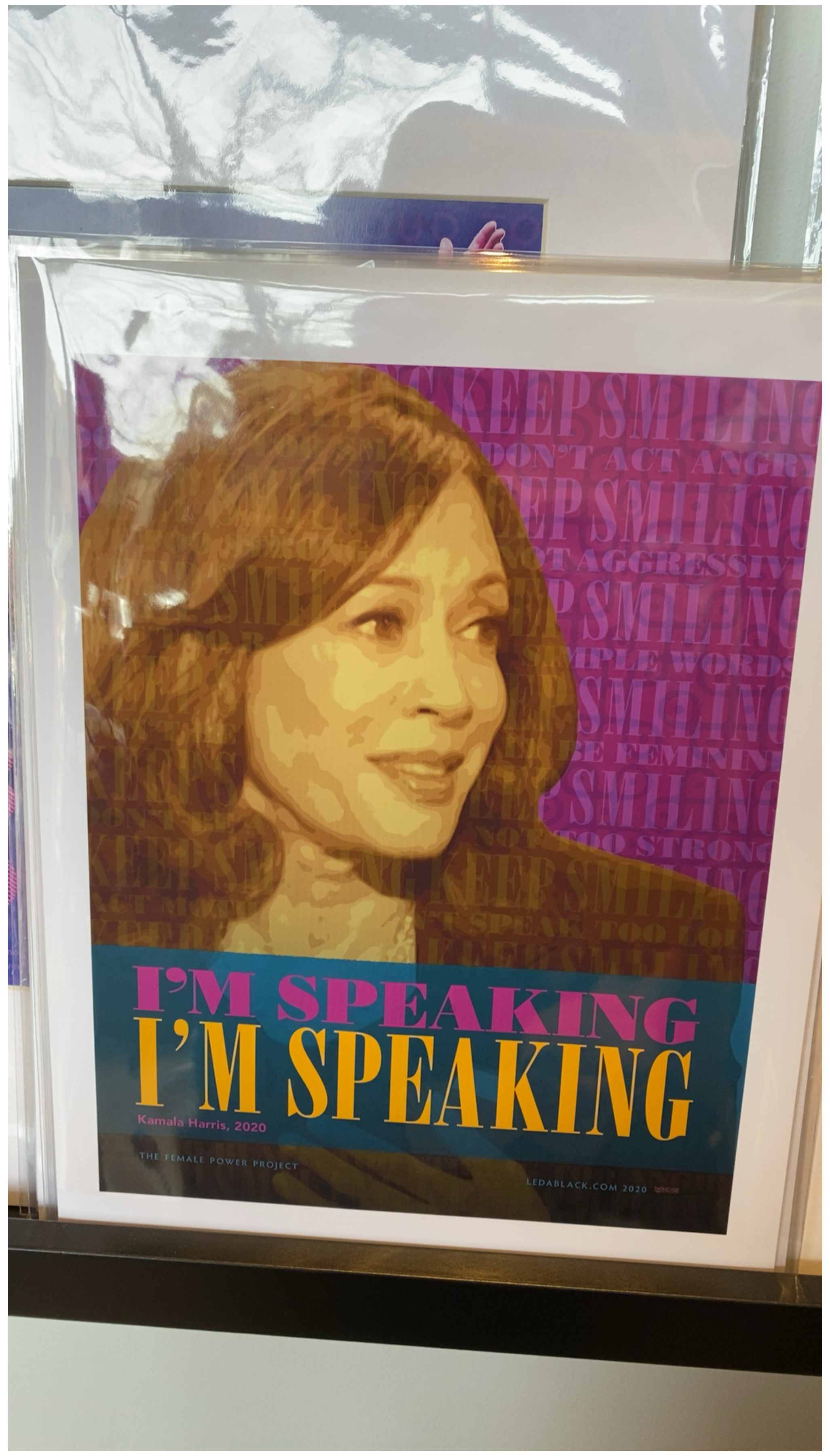
Artistic Posterization

Artistic Posterization

Artistic Posterization







Artistic Posterization

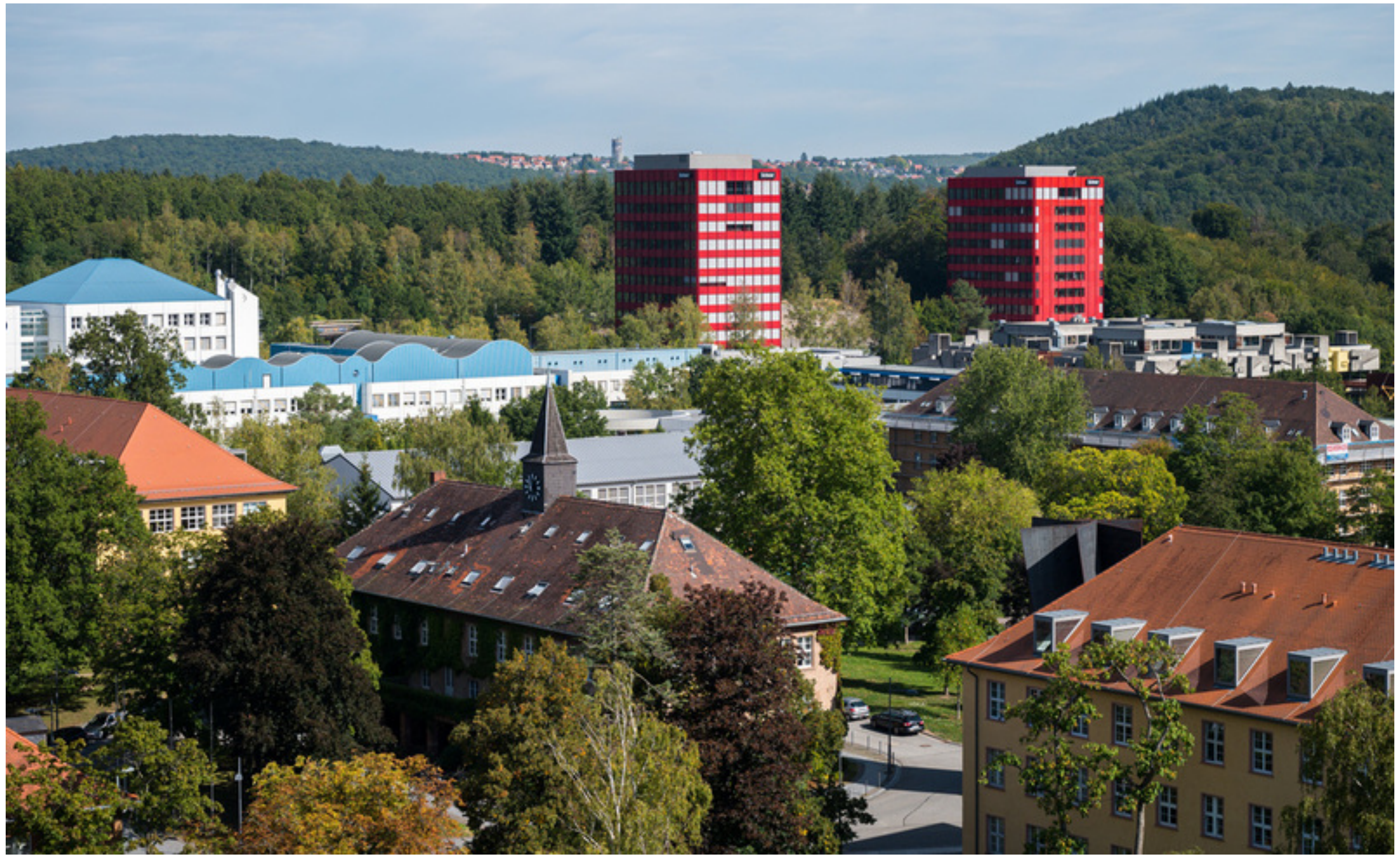
KEEP SMILING
DON'T ACT ANGRY
I'M SPEAKING
I'M SPEAKING
Kamala Harris, 2020
THE FEMALE POWER PROJECT
LEDABLACK.COM 2020

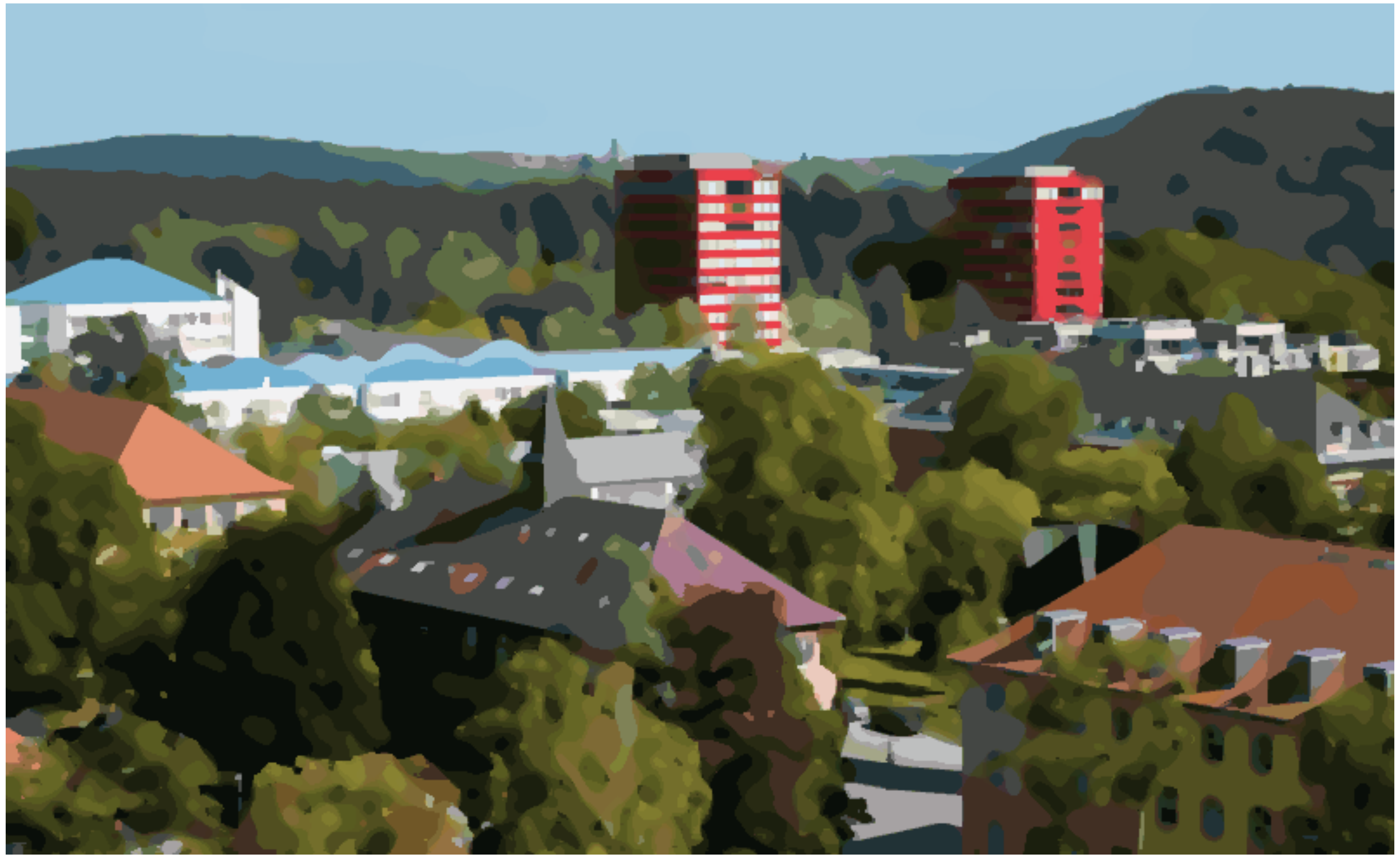
Saarland University

Saarland University

Saarland University

Saarland University





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

Problem Statement

Problem Statement

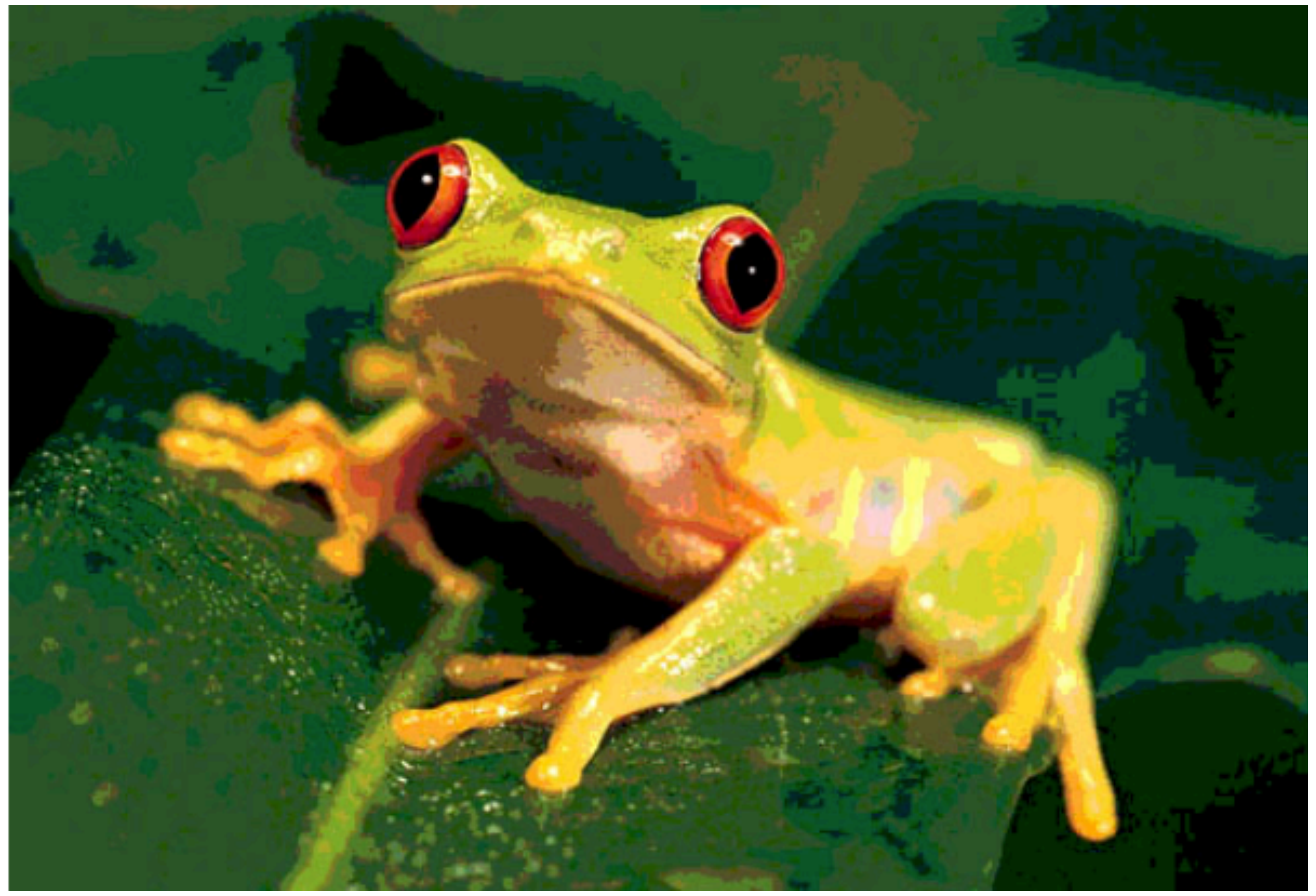
- Create a posterized image using a discrete set of colors representing the input image and also provided convenient handles for recoloring.
  - Existing automatic posterization tools produce output quite different from artists.
  - Manual approaches are time-consuming.

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

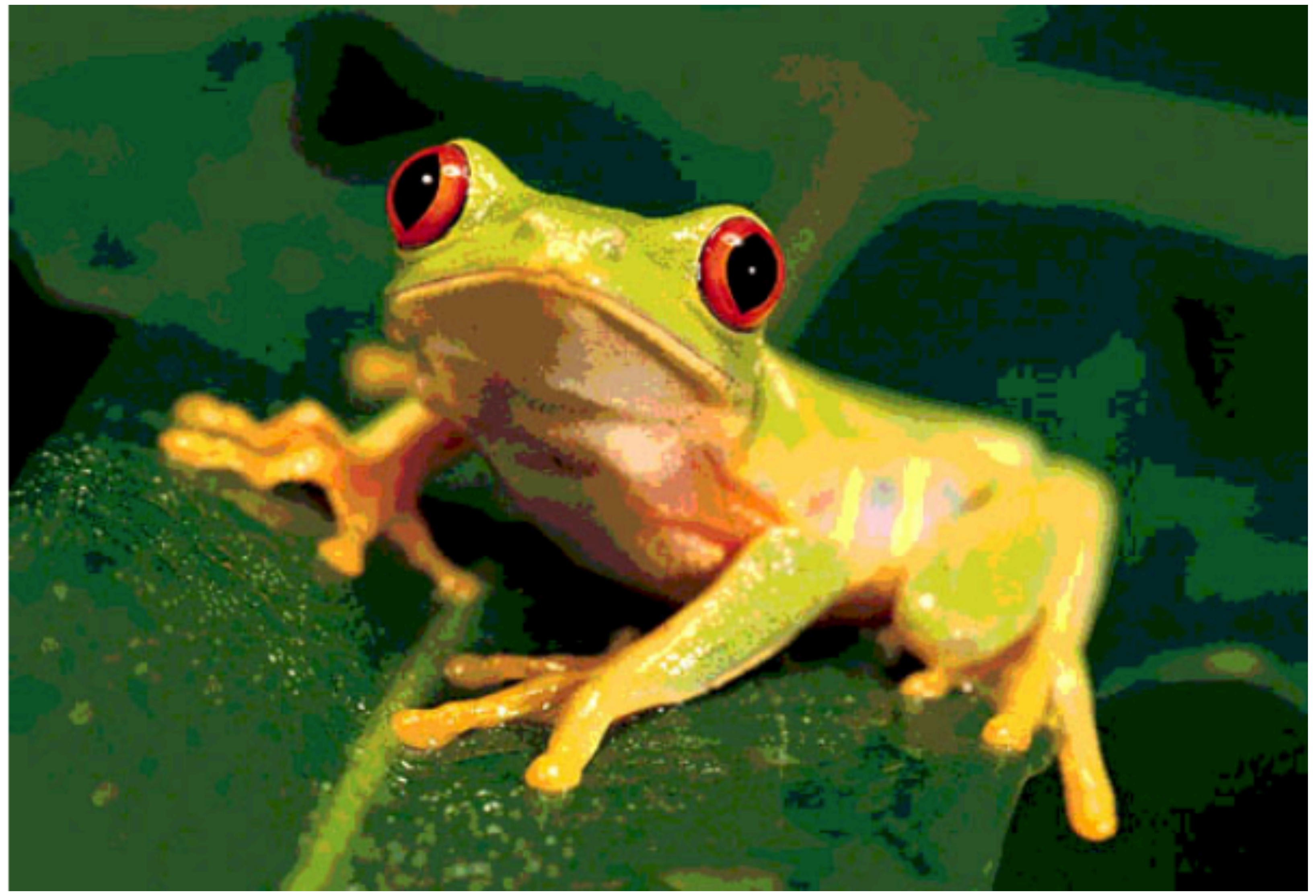
- Create a posterized image using a discrete set of colors representing the input image and also provided convenient handles for recoloring.
  - Existing automatic posterization tools produce output quite different from artists.
  - Manual approaches are time-consuming.



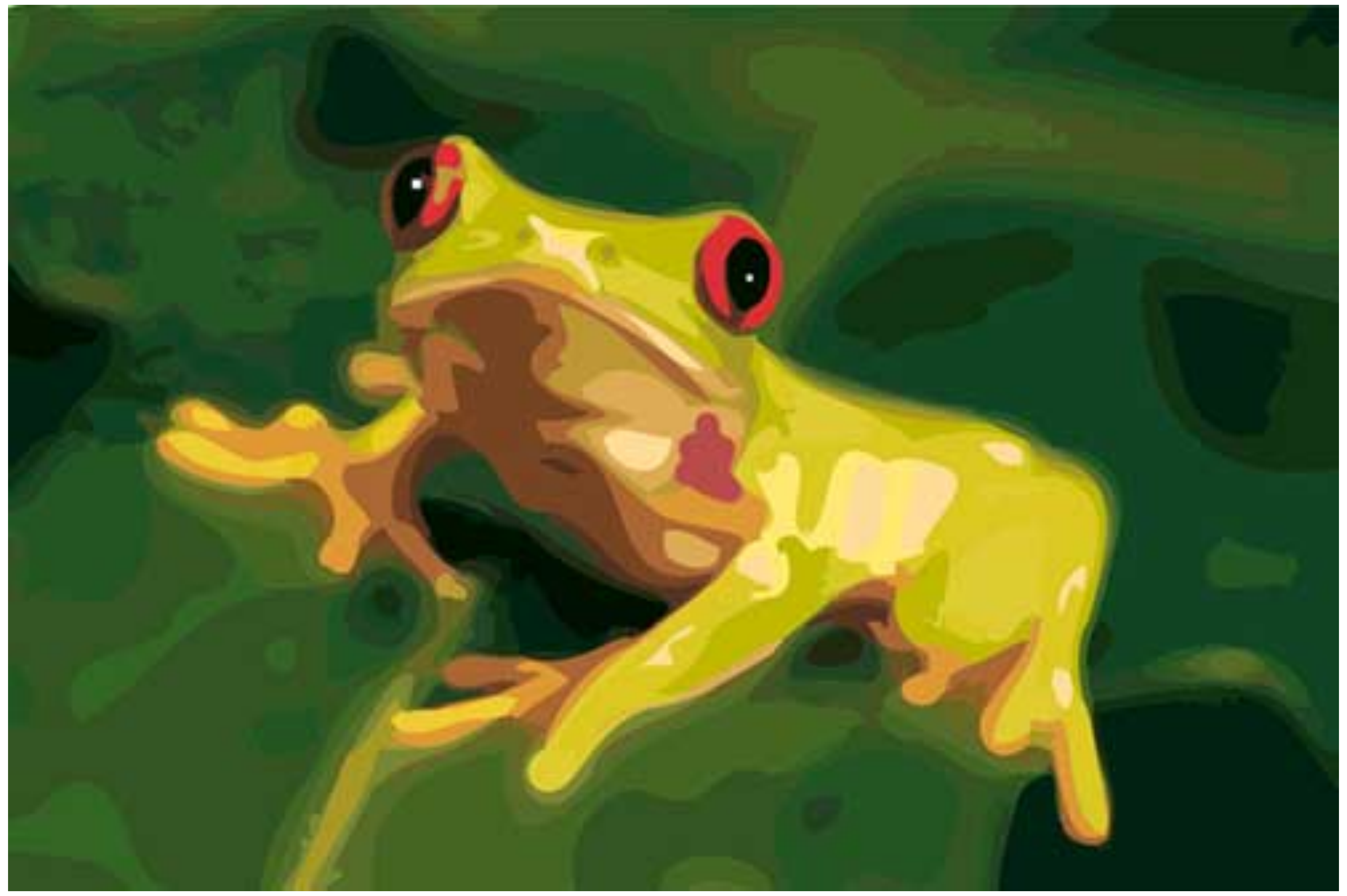
Photoshop's Posterization Filter

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.

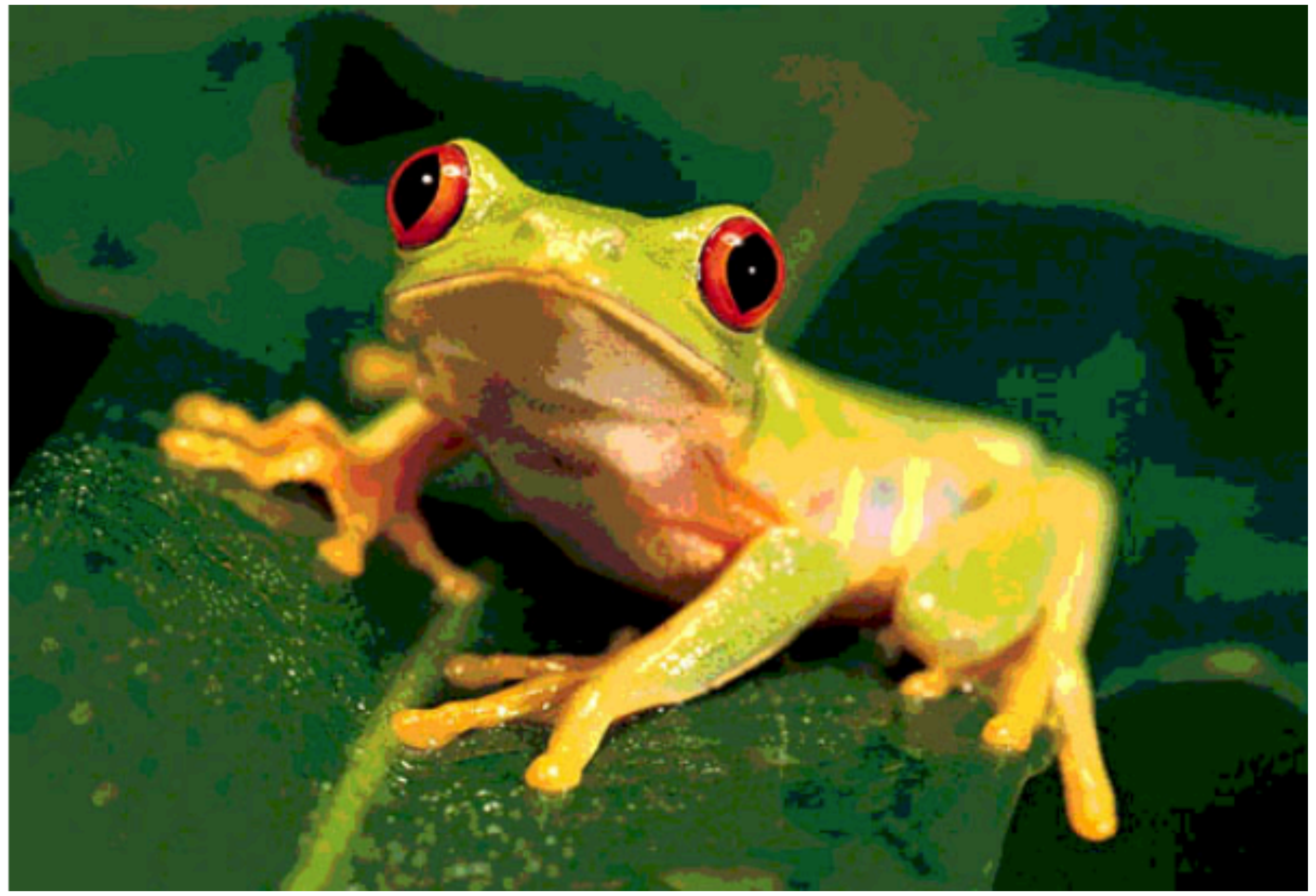


Photoshop's Posterization Filter

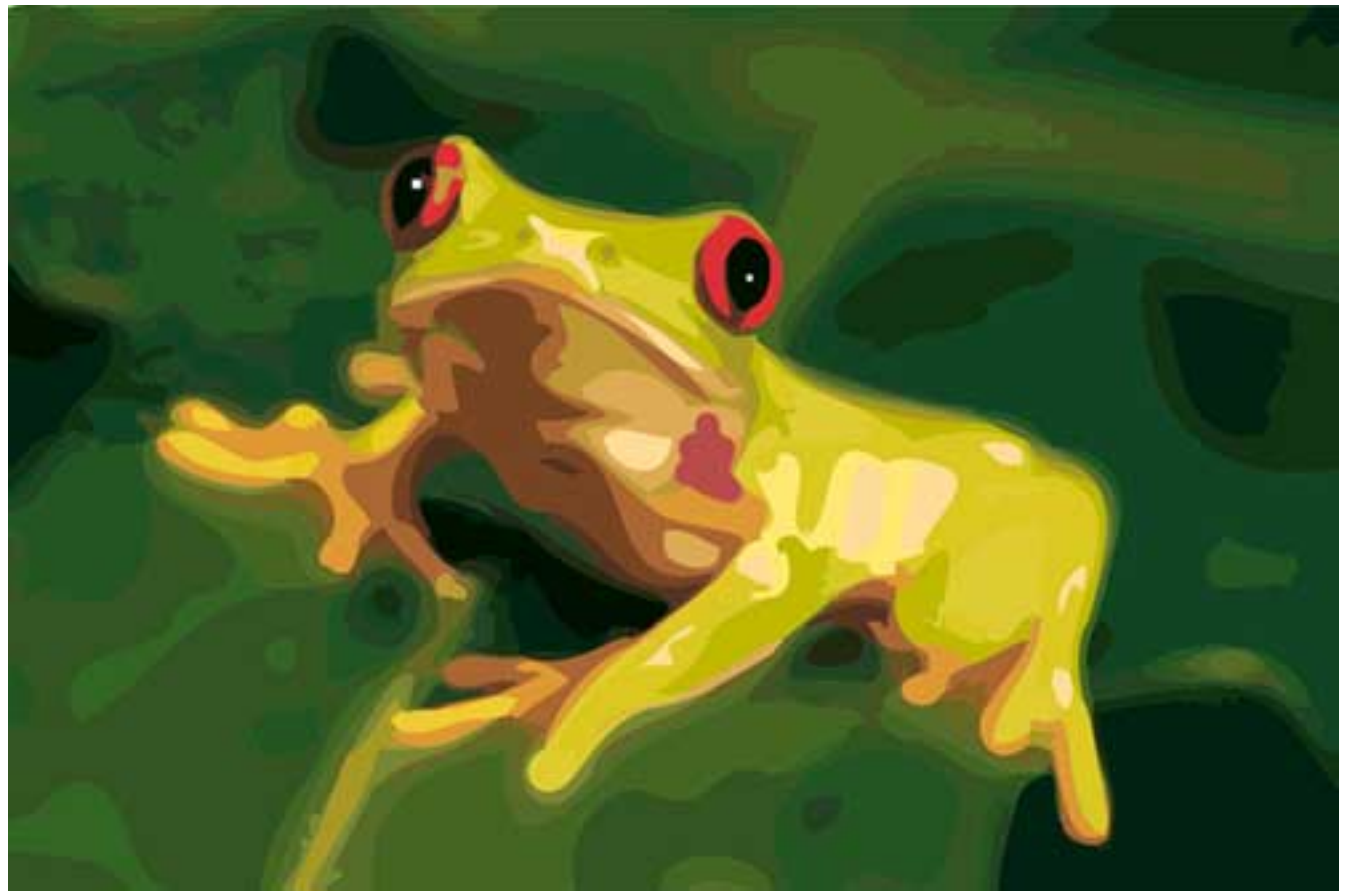


Problem Statement

- 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 Approach

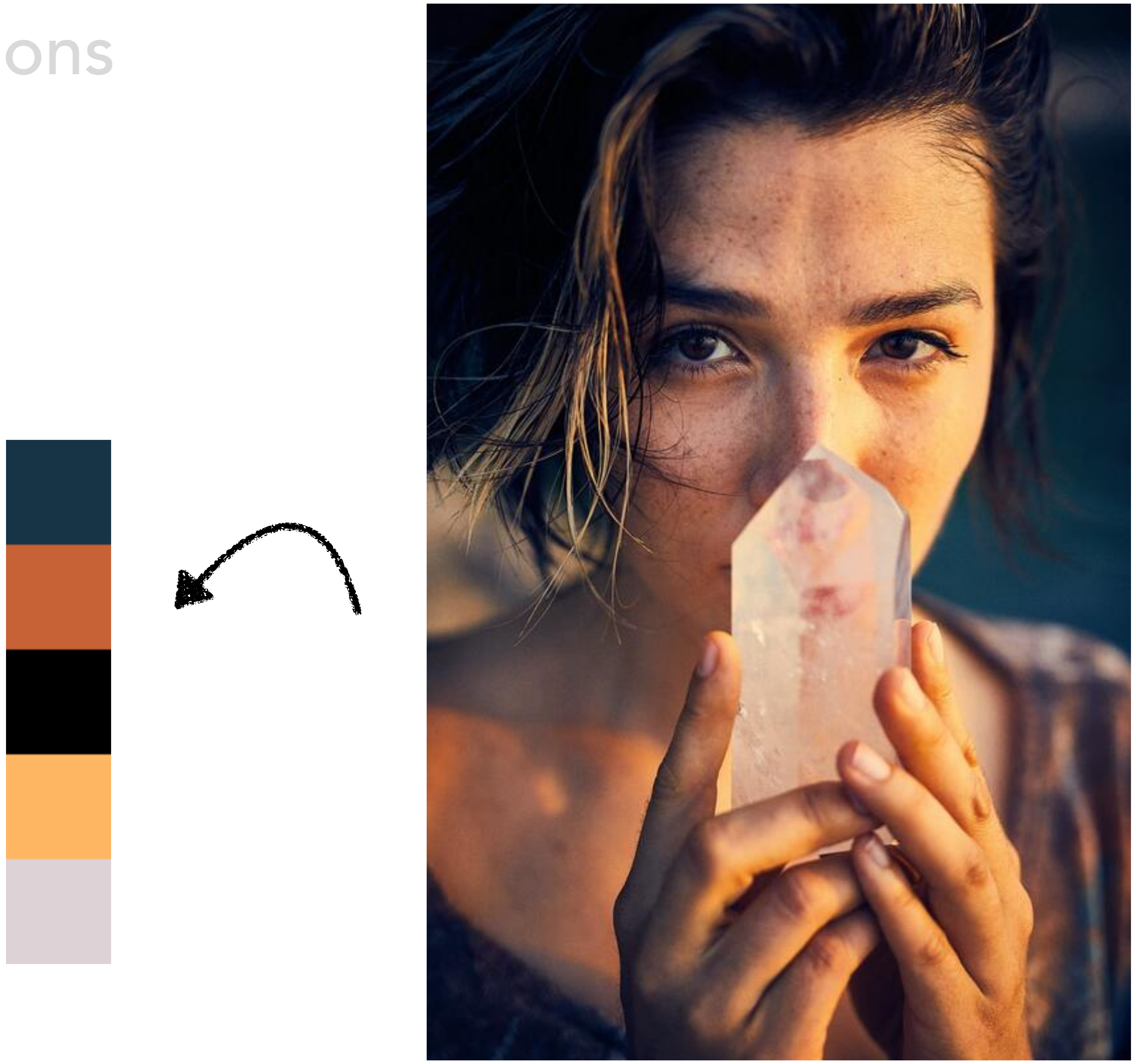
- Step 1: Convex-hull based palette extraction
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

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: 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

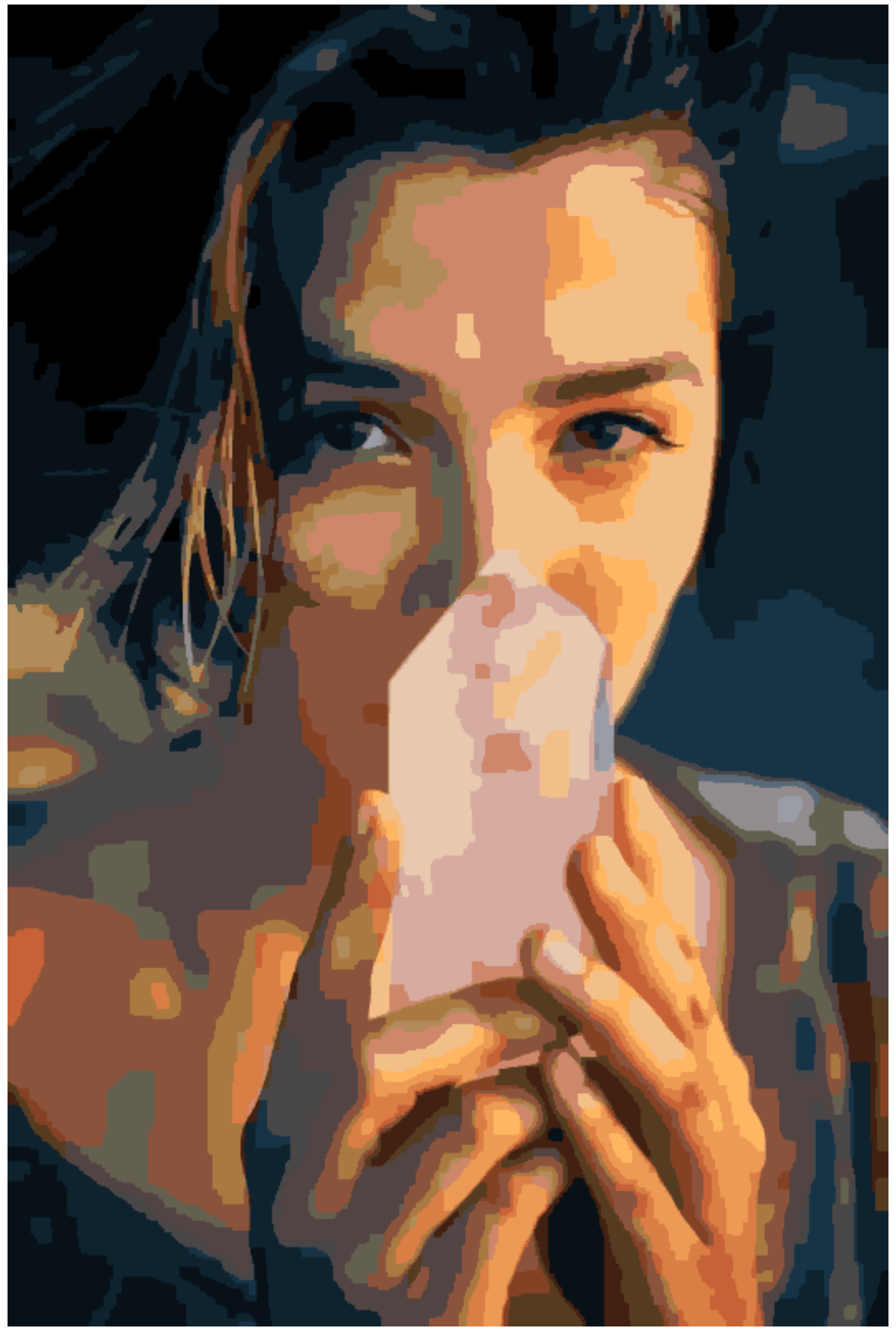
- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

Our 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

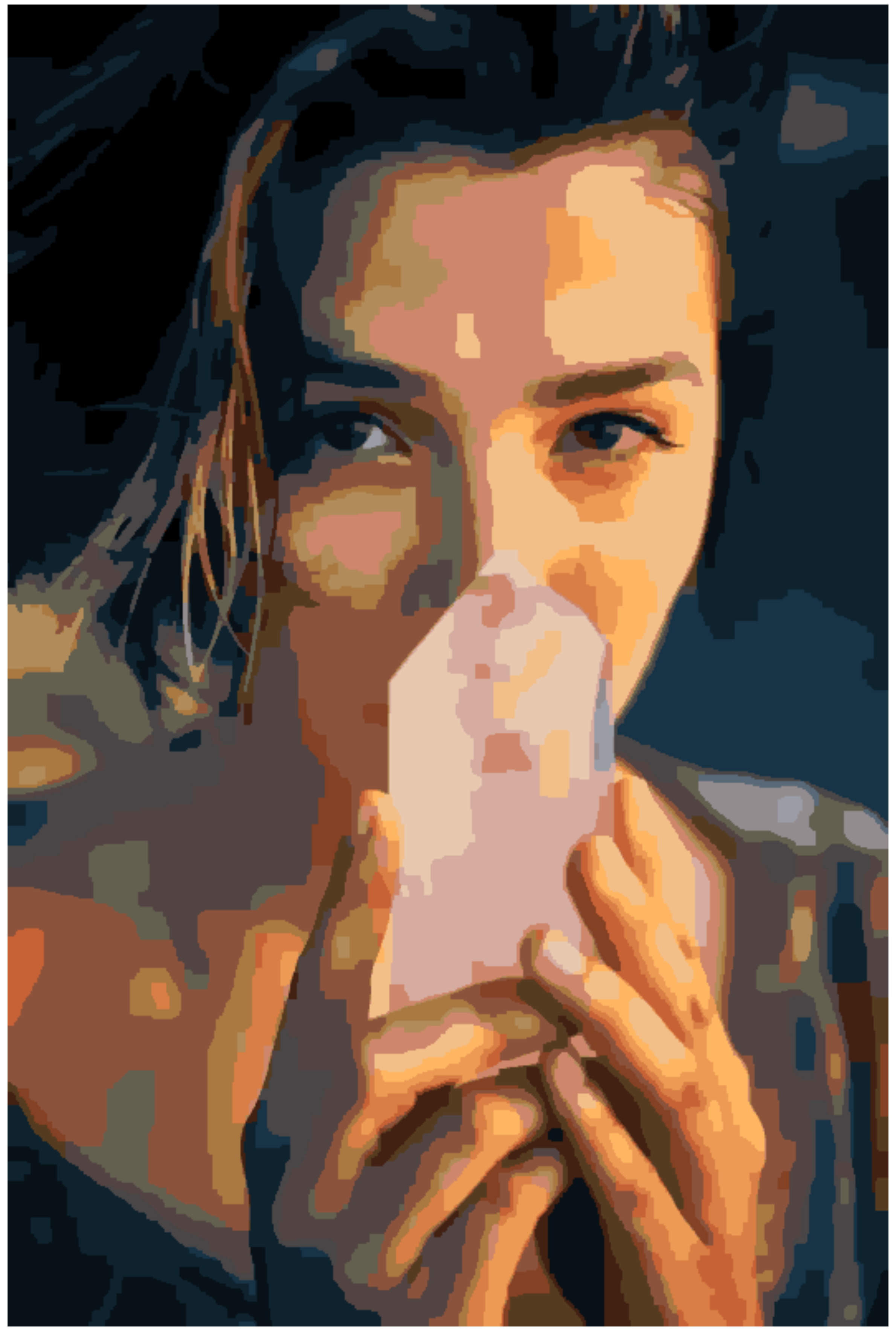


Our Approach

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

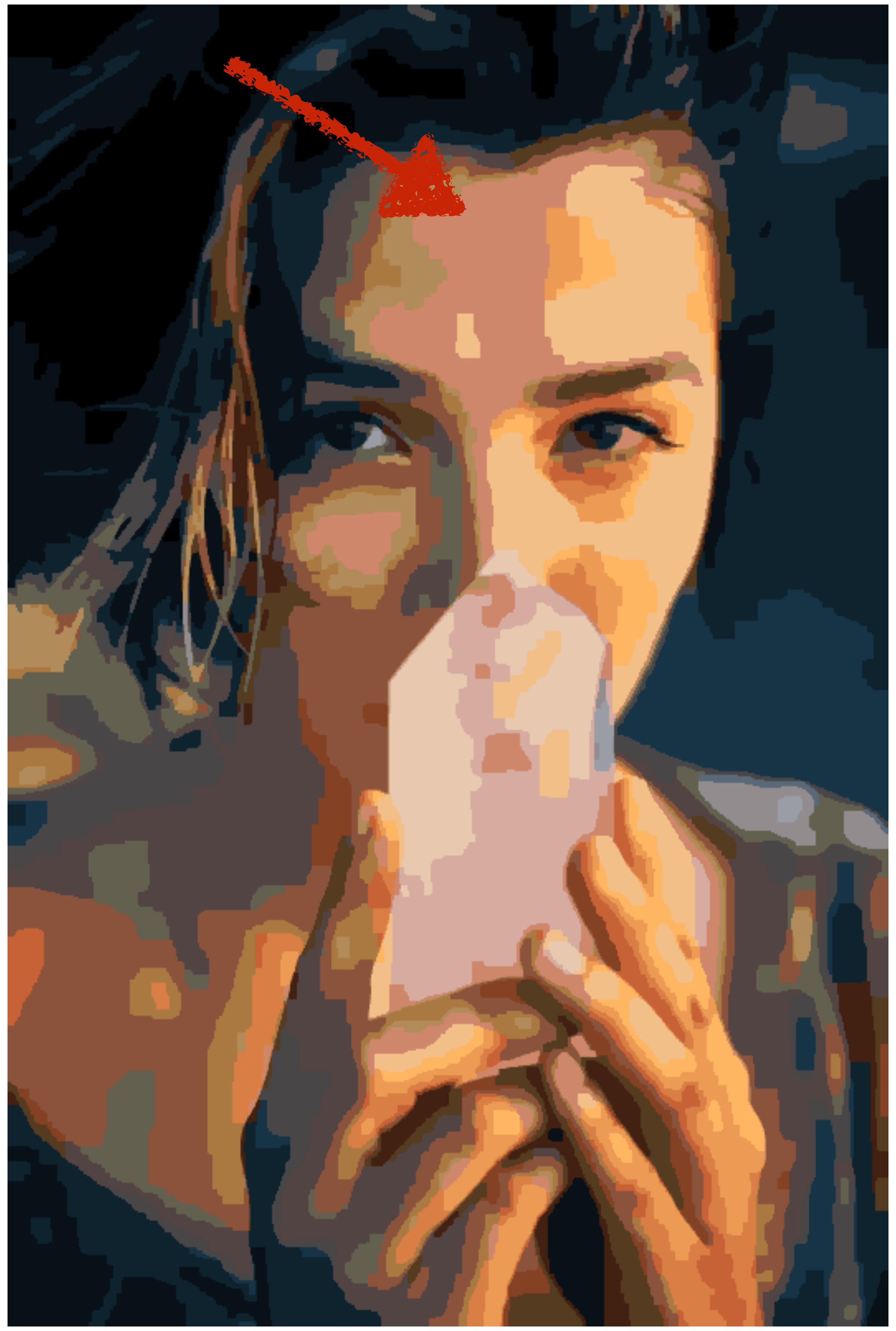
Our Approach

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



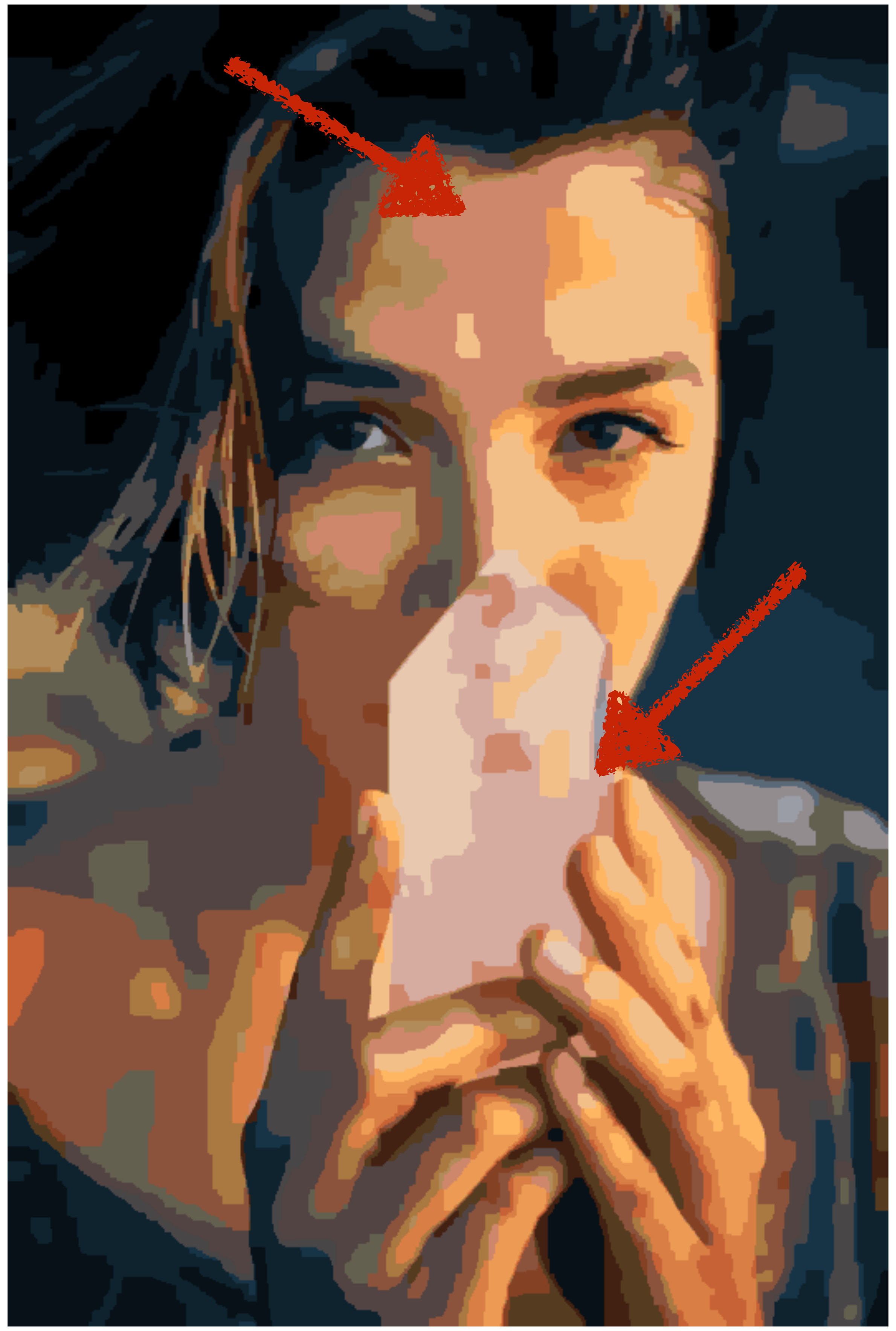
Our Approach

- 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

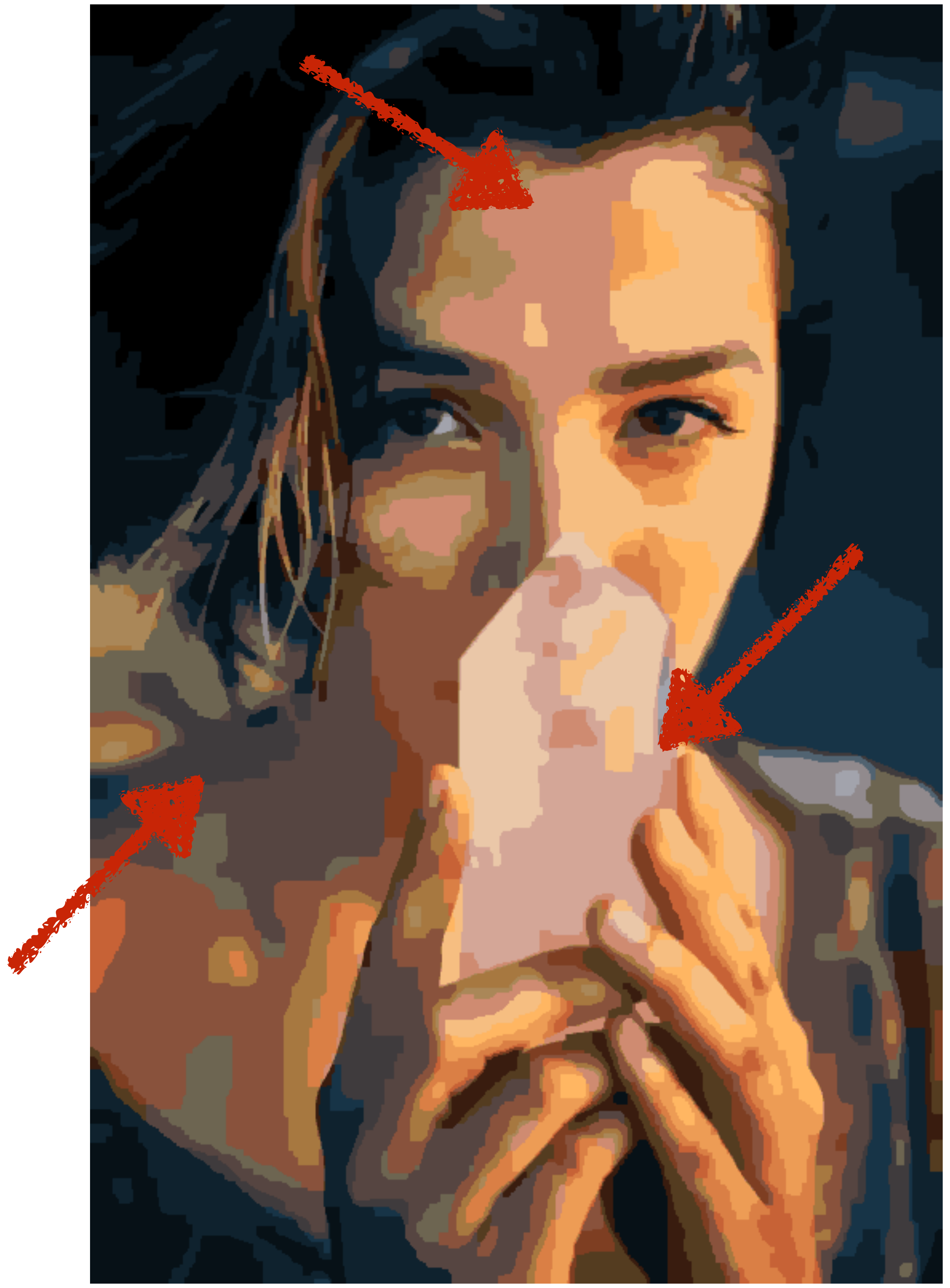


Our Approach

- 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

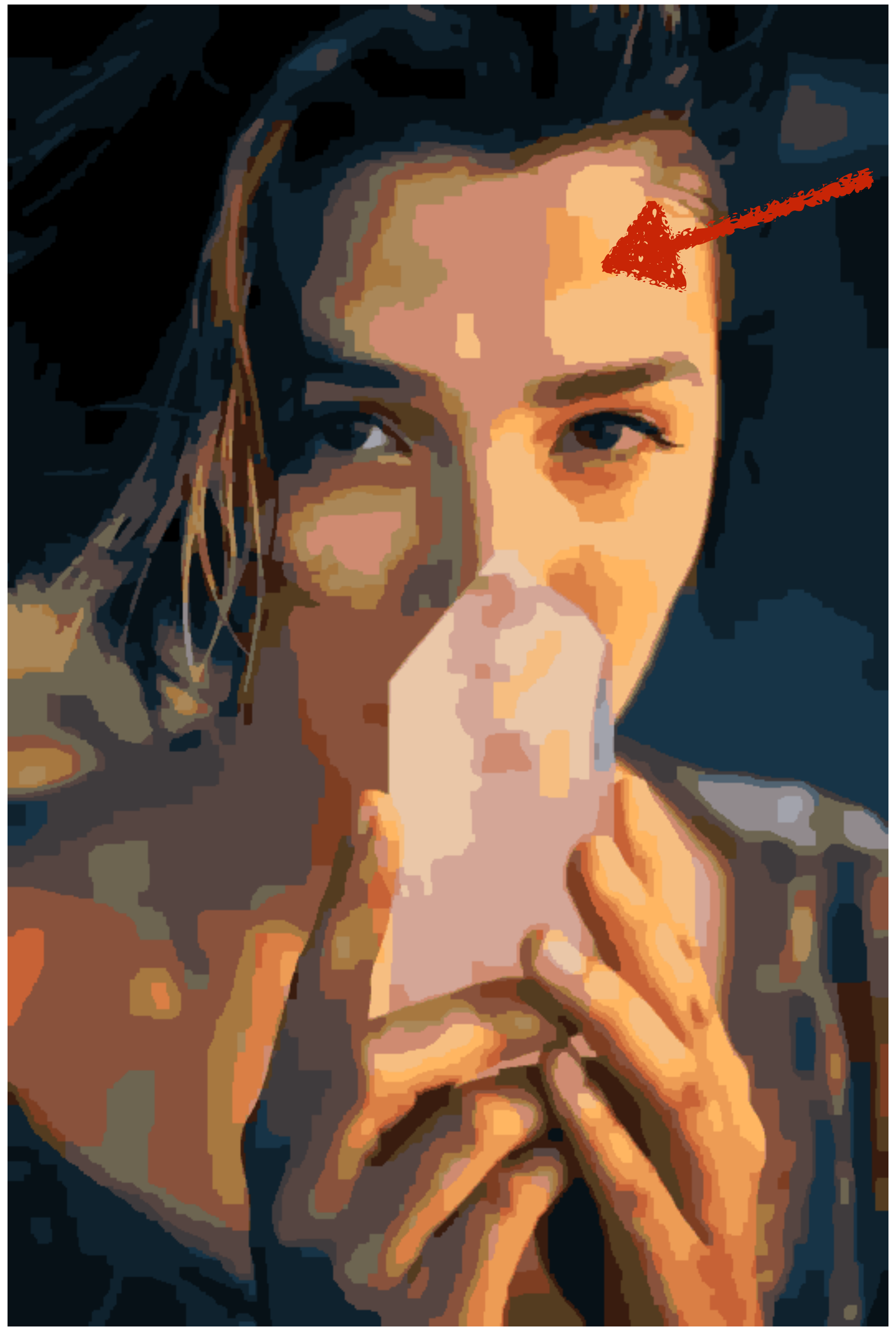
- Step 1: Choose a color palette
- Step 2: Form approximate solid-color regions
- Step 3: Improve region color blends
- Step 4: Improve region boundaries

Our 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

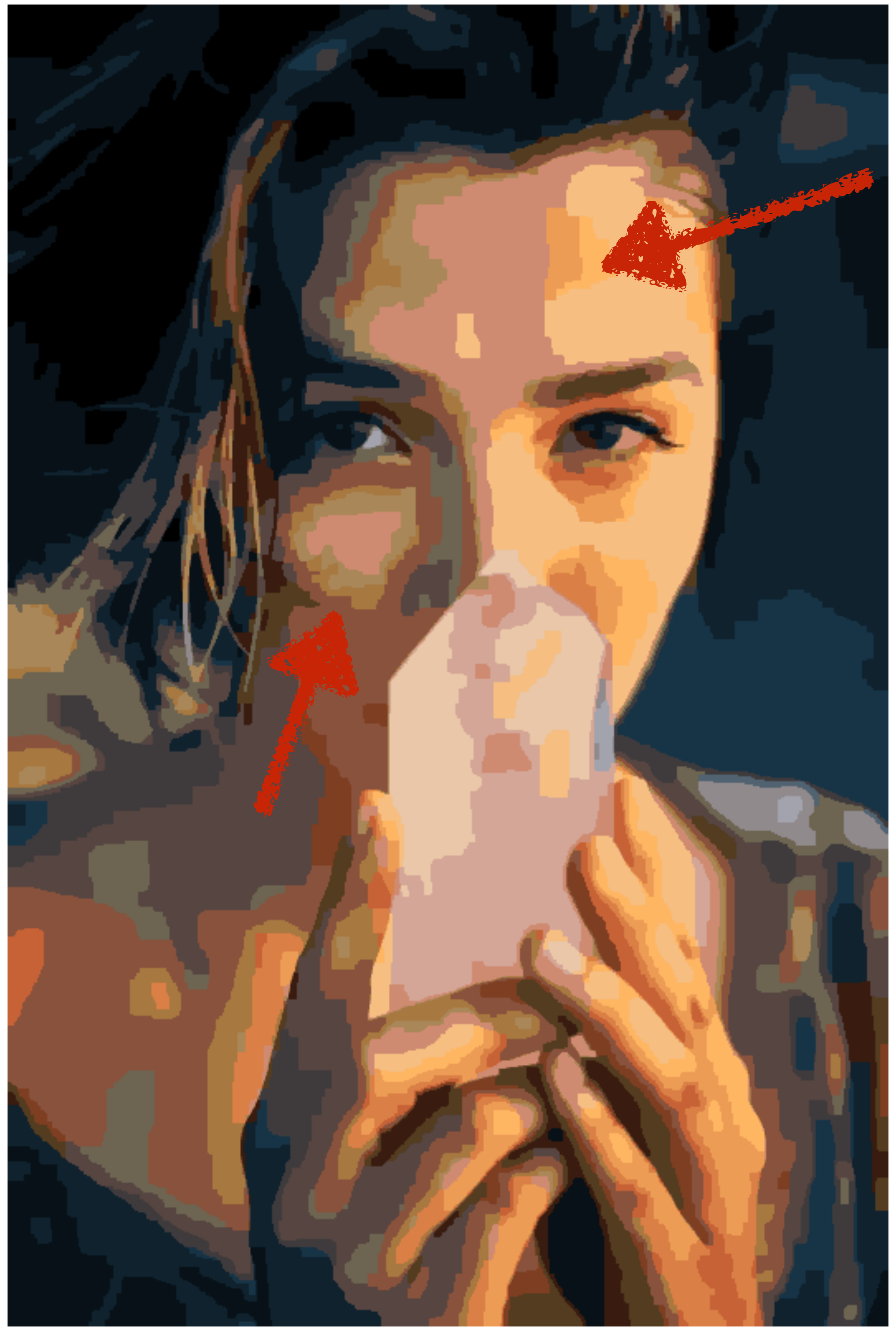
Our Approach

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



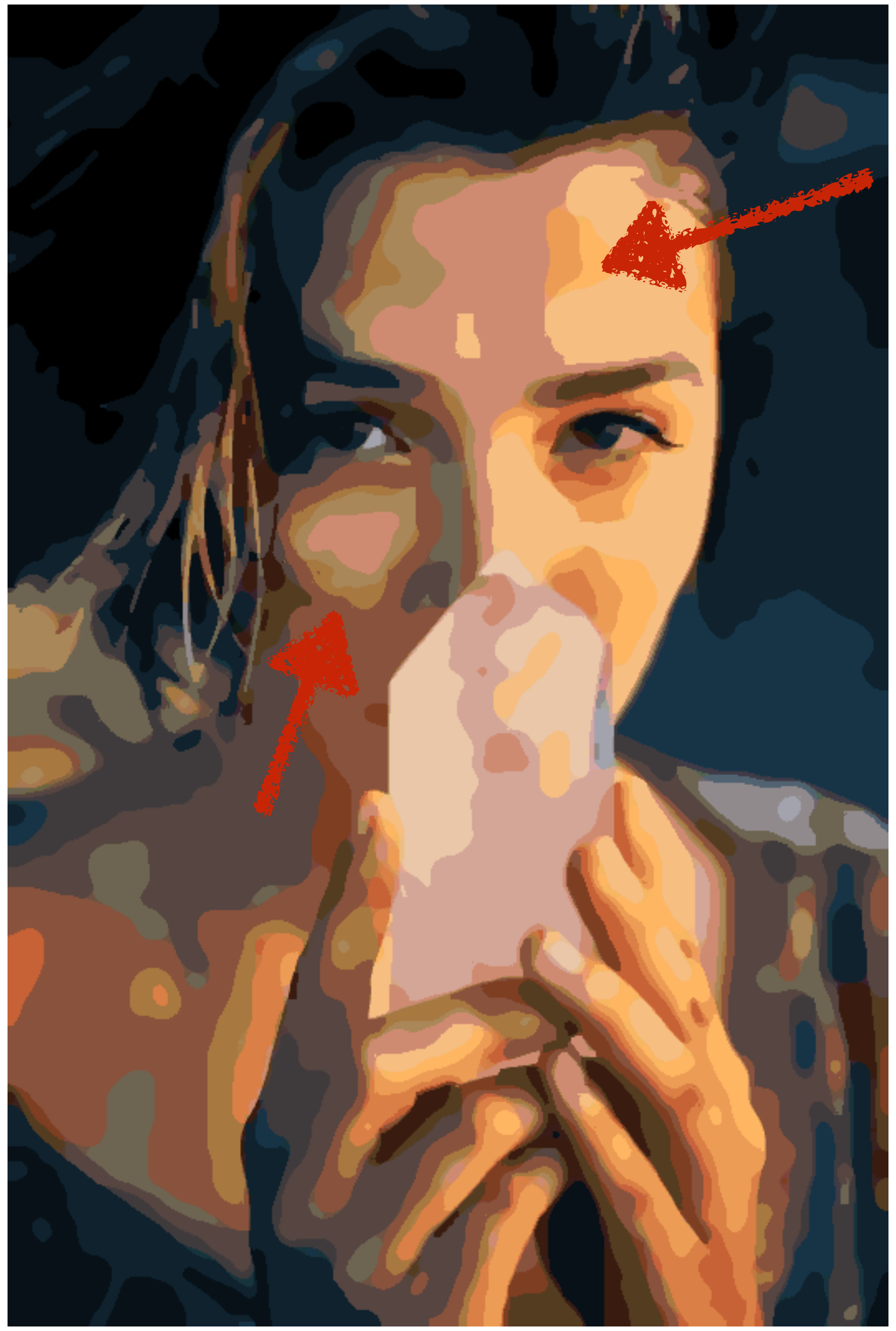
Our Approach

- 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

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

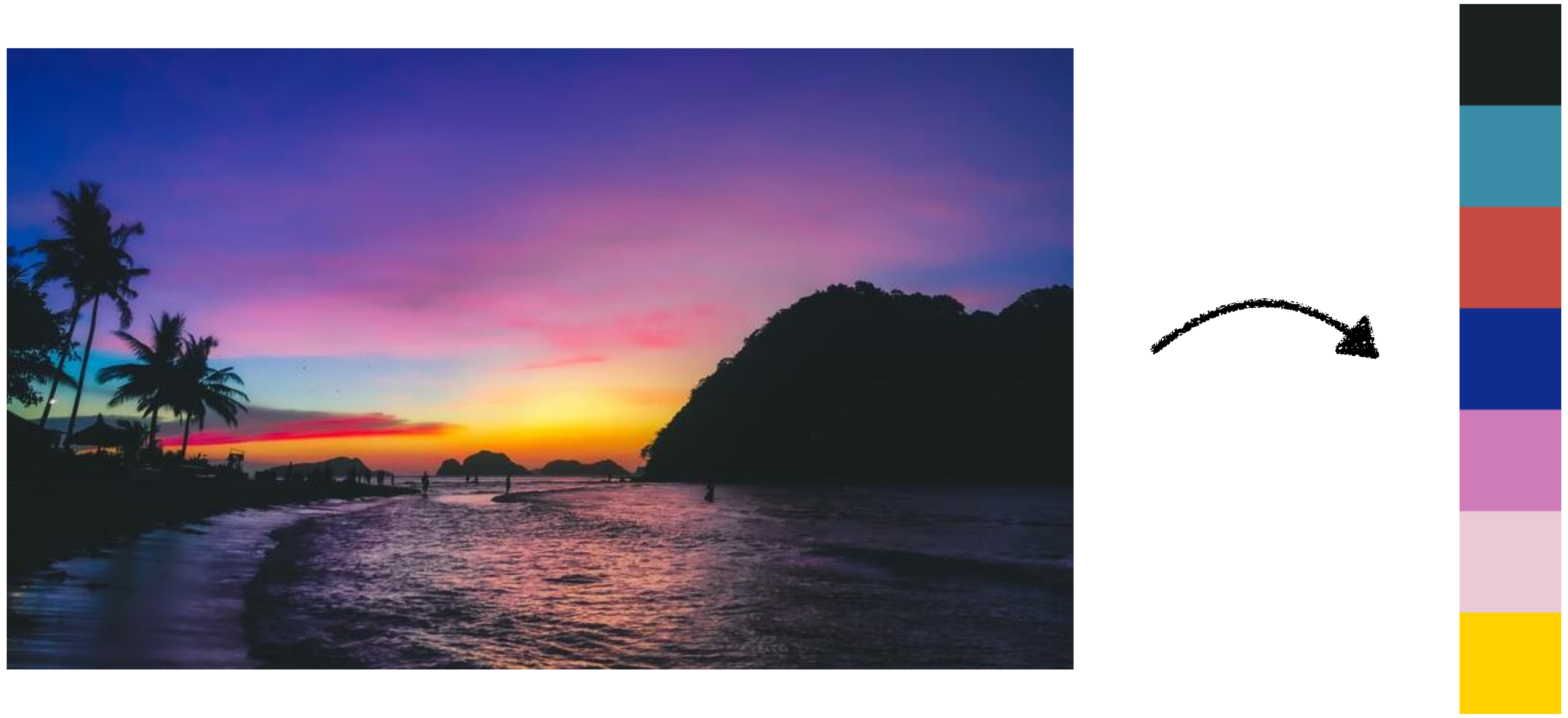
- Goal: Find a small set of colors to represent the image.
- Allow blends of any two palette colors.
- We follow [Tan et al. 2016]'s simplified convex-hull approach.
- Improvement: We use K-means to reduce outlier sensitivity. See paper for details.

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.

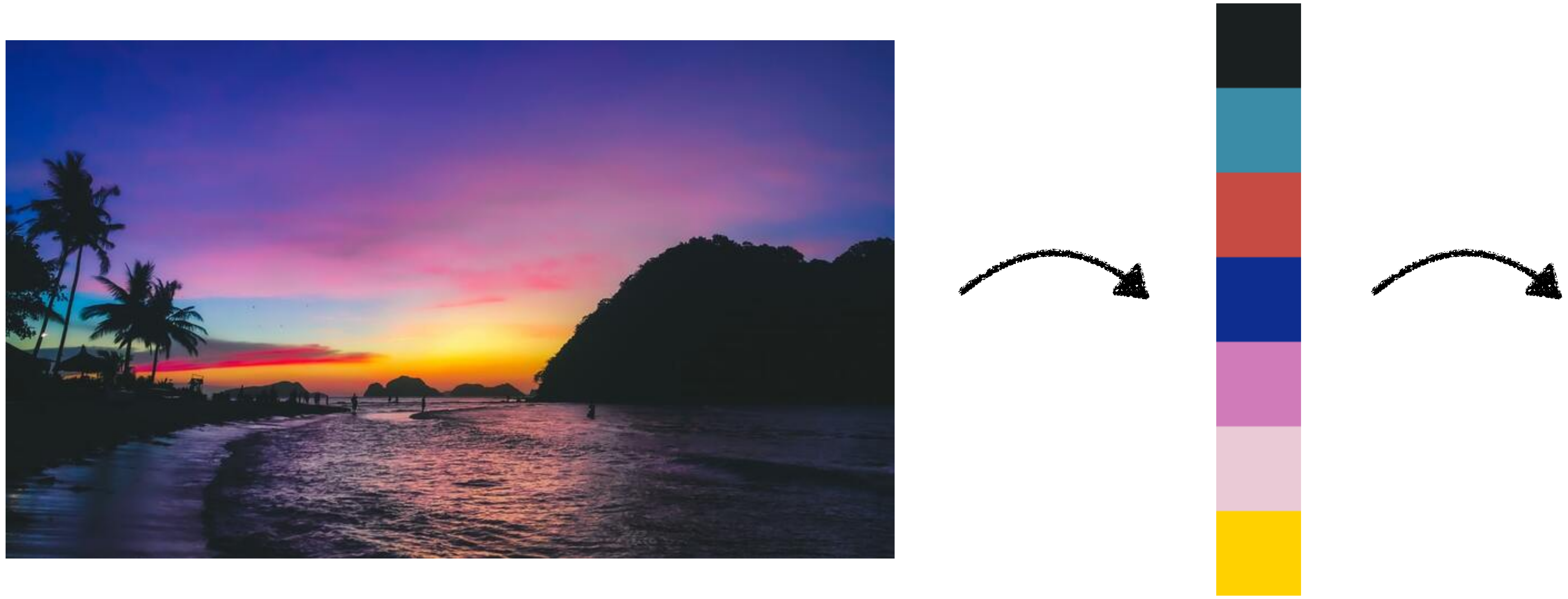
Step 1: Convex-hull based palette extraction

- 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.



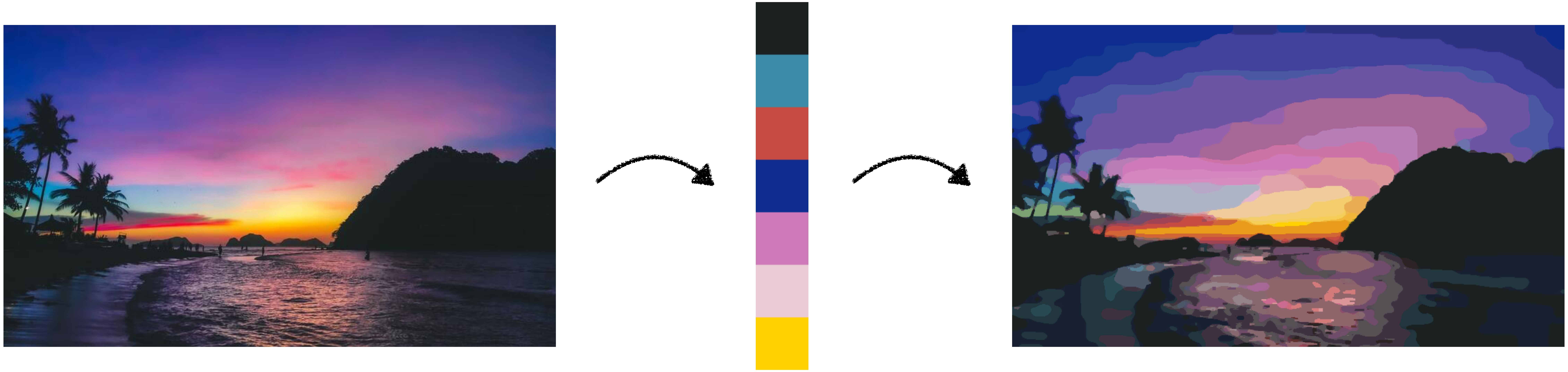
Step 1: Convex-hull based palette extraction

- 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

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.

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

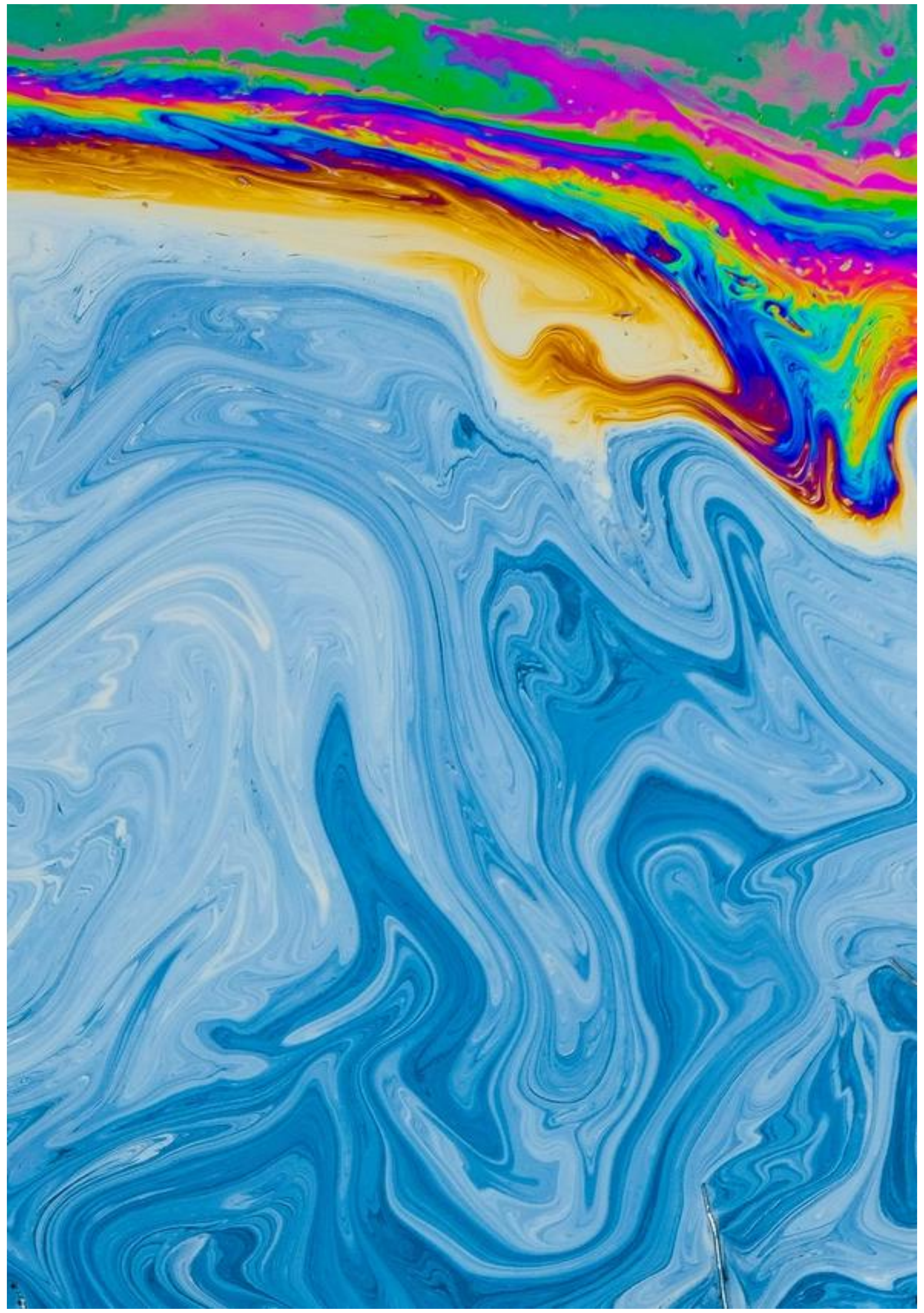
  - $E_{pairwise}$ penalizes neighboring pixels with different region labels L_p and L_q.

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

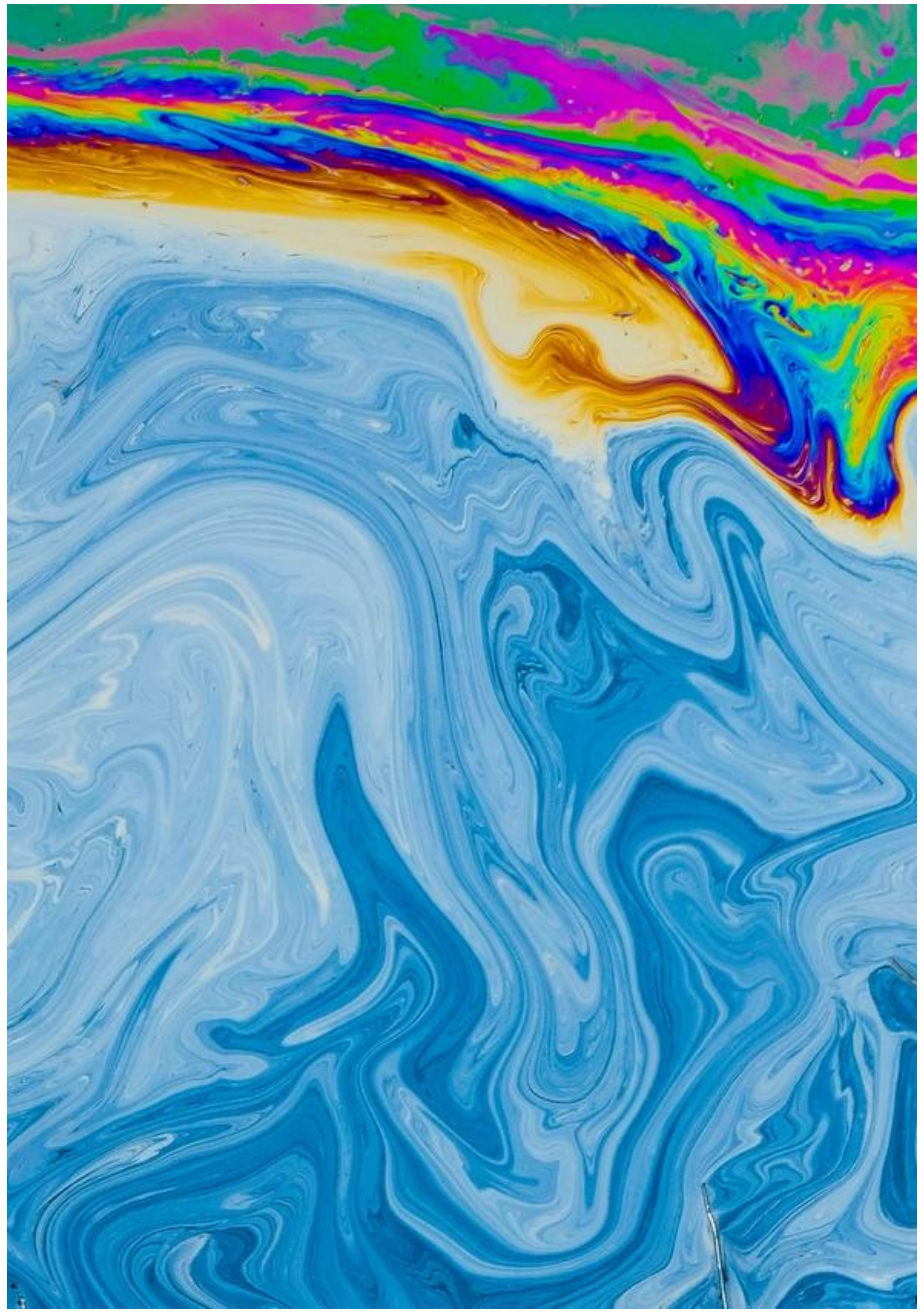
- Overall objective function: $E(f) = E_{data}(f) + \lambda E_{pairwise}(f)$
  - λ controls the clumpiness of the regions in the output.
  - We solve this problem with multi-label optimization [Boykov and Kolmogorov 2001].

Step 2: Rough region and color assignment

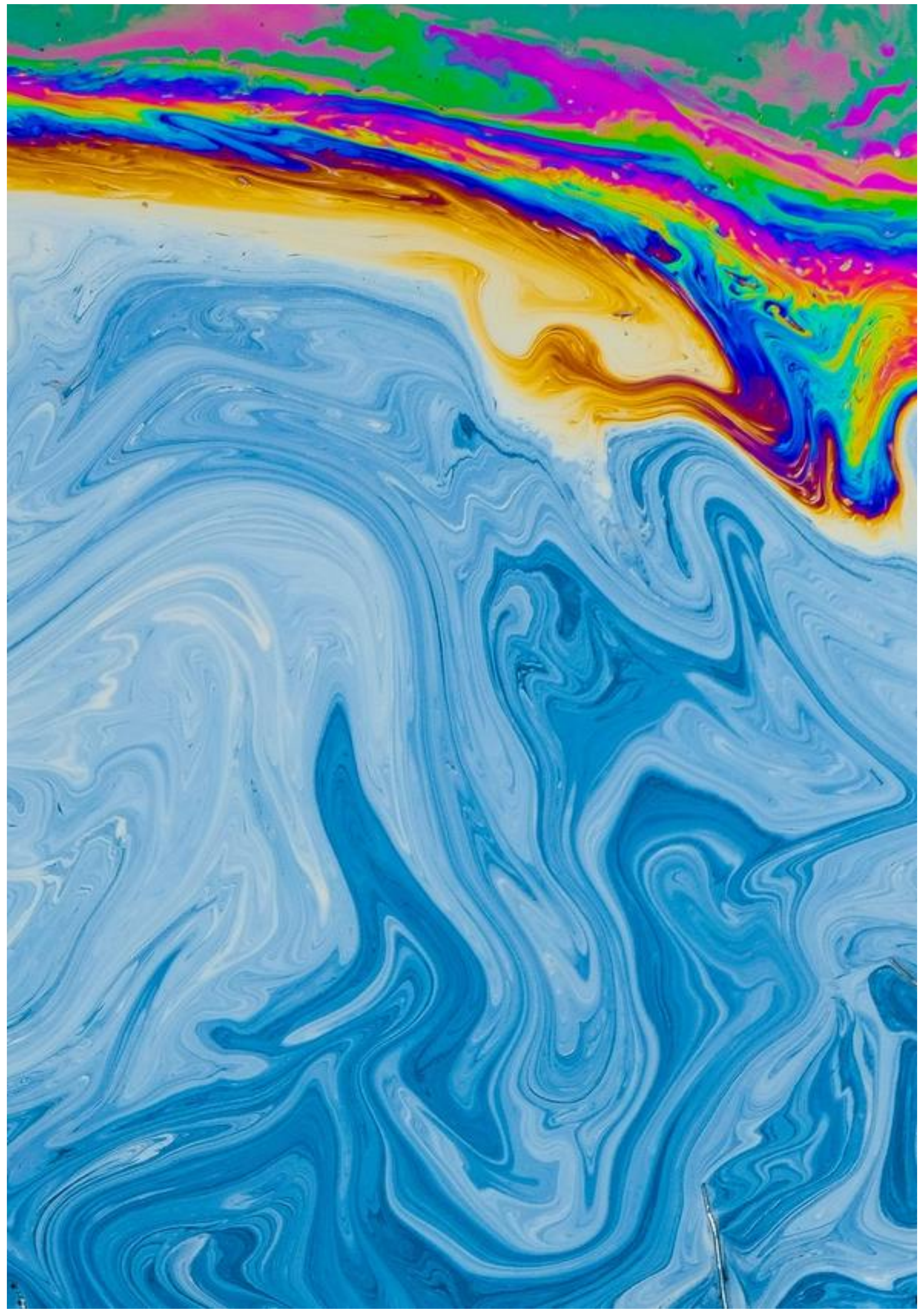
Step 2: Rough region and color assignment



Step 2: Rough region and color assignment

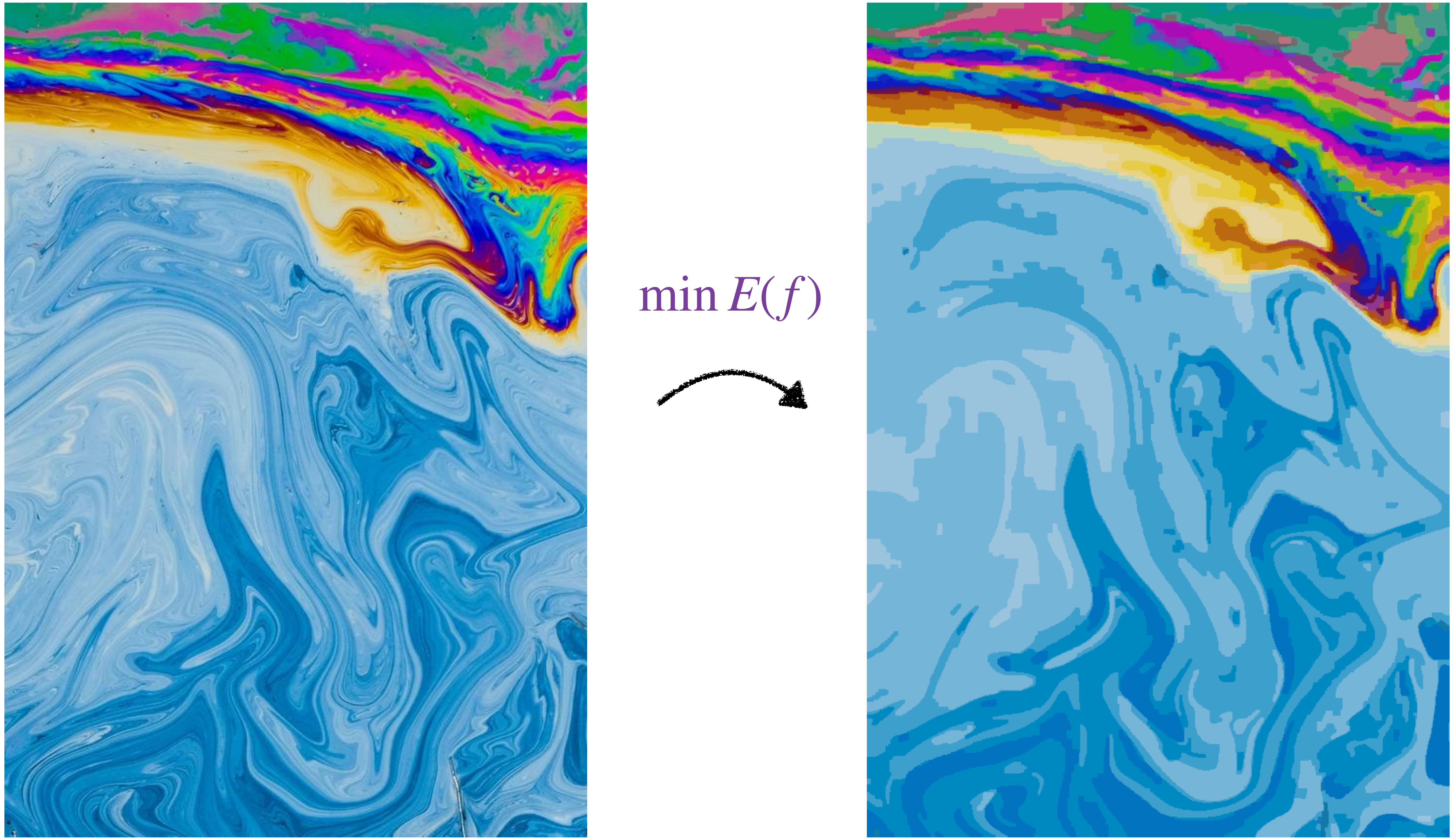


Step 2: Rough region and color assignment



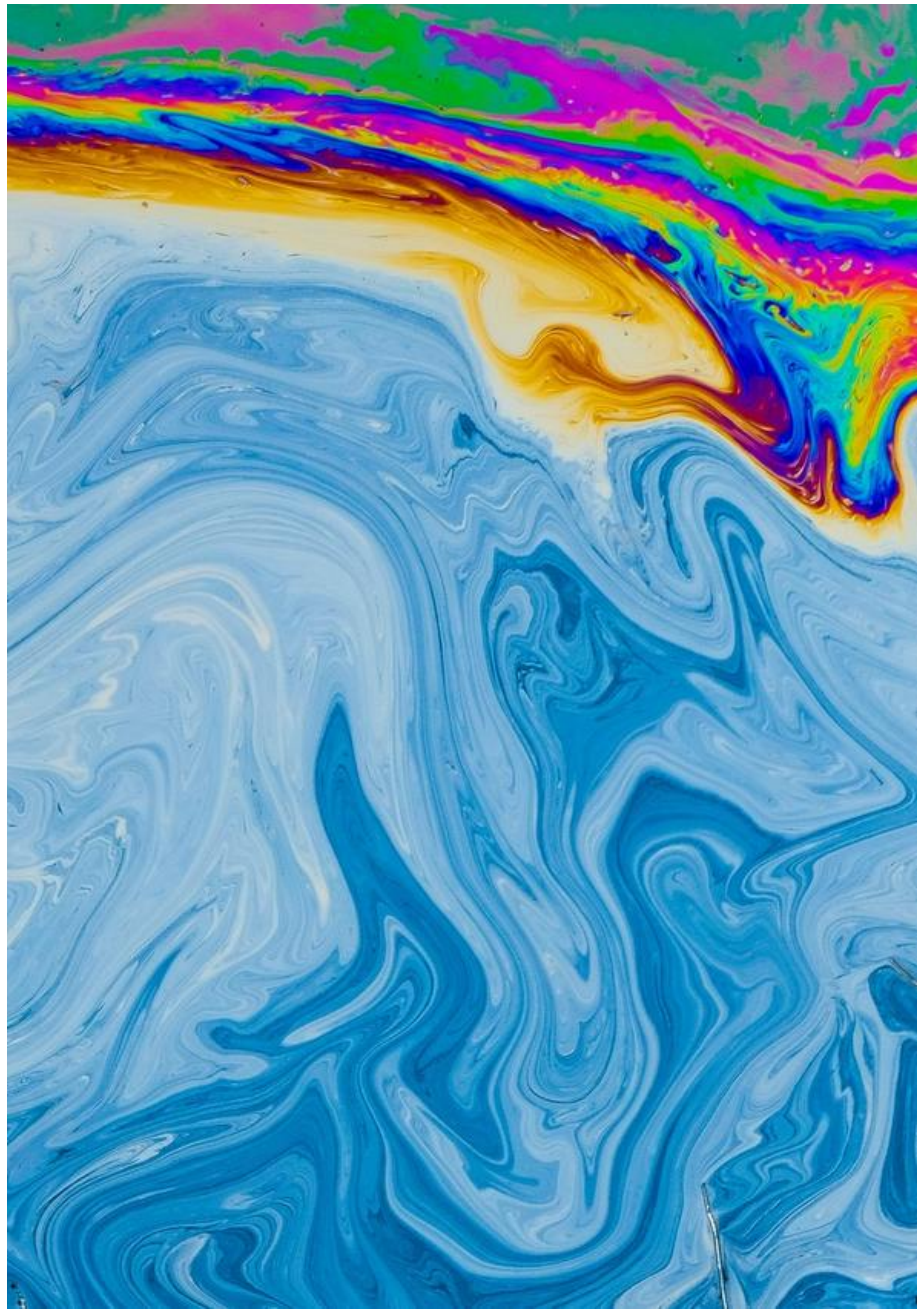
$\min E(f)$

Step 2: Rough region and color assignment

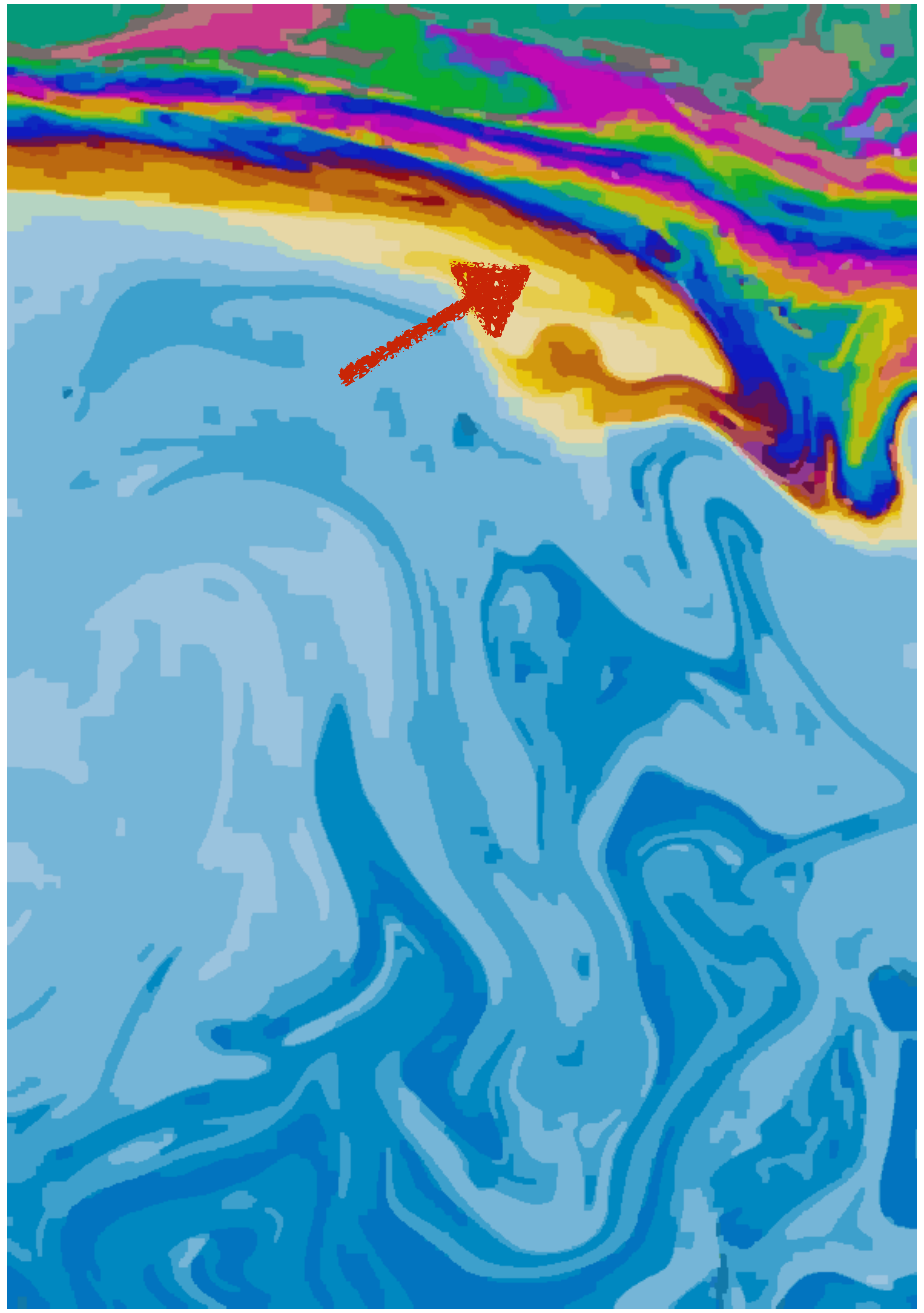


$\min E(f)$

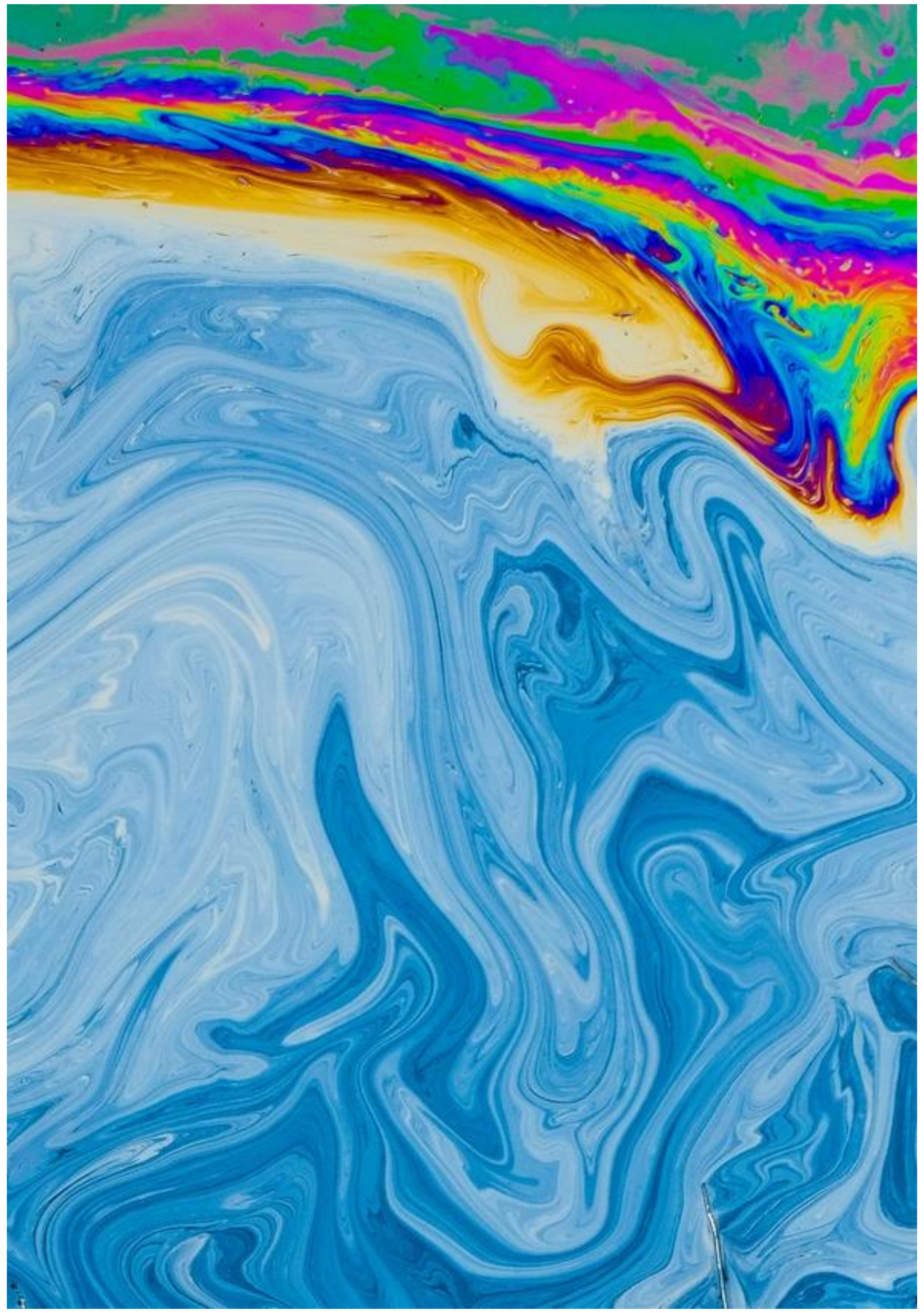
Step 2: Rough region and color assignment



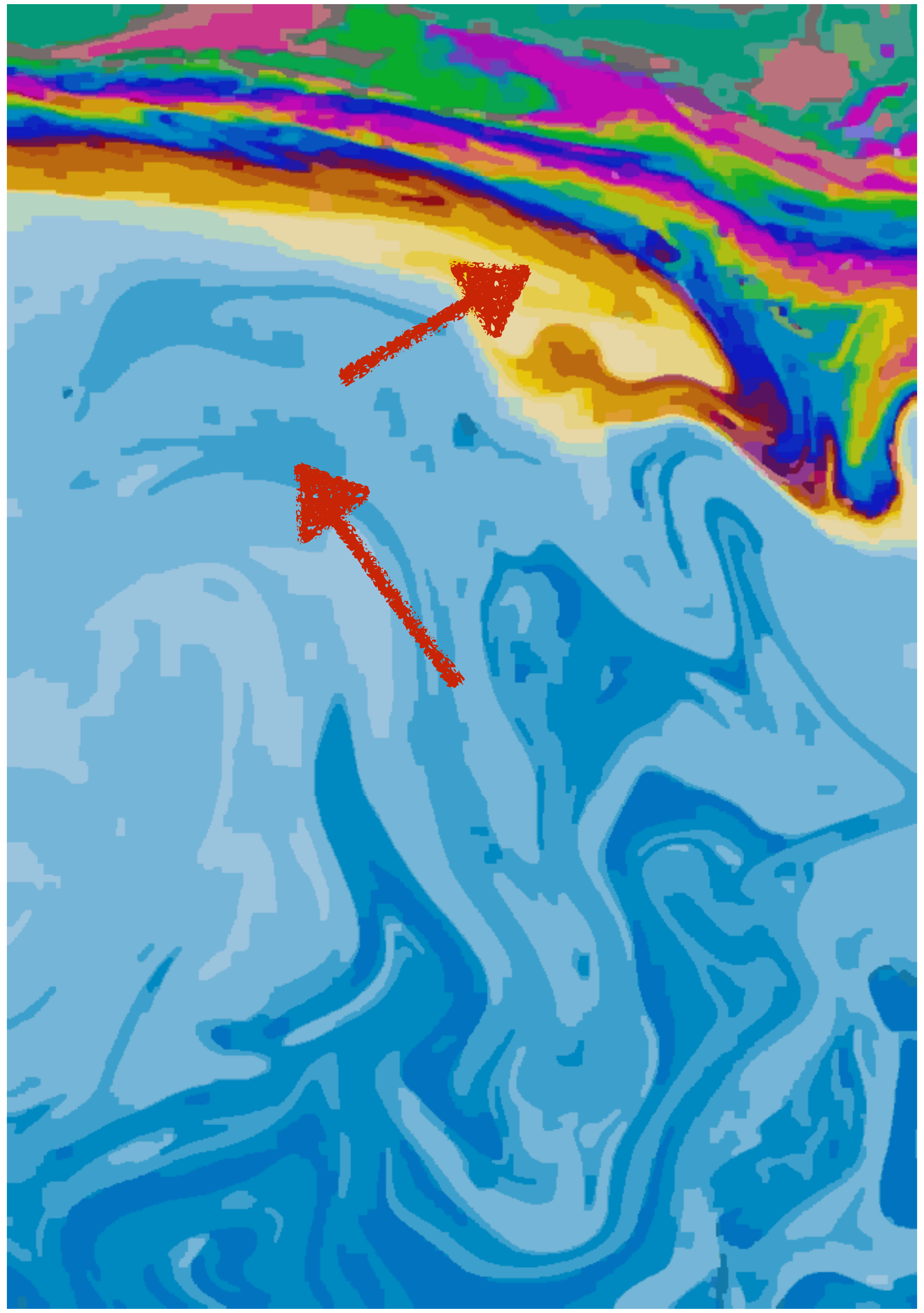
$\min E(f)$



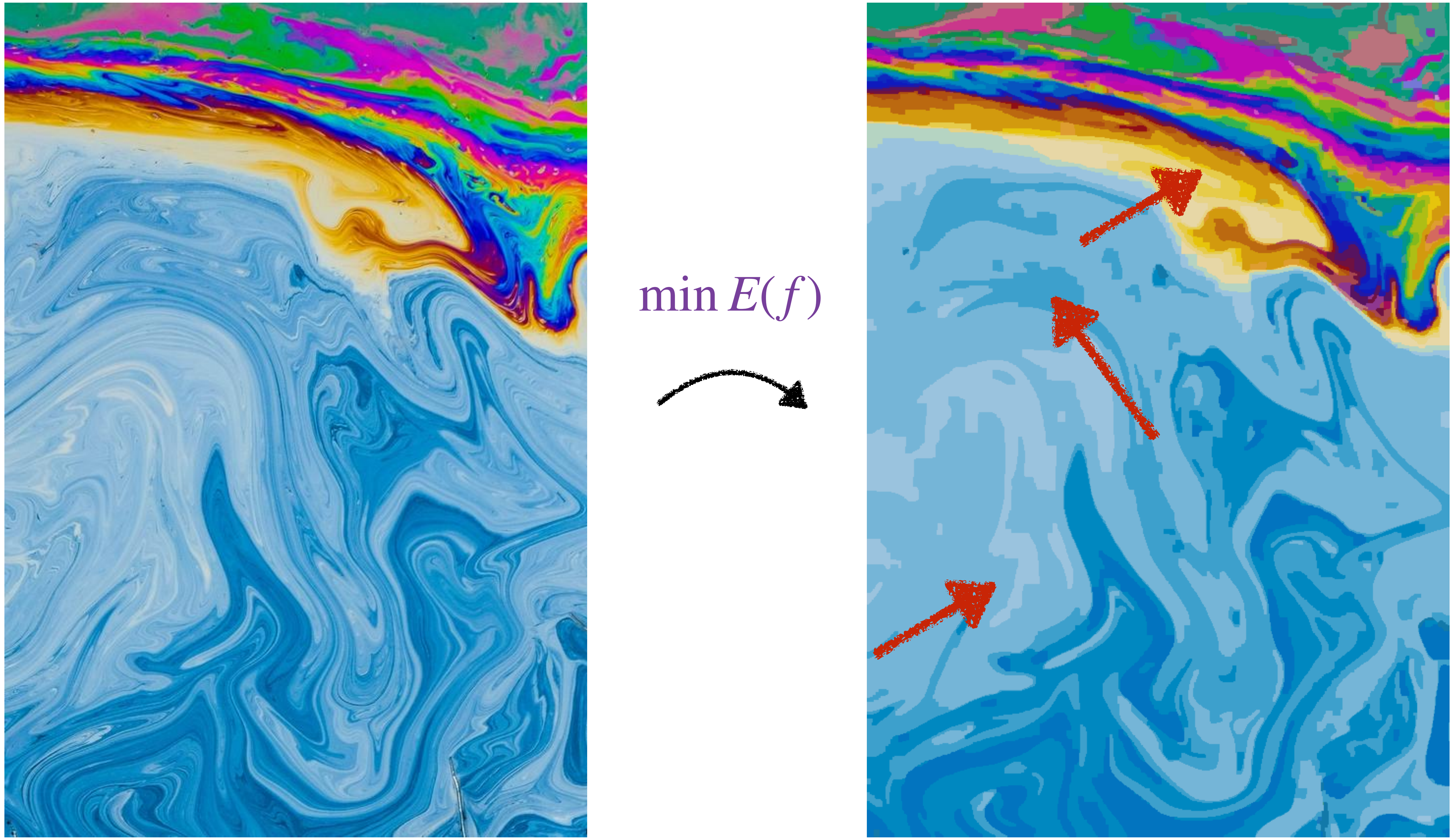
Step 2: Rough region and color assignment



$\min E(f)$



Step 2: Rough region and color assignment



$\min E(f)$

Step 2: Rough region and color assignment

Step 2: Rough region and color assignment

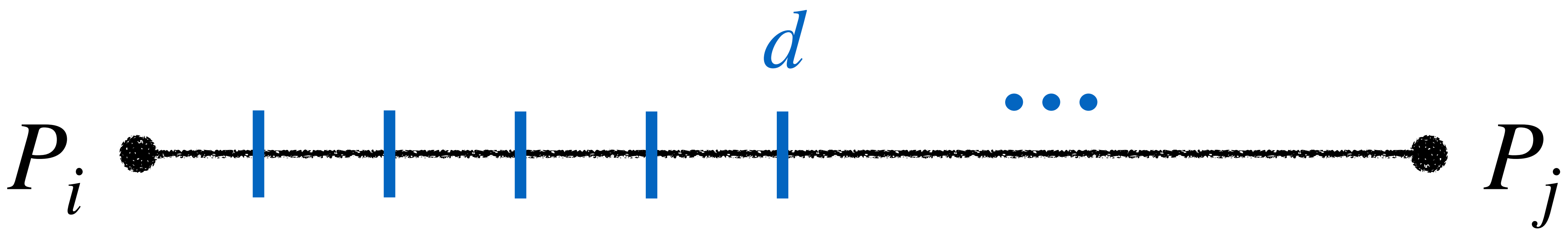
- Palette colors P from palette extraction in step 1.

Step 2: Rough region and color assignment

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

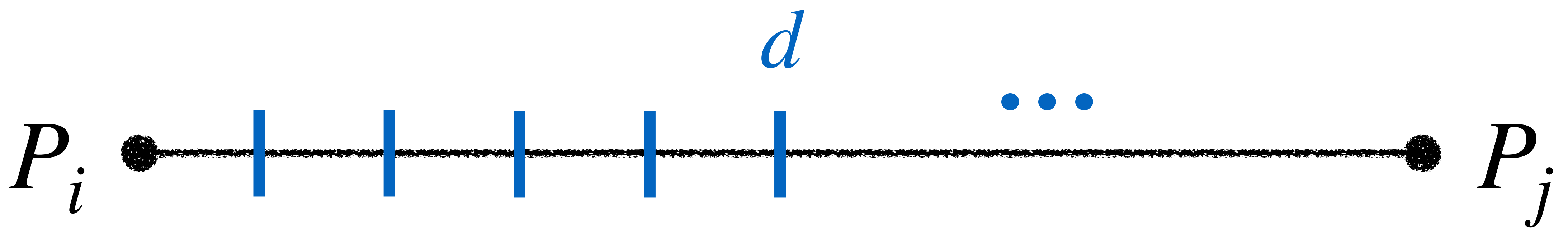
Step 2: Rough region and color assignment

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



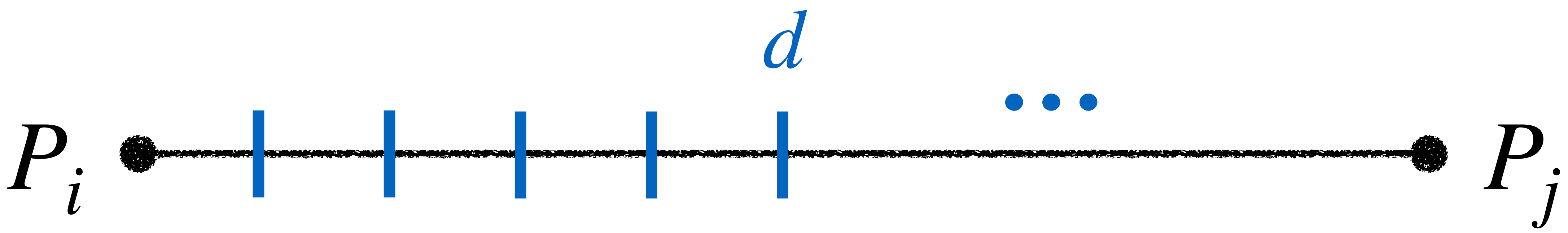
Step 2: Rough region and color assignment

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



Step 2: Rough region and color assignment

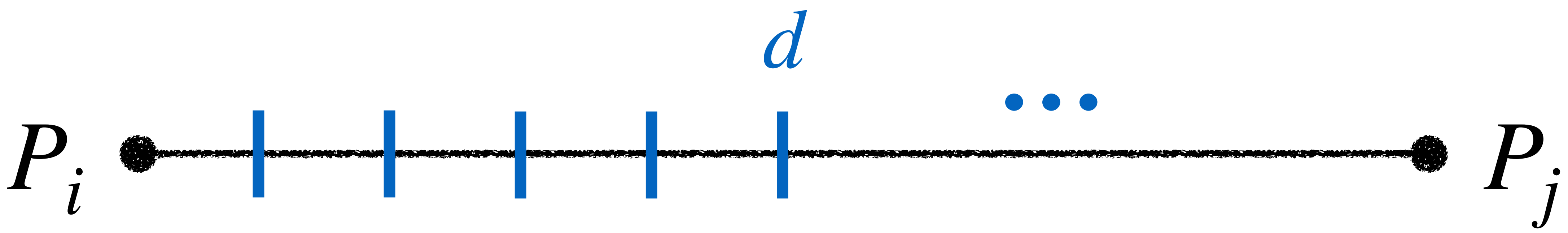
- Palette colors P from palette extraction in step 1.
- Pairwise blends from palette colors P with weights.

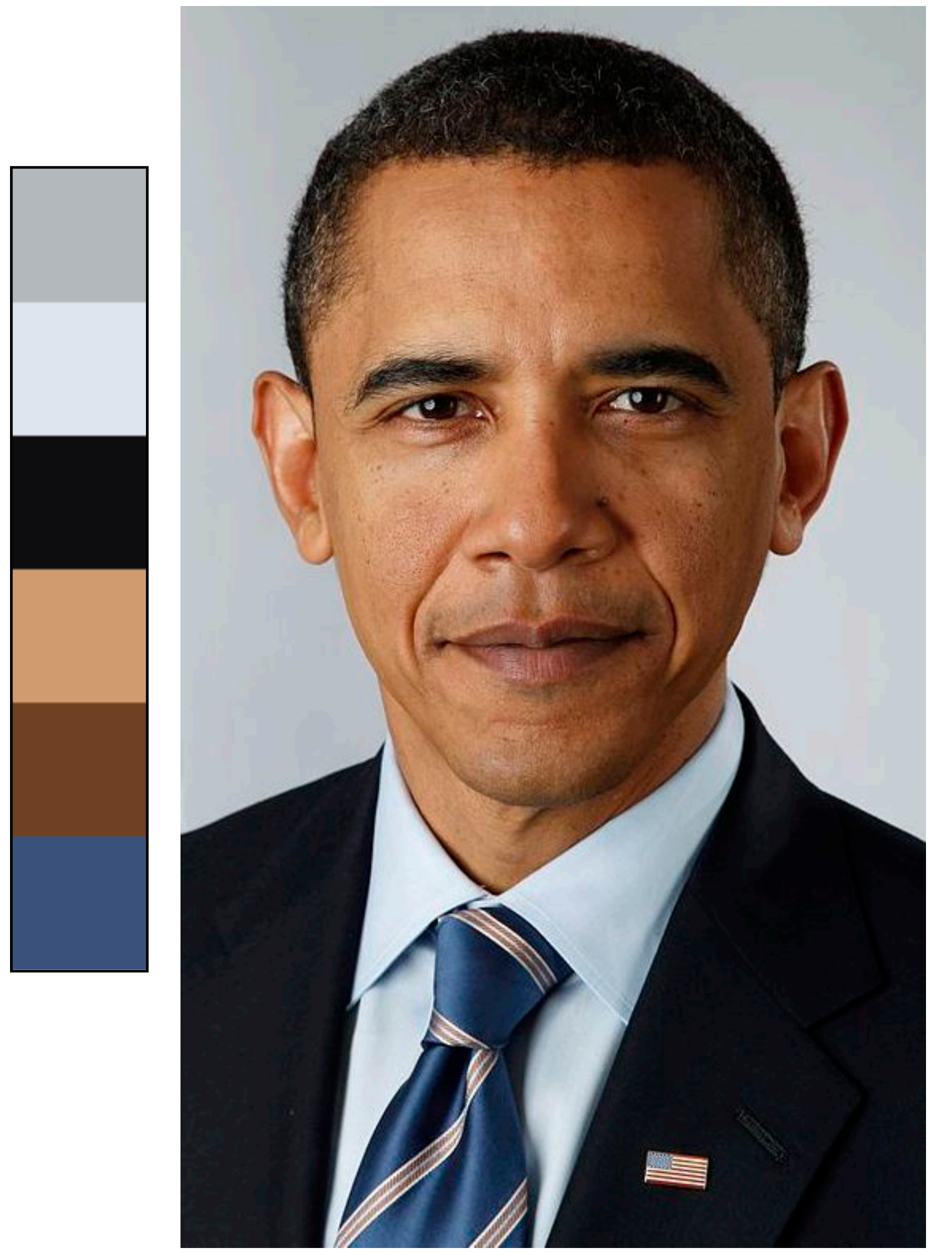


Input

Step 2: Rough region and color assignment

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

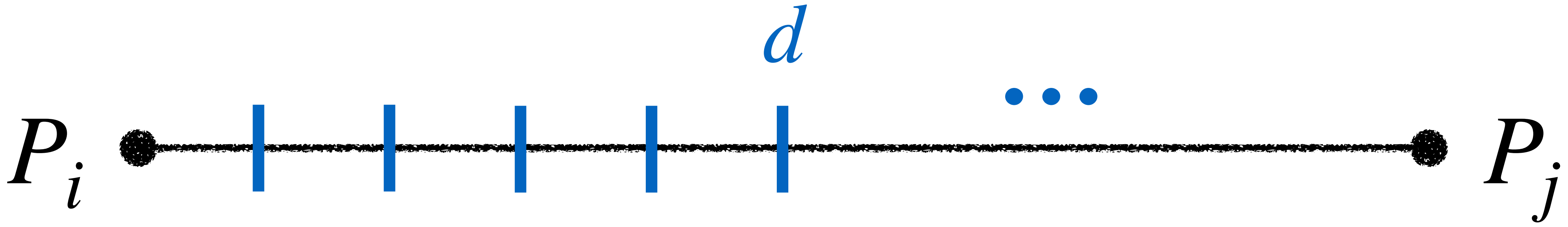


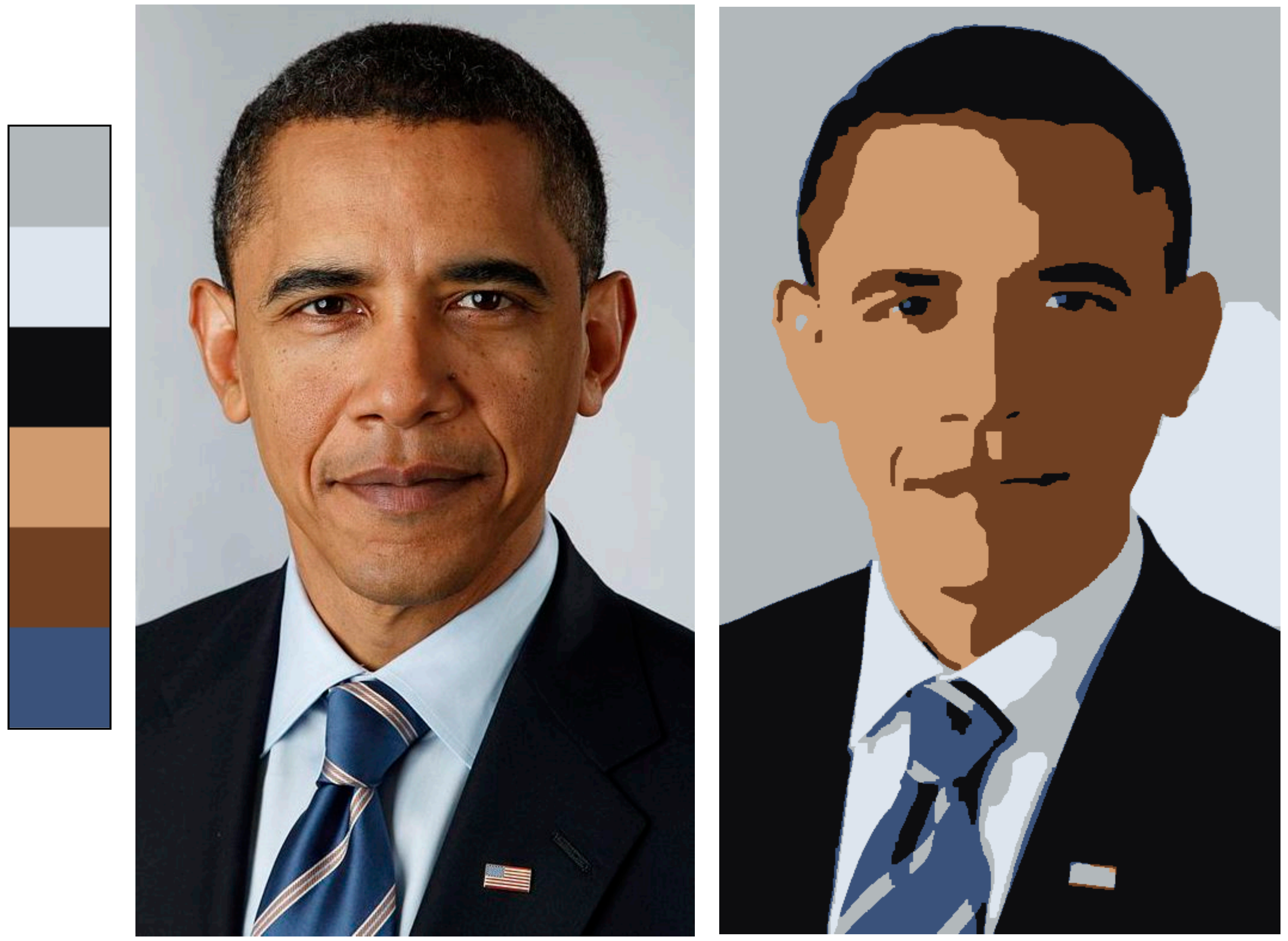


Input

Step 2: Rough region and color assignment

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

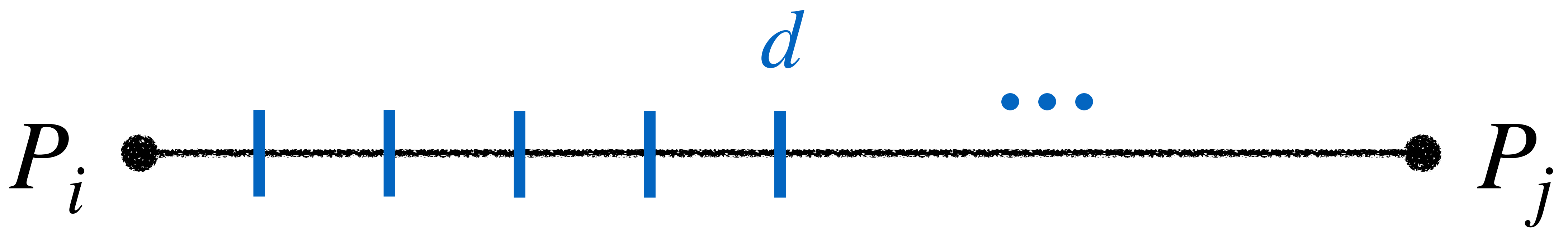


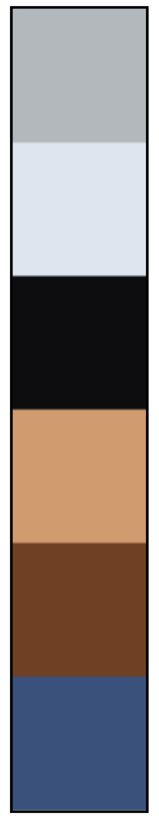


Input

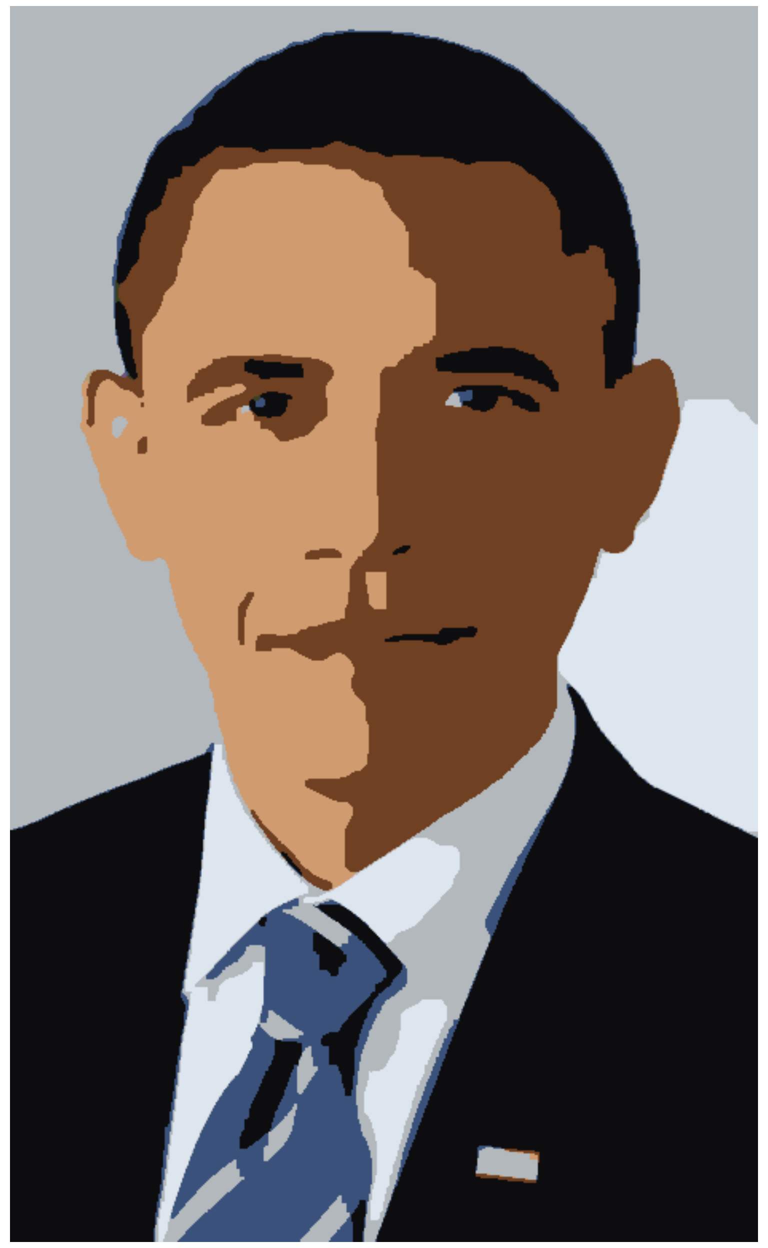
Step 2: Rough region and color assignment

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





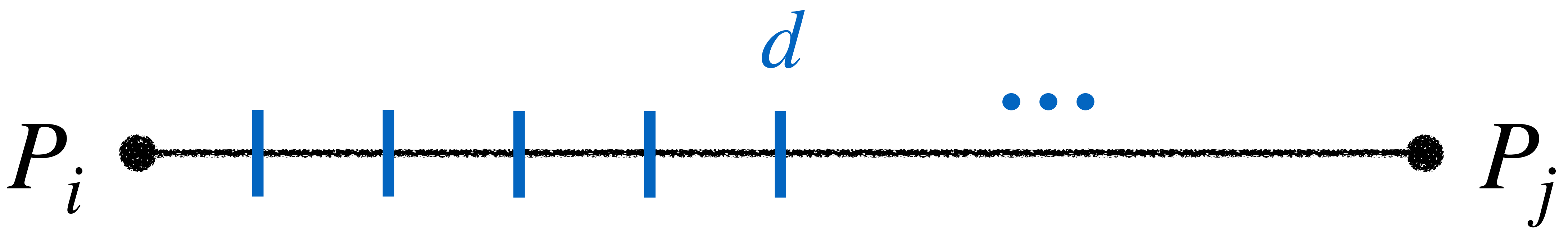
Input

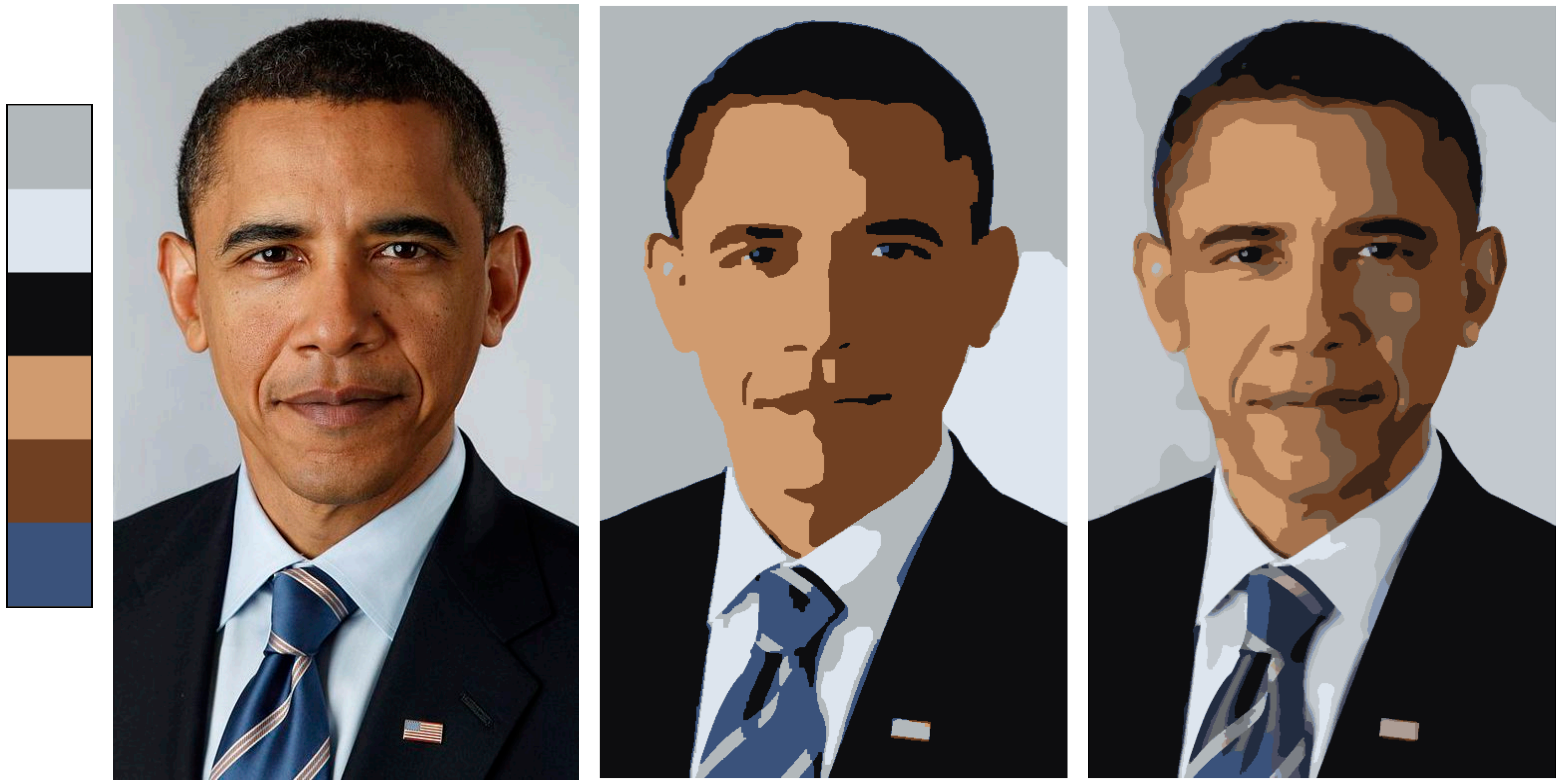


$d = 0$

Step 2: Rough region and color assignment

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



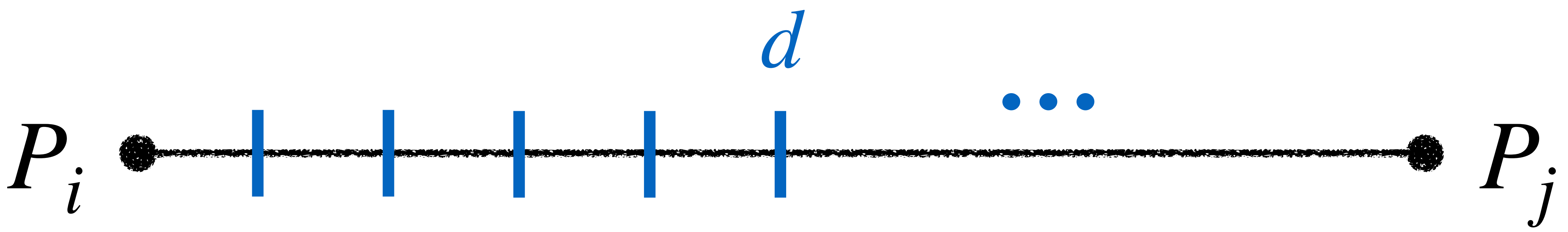


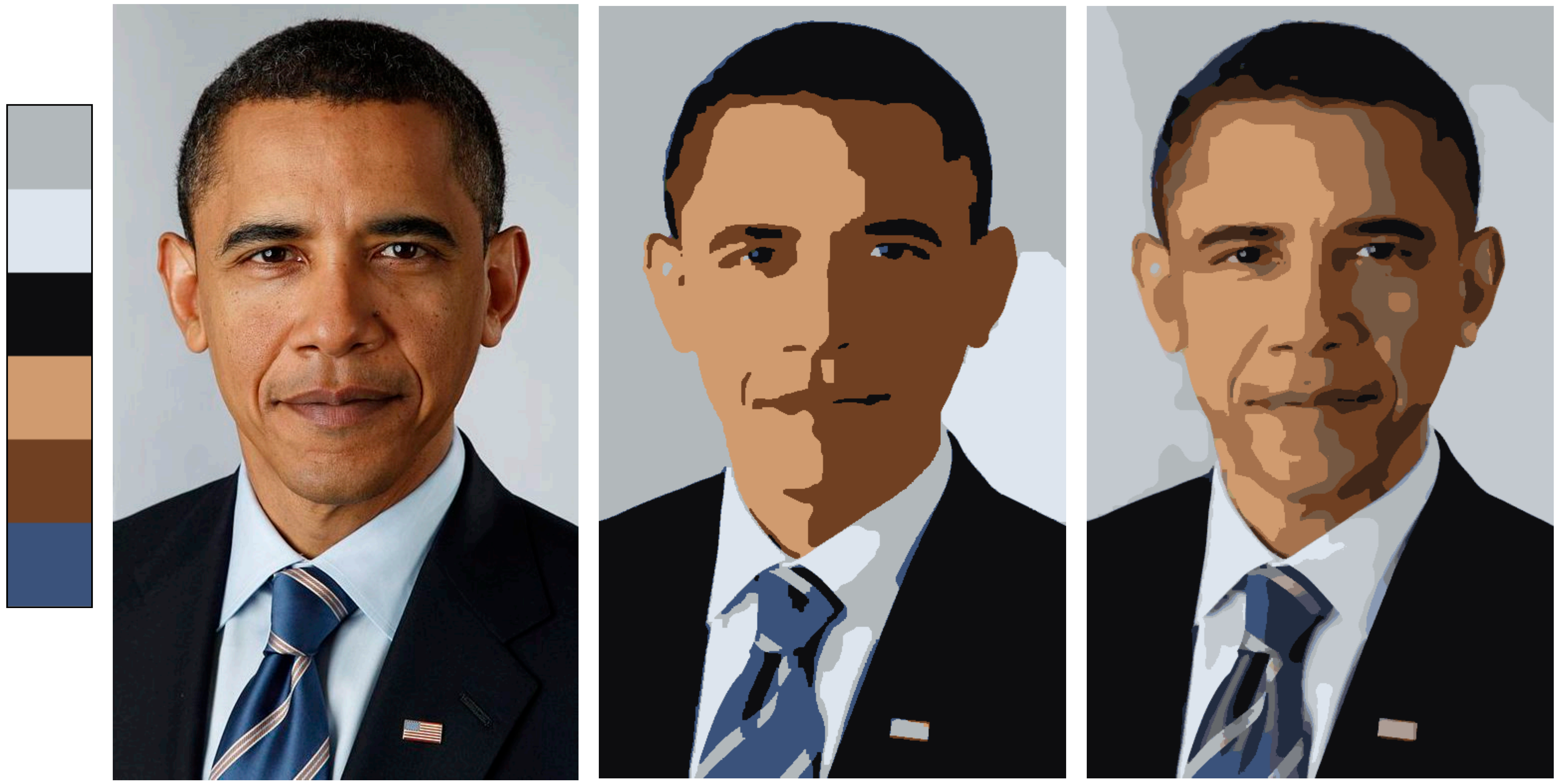
Input

$d = 0$

Step 2: Rough region and color assignment

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

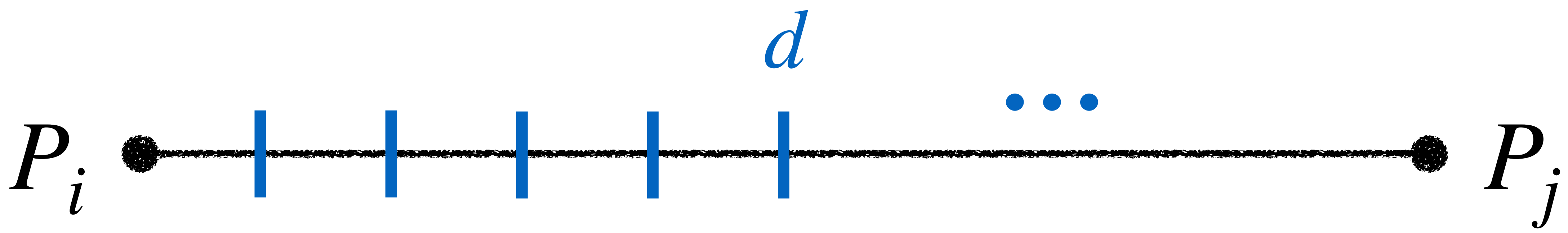


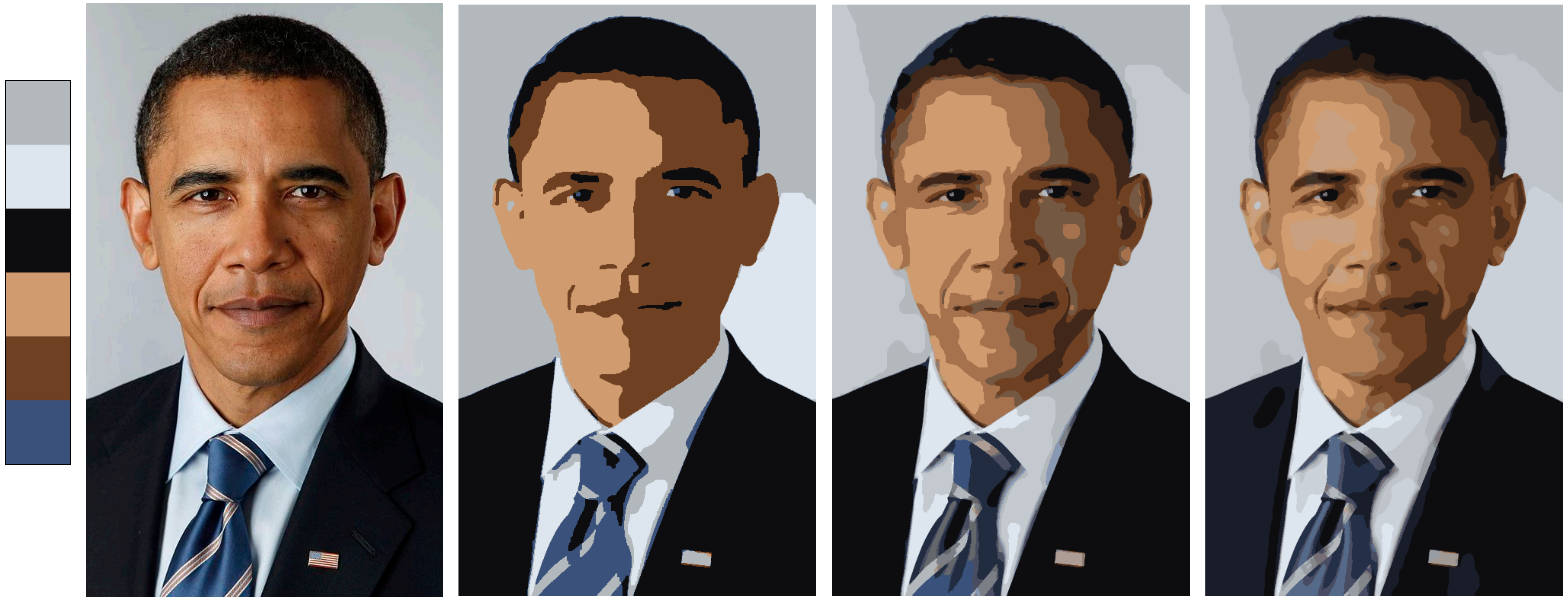


Input $d = 0$ $d = 1$

Step 2: Rough region and color assignment

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





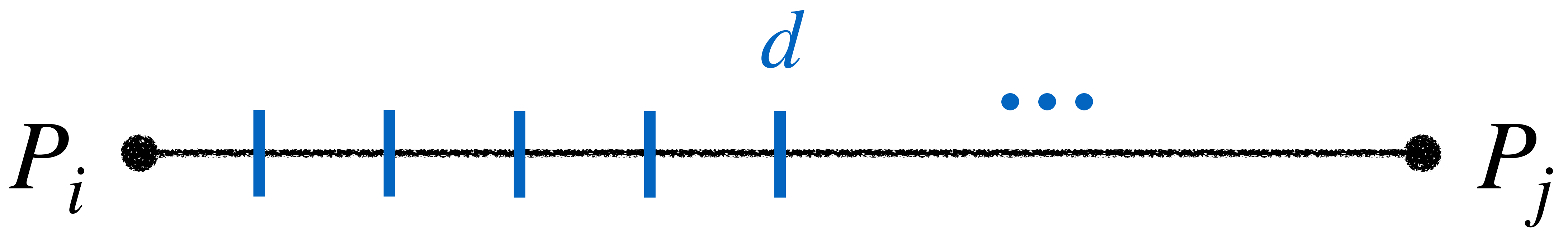
Input

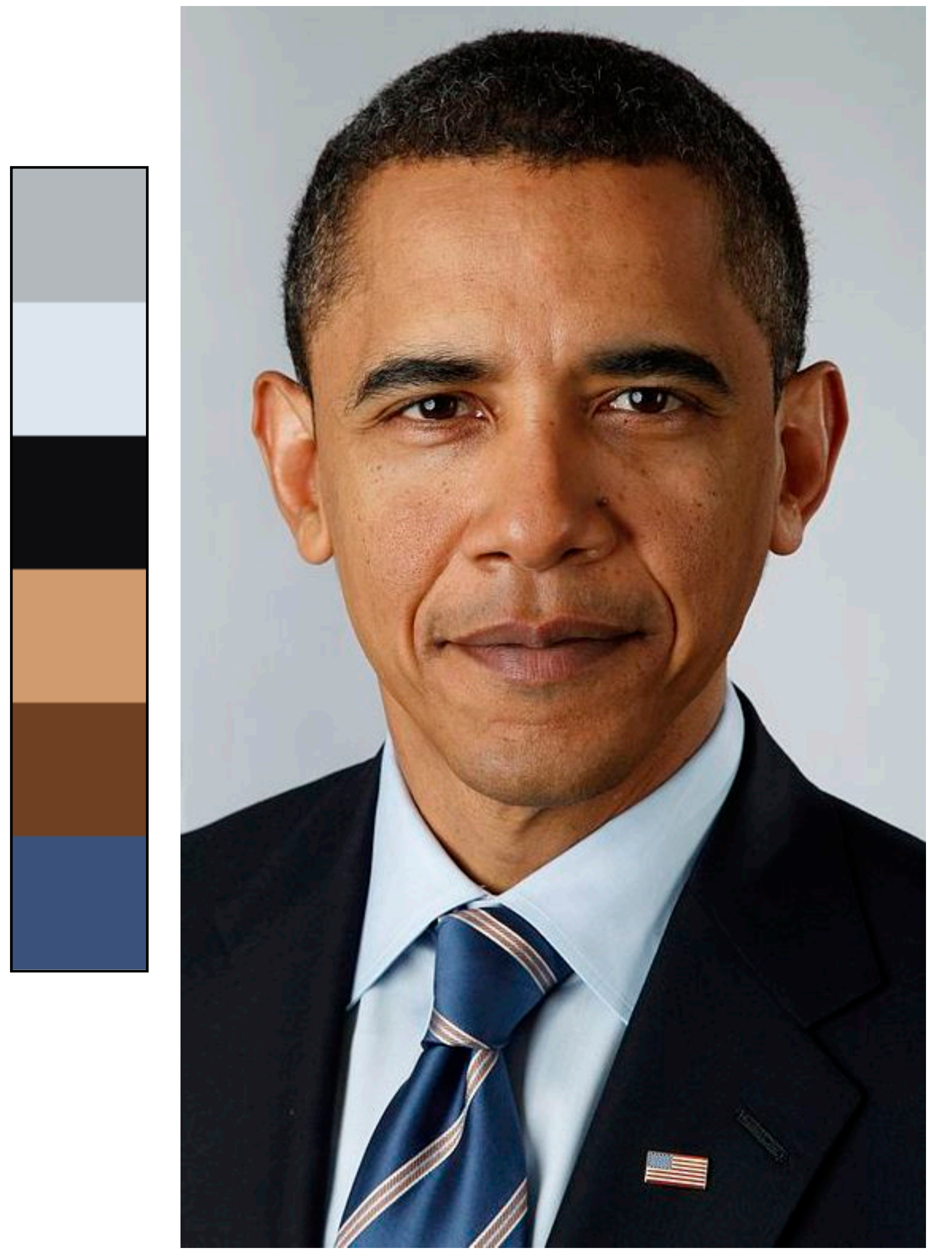
$d = 0$

$d = 1$

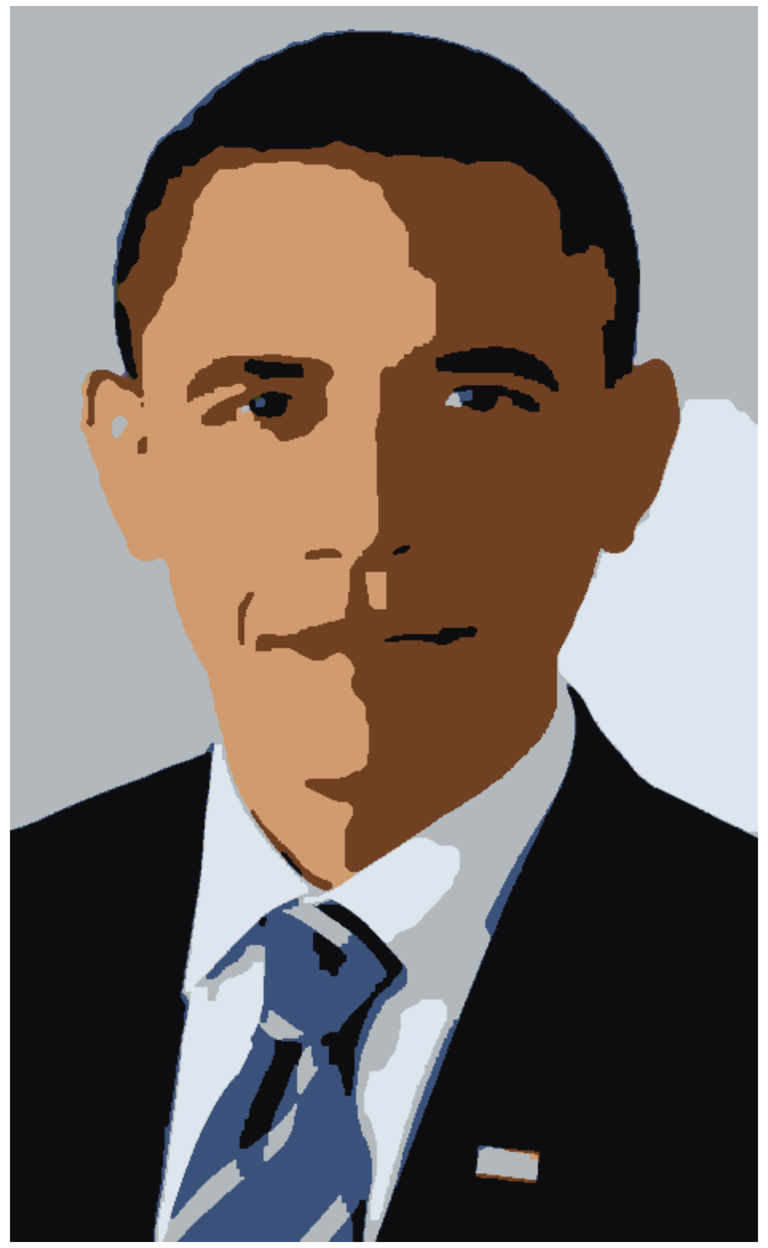
Step 2: Rough region and color assignment

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

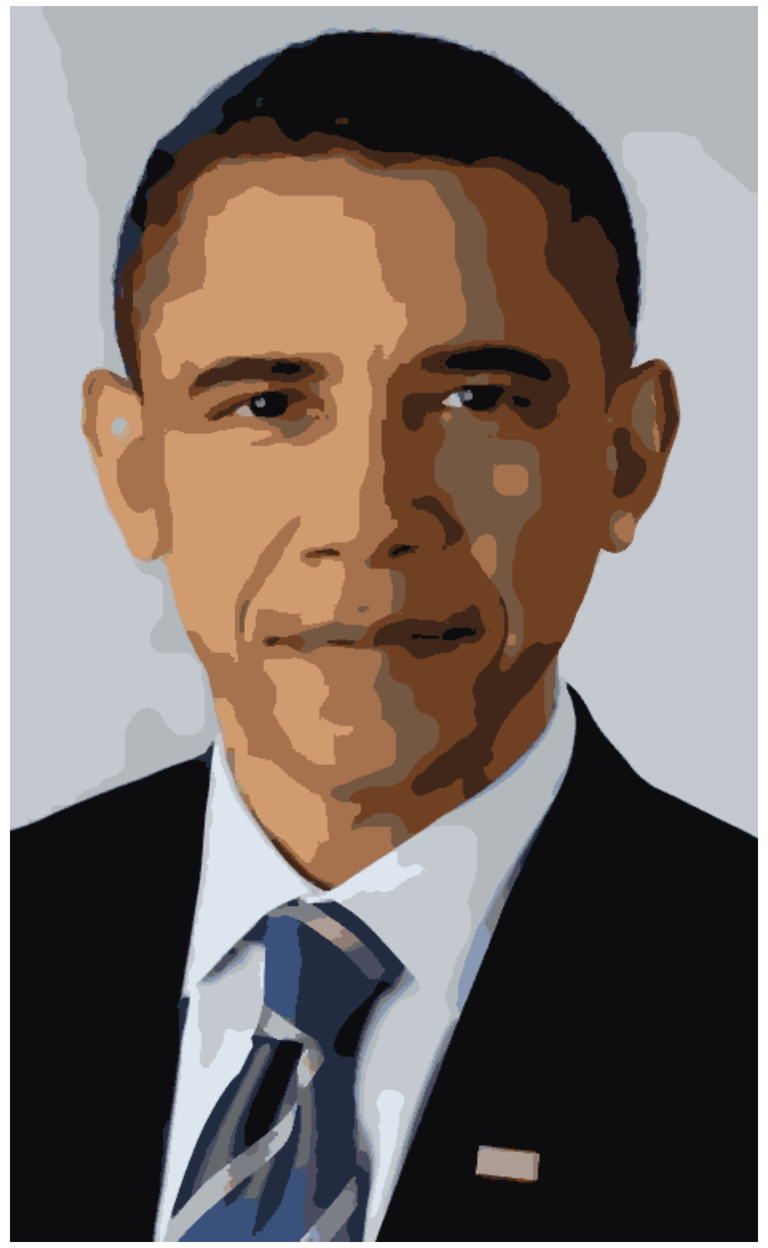




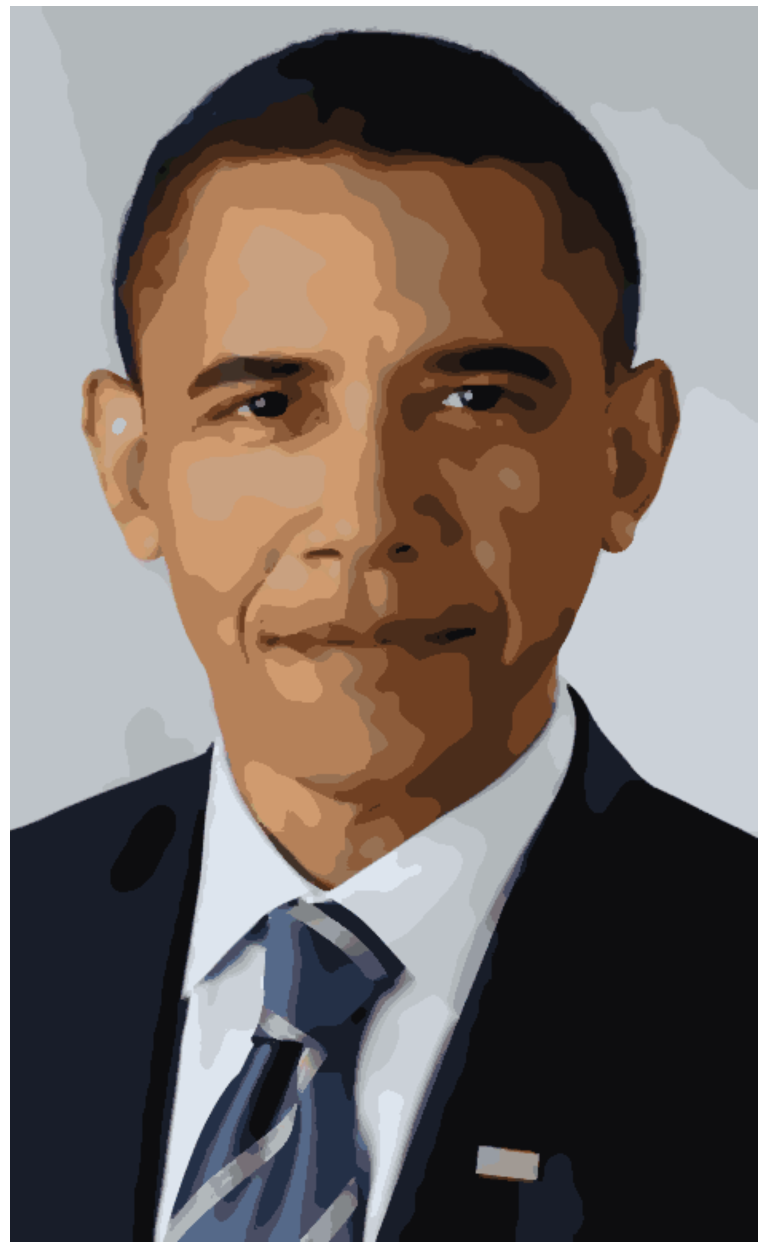
Input



$d = 0$



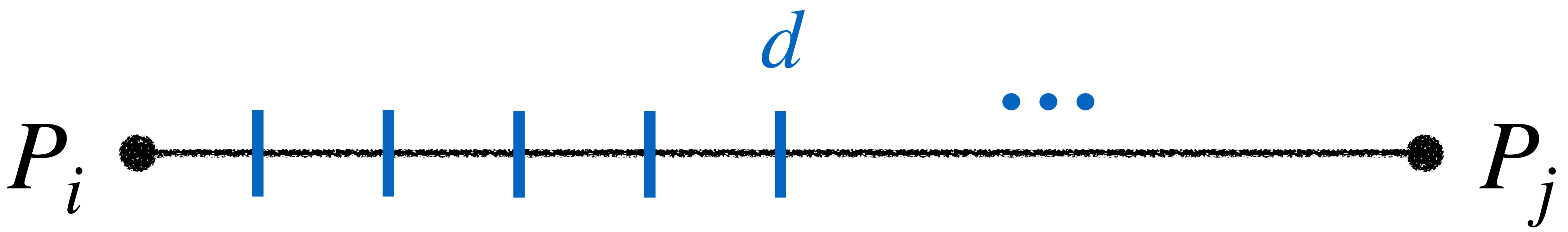
$d = 1$

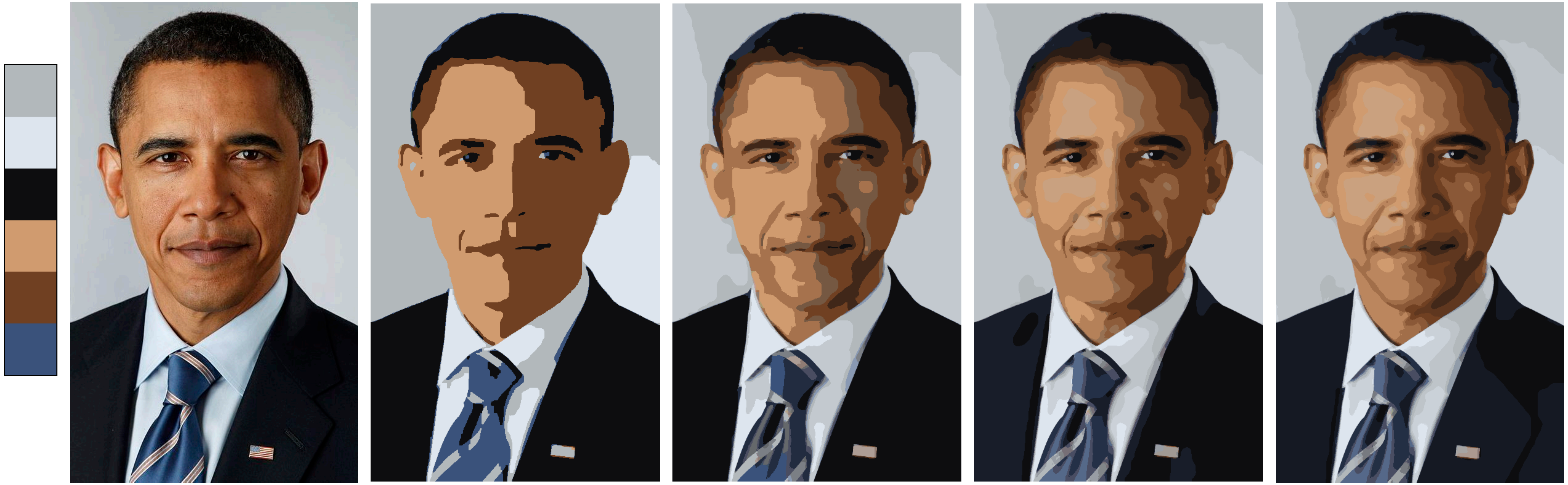


$d = 2$

Step 2: Rough region and color assignment

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

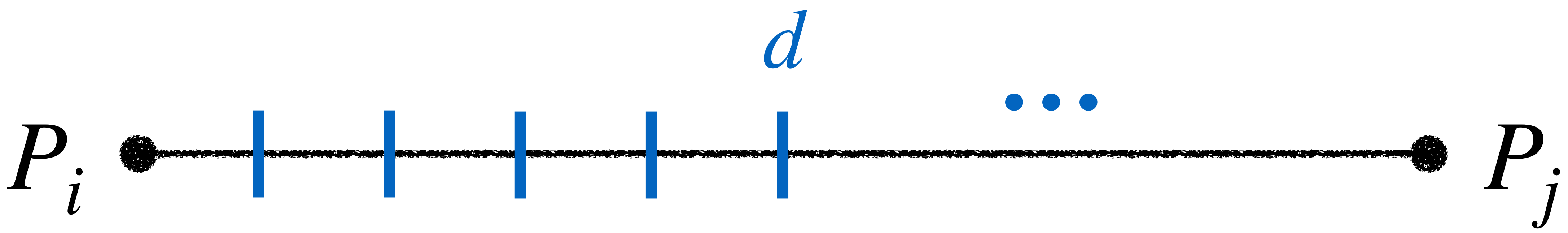


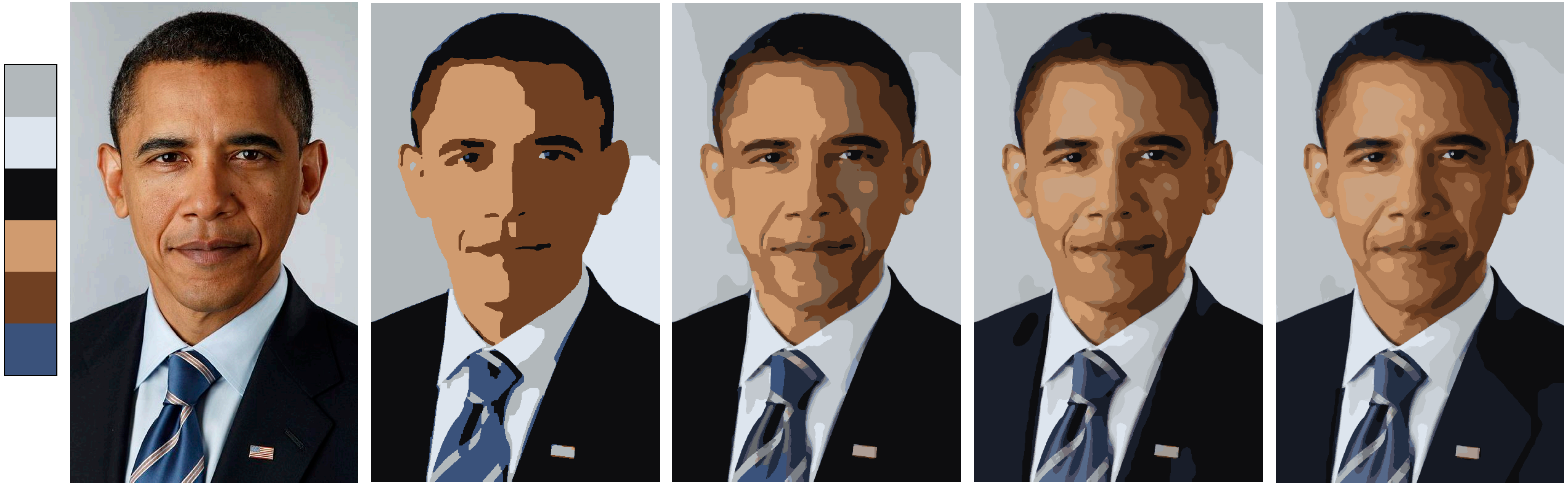


Input $d = 0$ $d = 1$ $d = 2$

Step 2: Rough region and color assignment

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

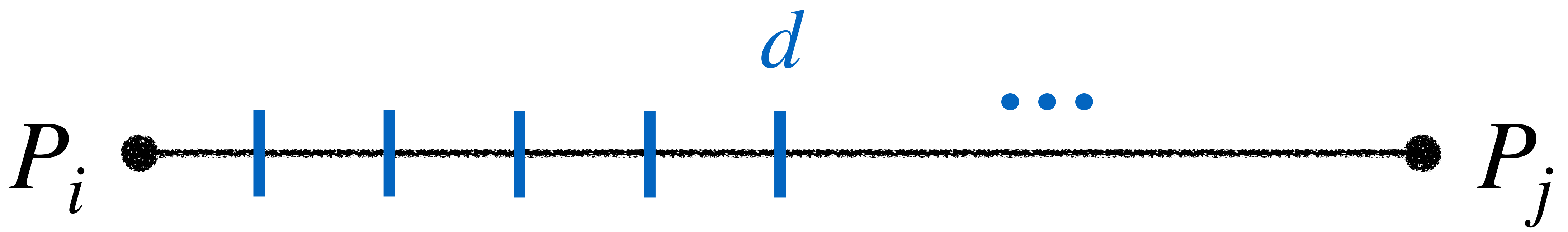


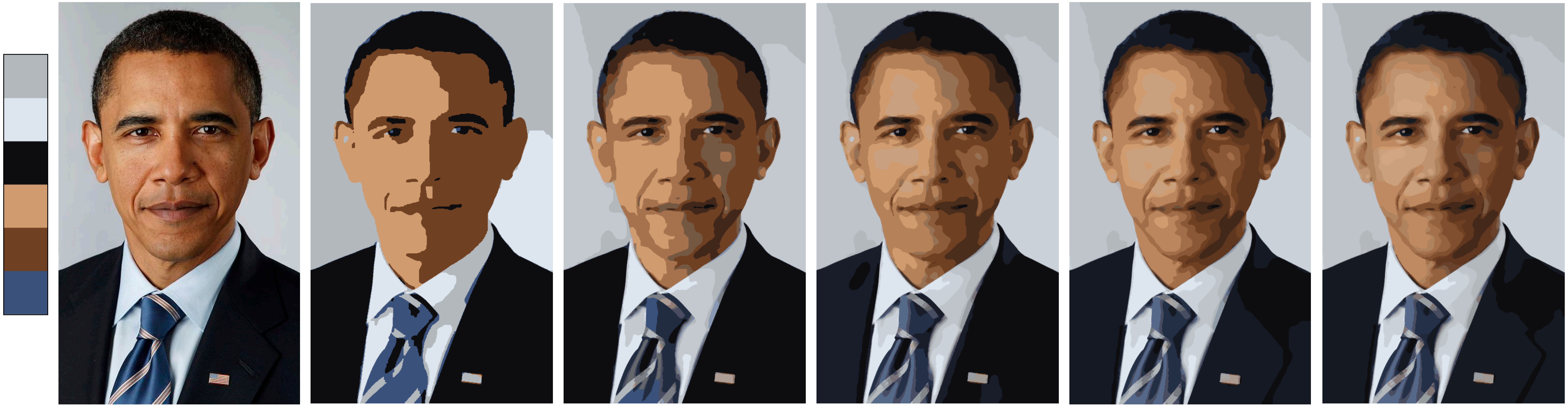


Input $d = 0$ $d = 1$ $d = 2$ $d = 3$

Step 2: Rough region and color assignment

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

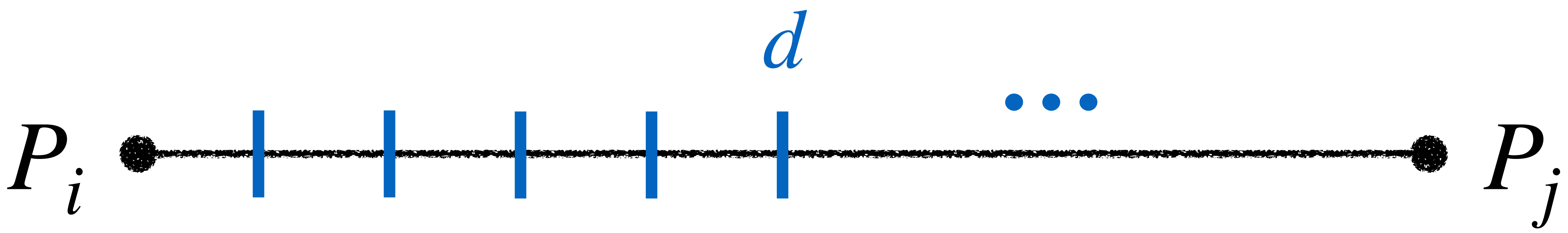


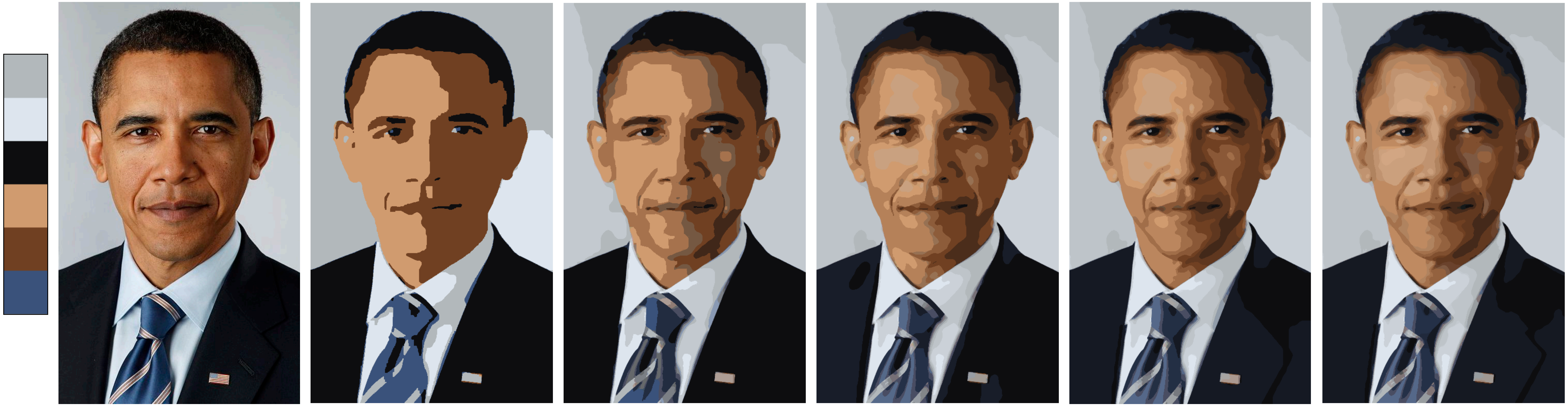


Input $d = 0$ $d = 1$ $d = 2$ $d = 3$

Step 2: Rough region and color assignment

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





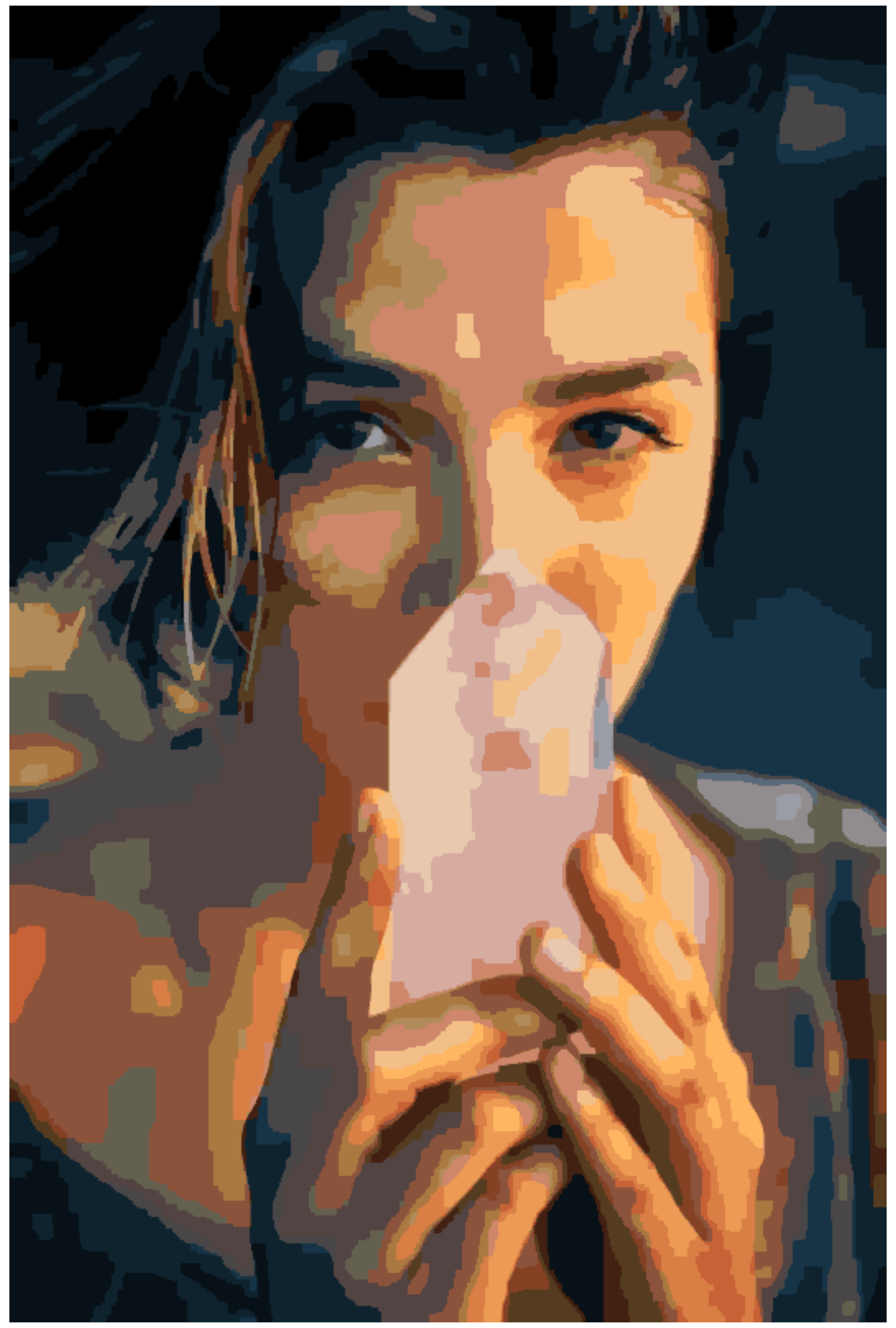
Input $d = 0$ $d = 1$ $d = 2$ $d = 3$ $d = 4$

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.

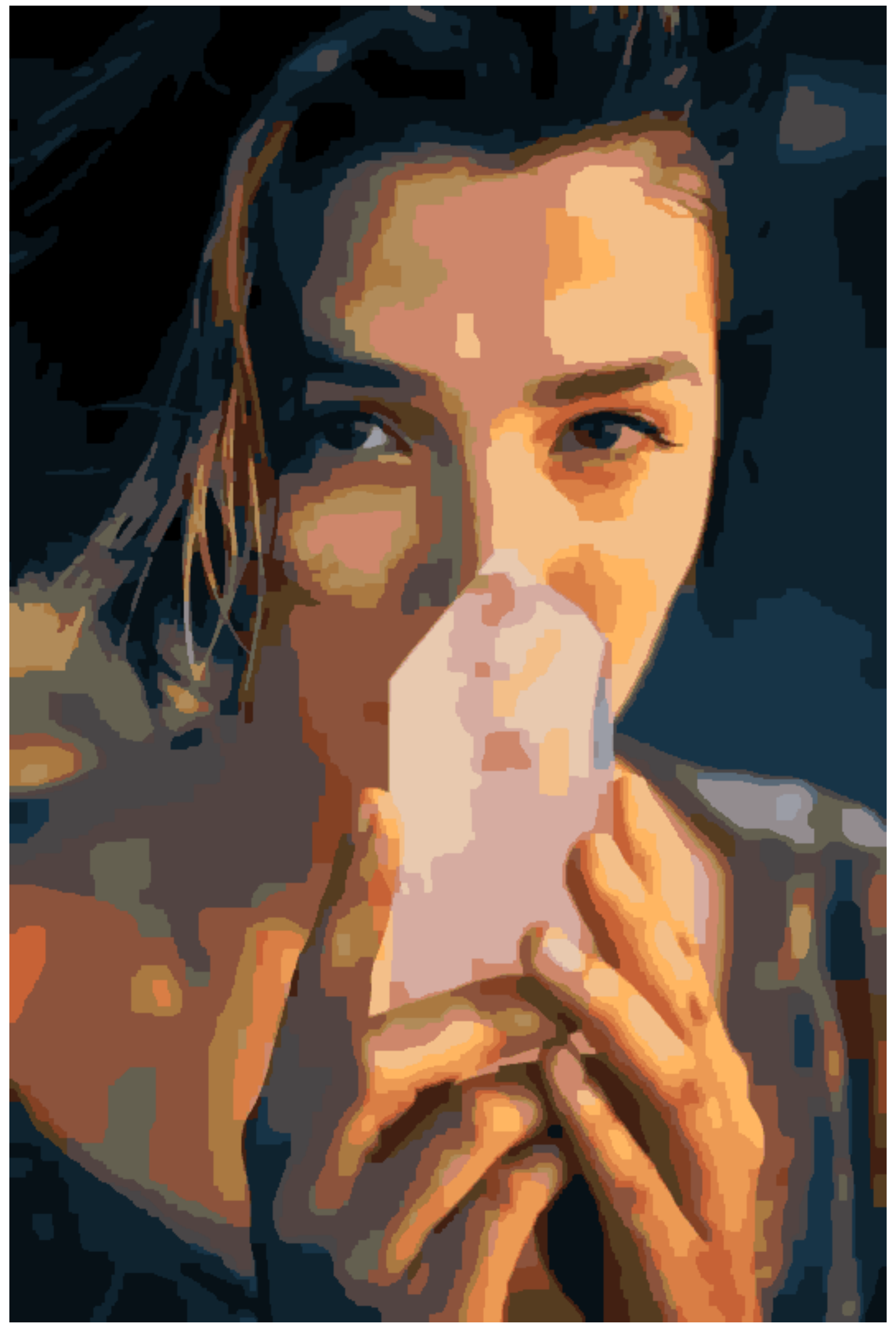
Step 3&4: Region Refinement

- 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.



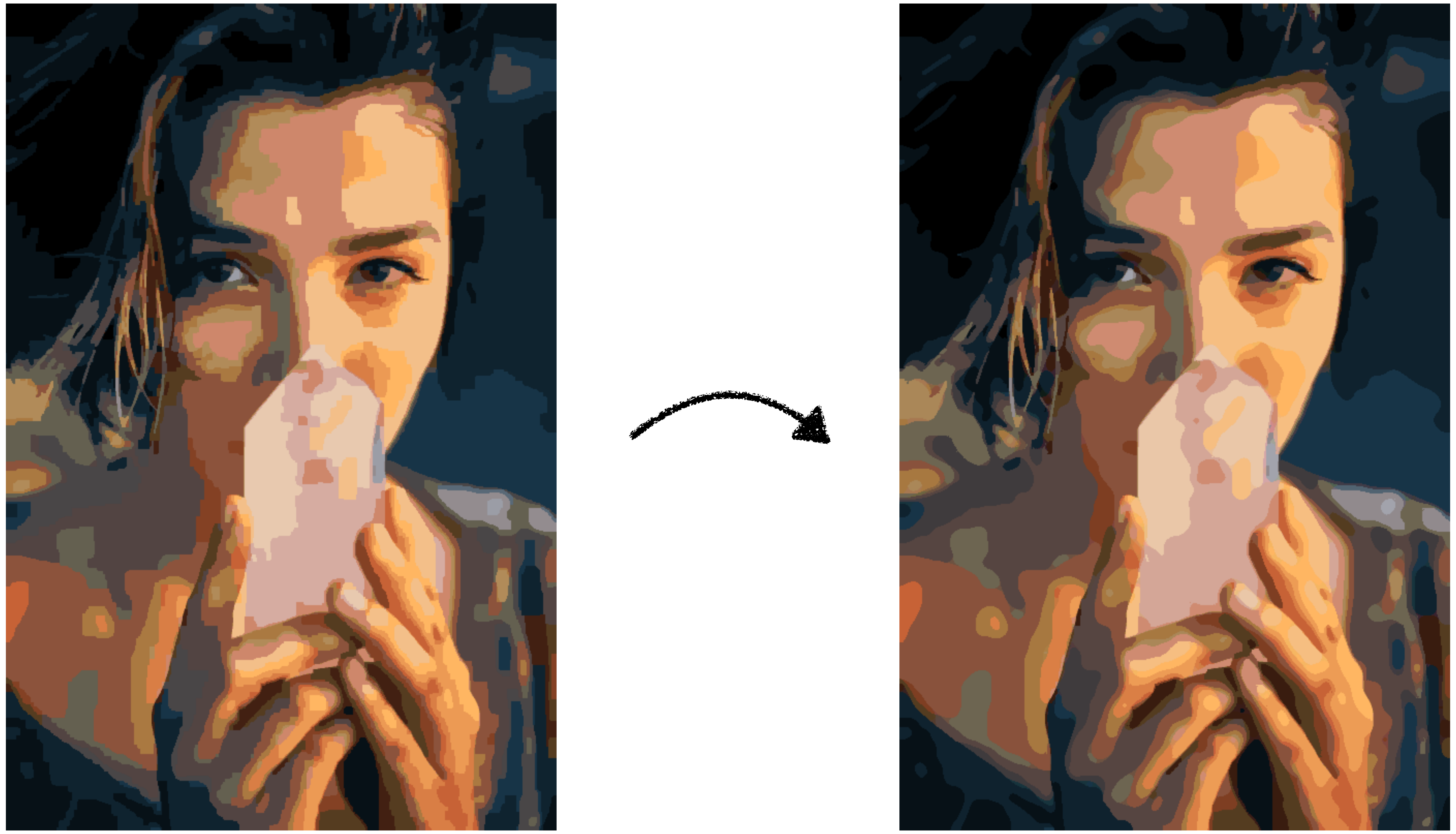
Step 3&4: Region Refinement

- 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.



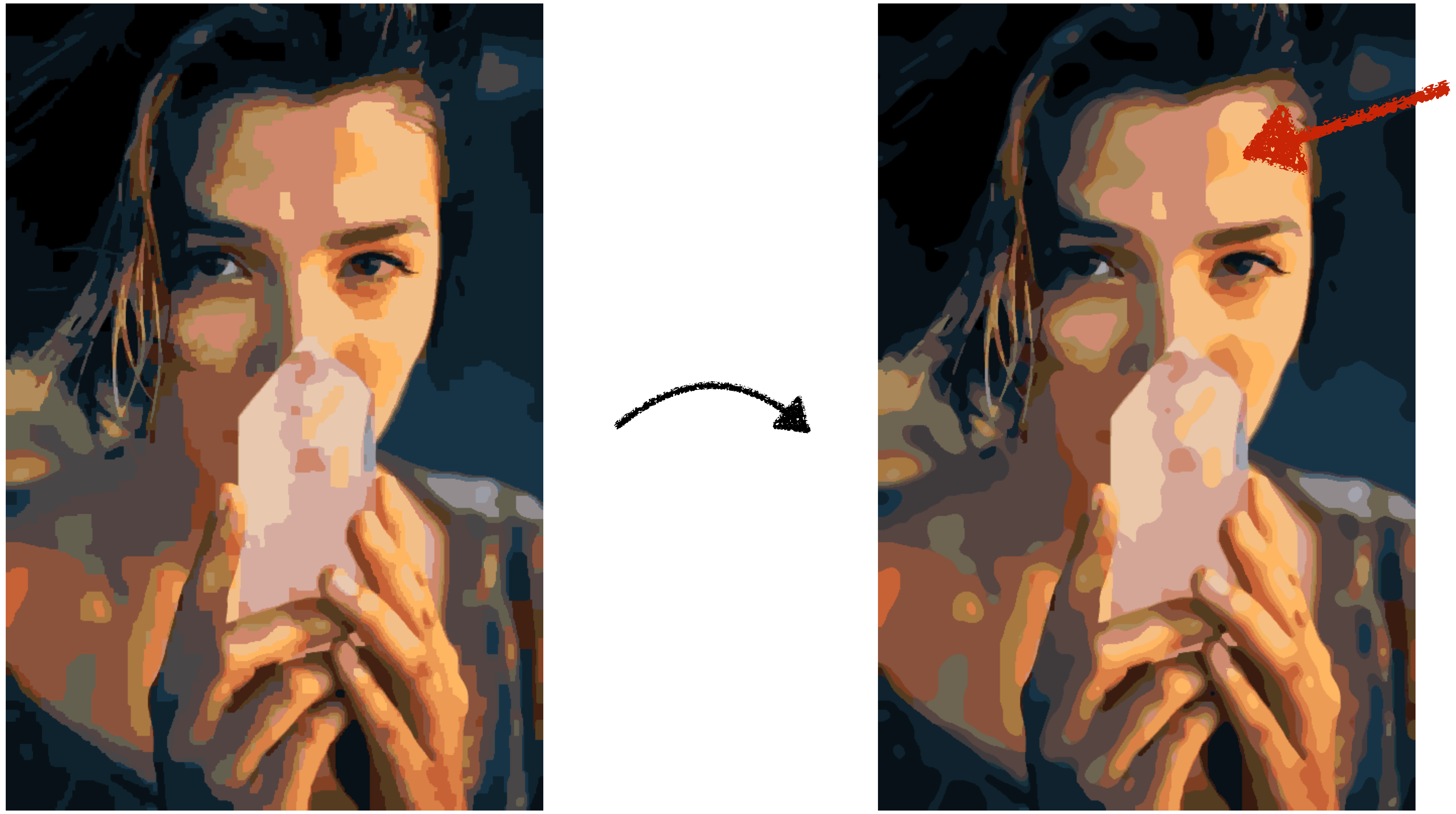
Step 3&4: Region Refinement

- 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.



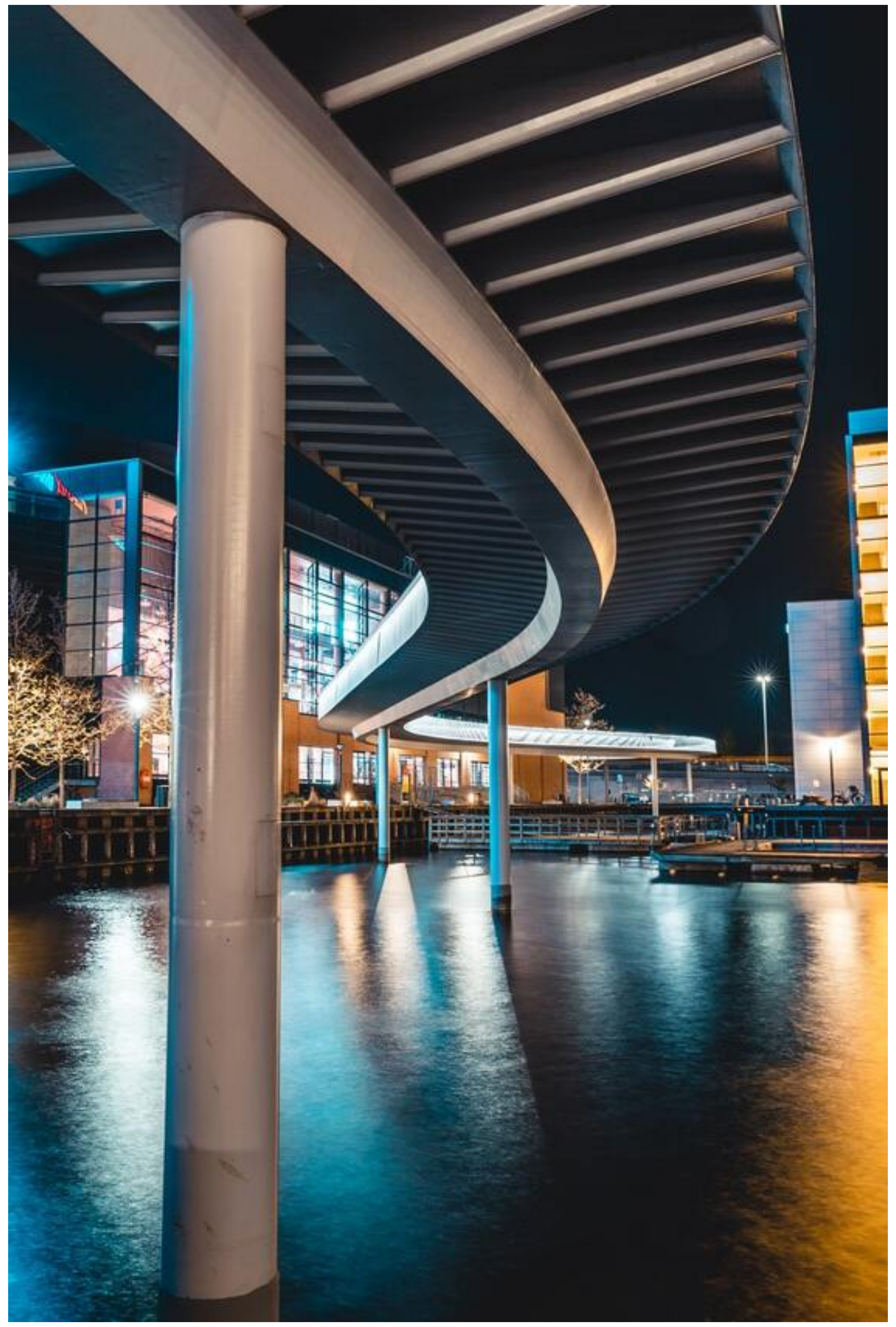
Step 3&4: Region Refinement

- 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.

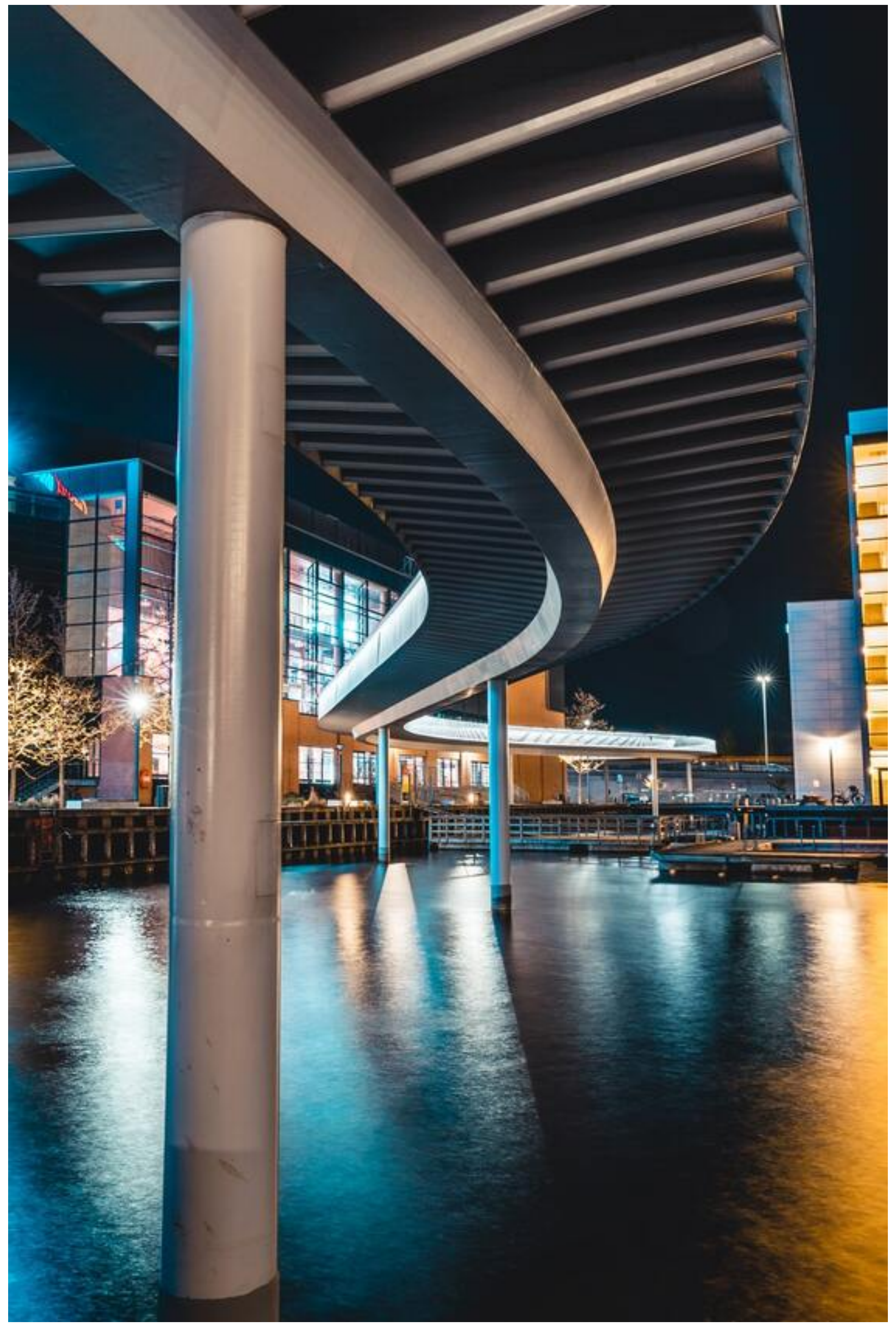


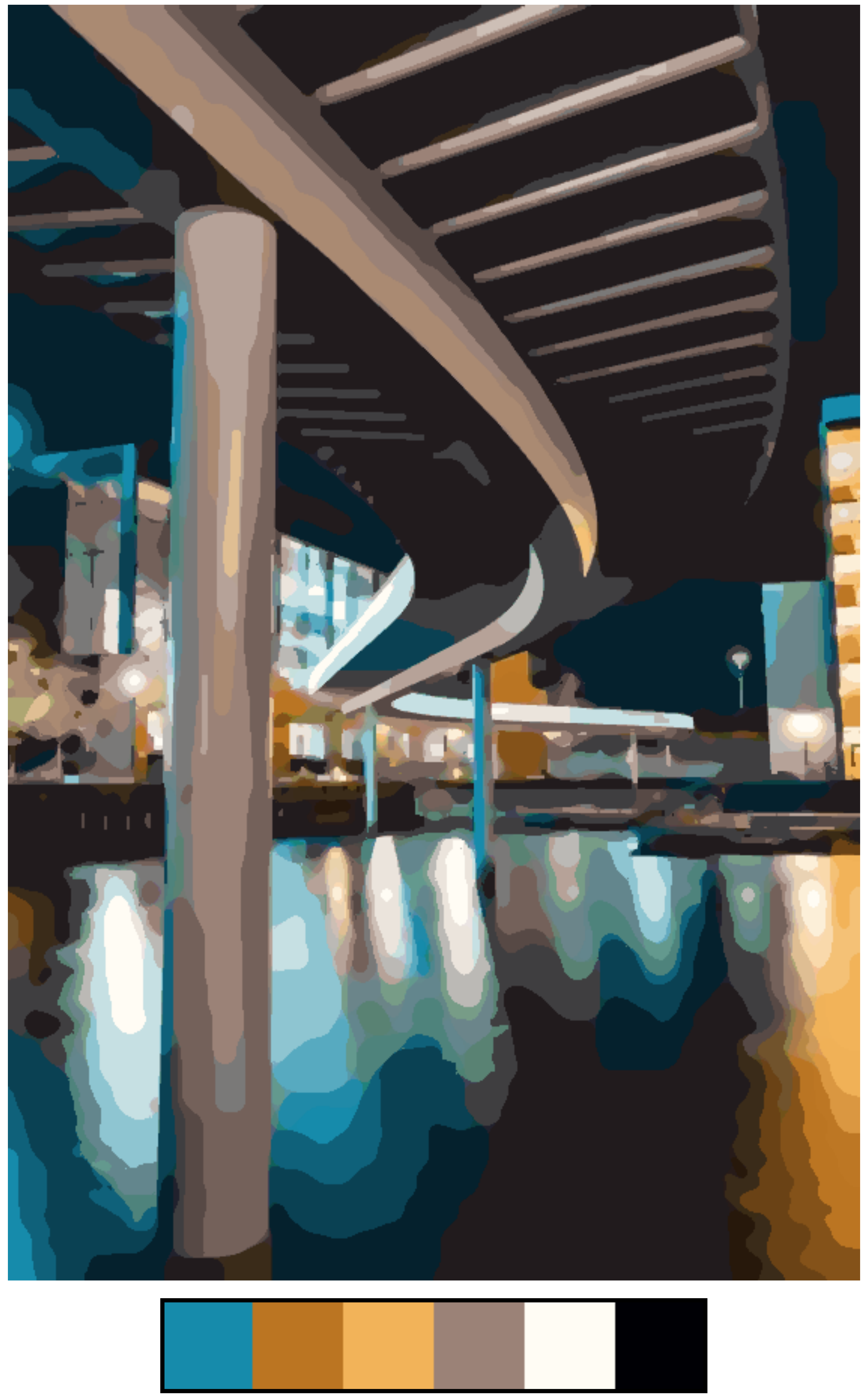
Results

Results

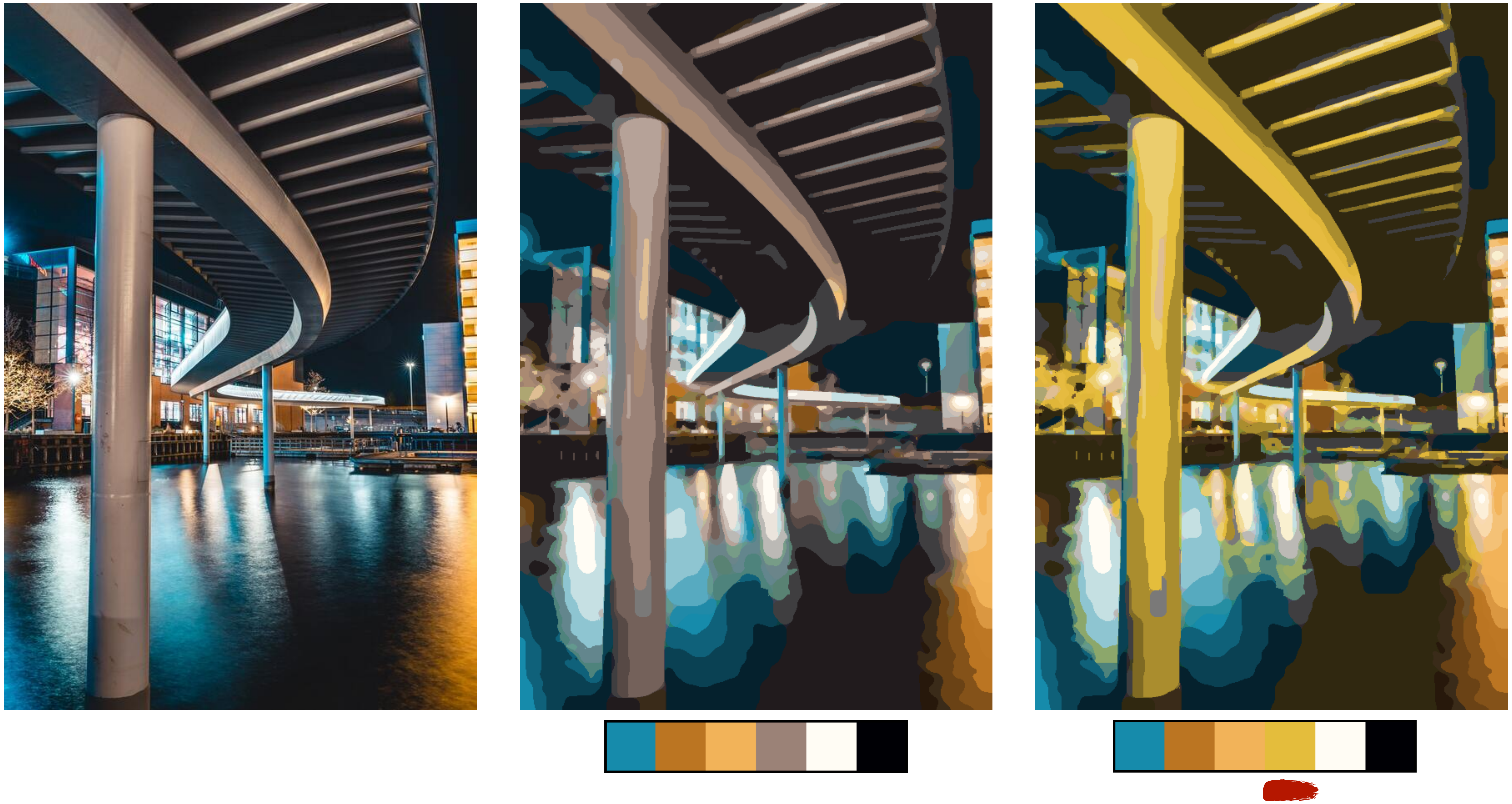


Results

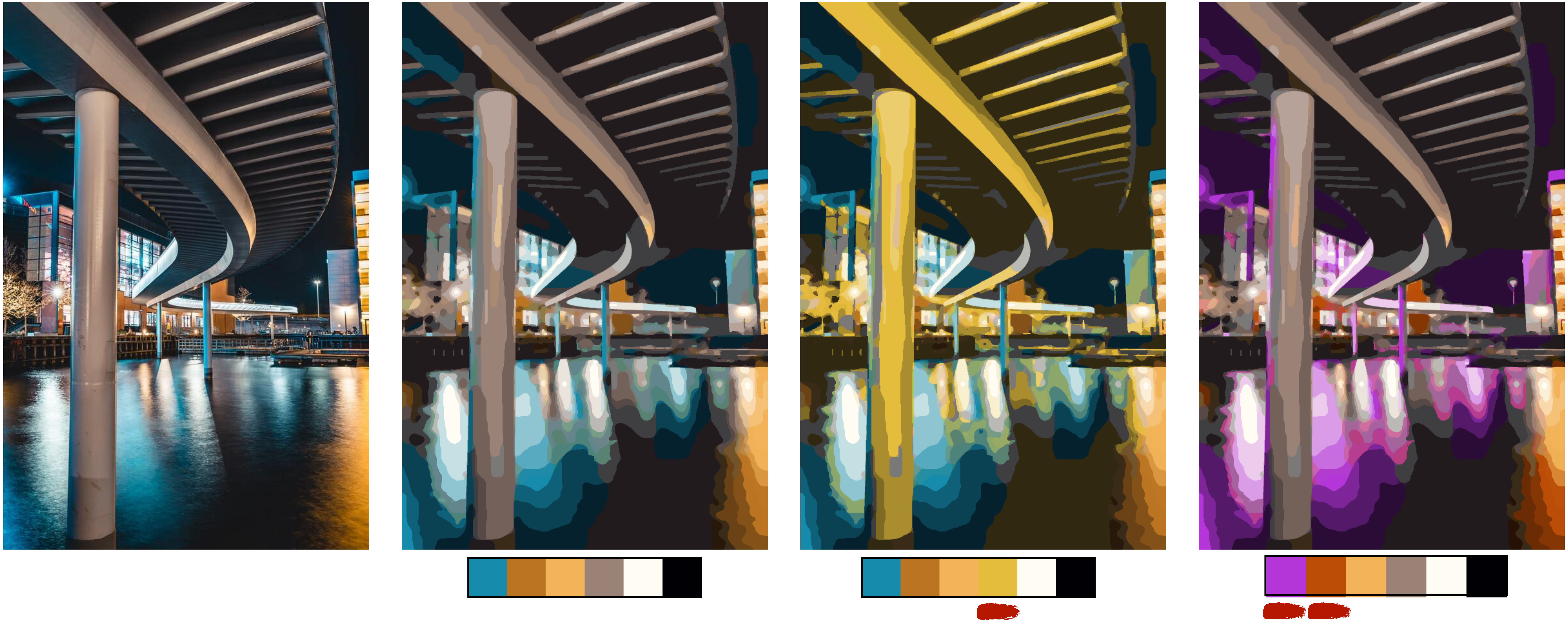


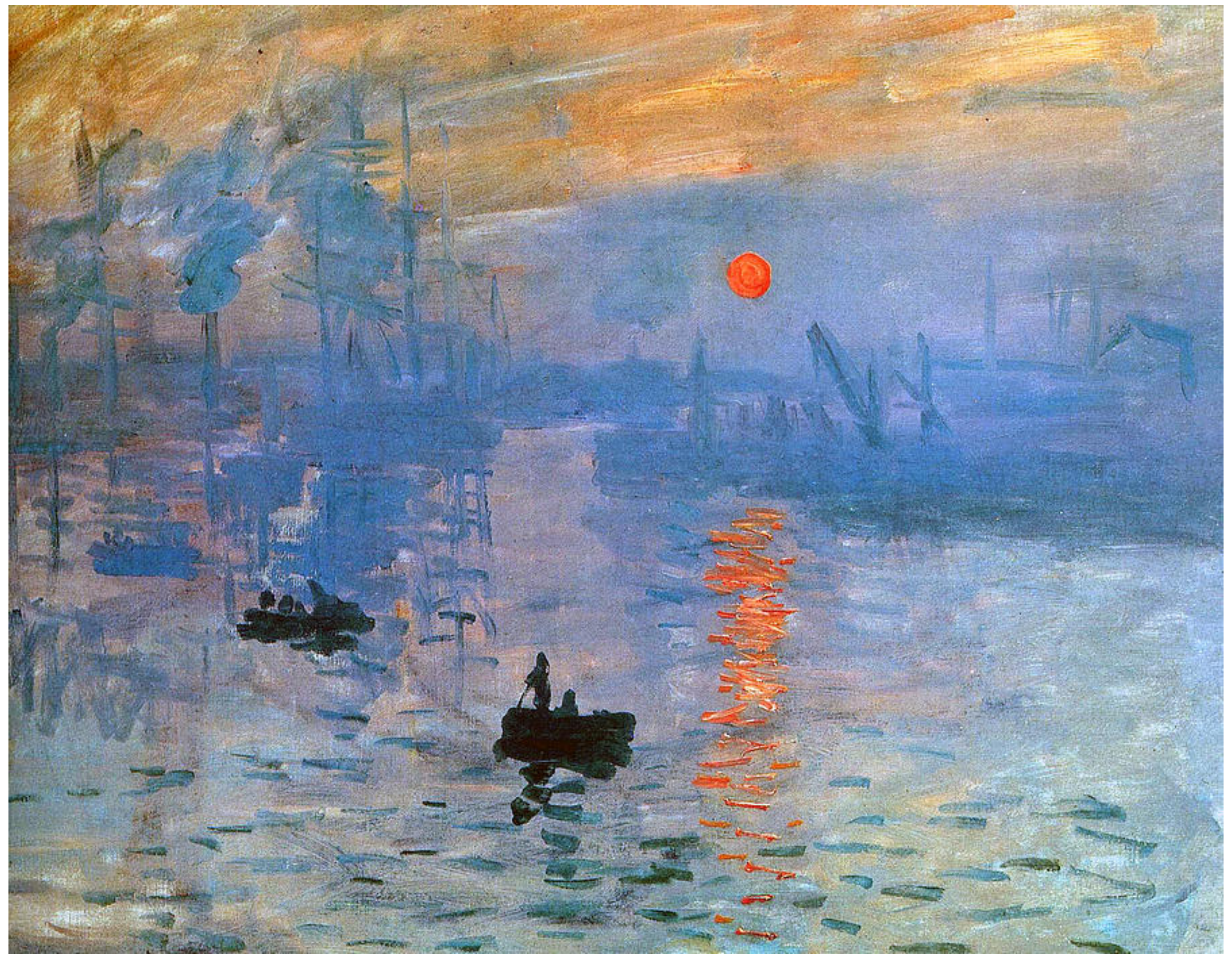


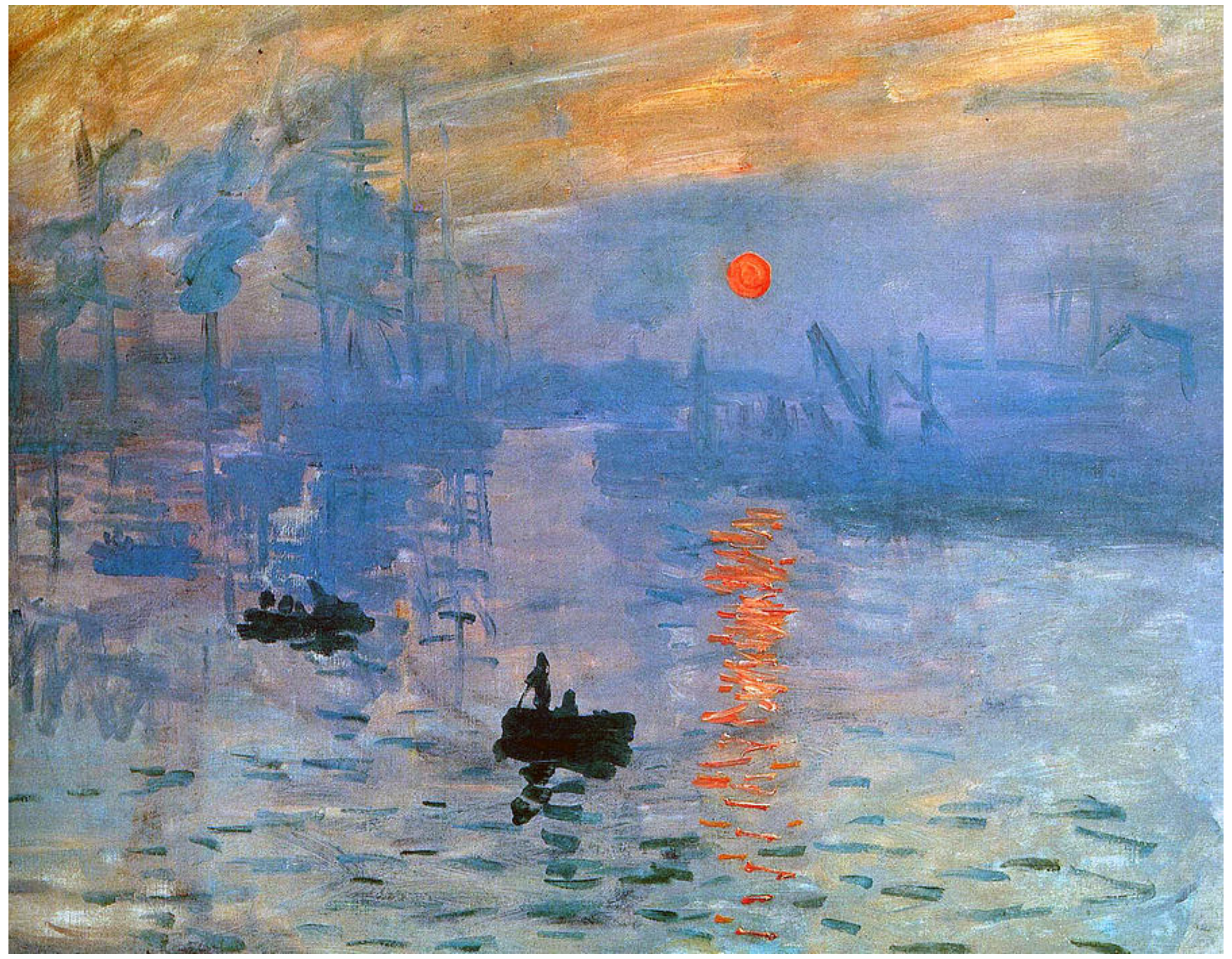
Results

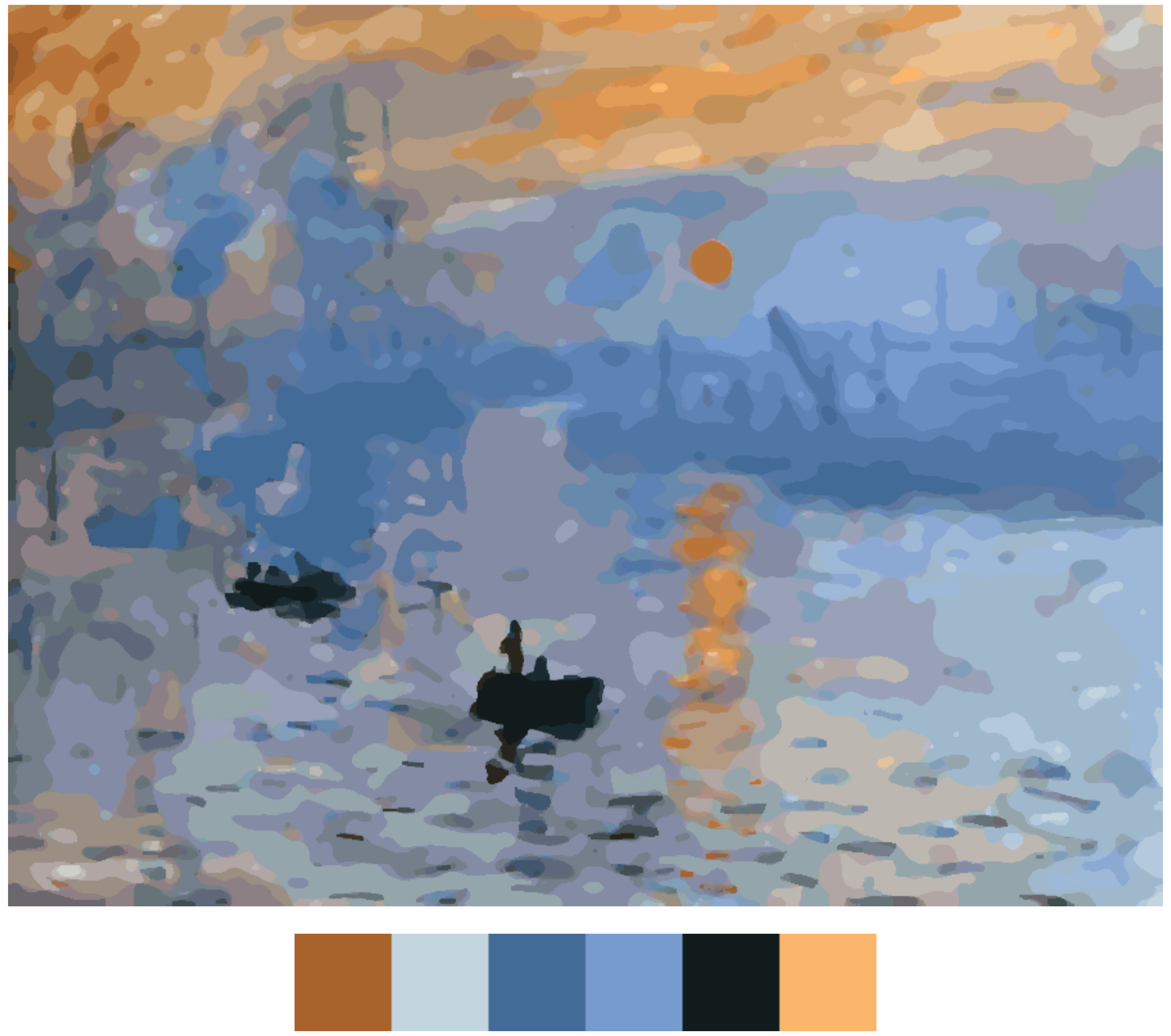


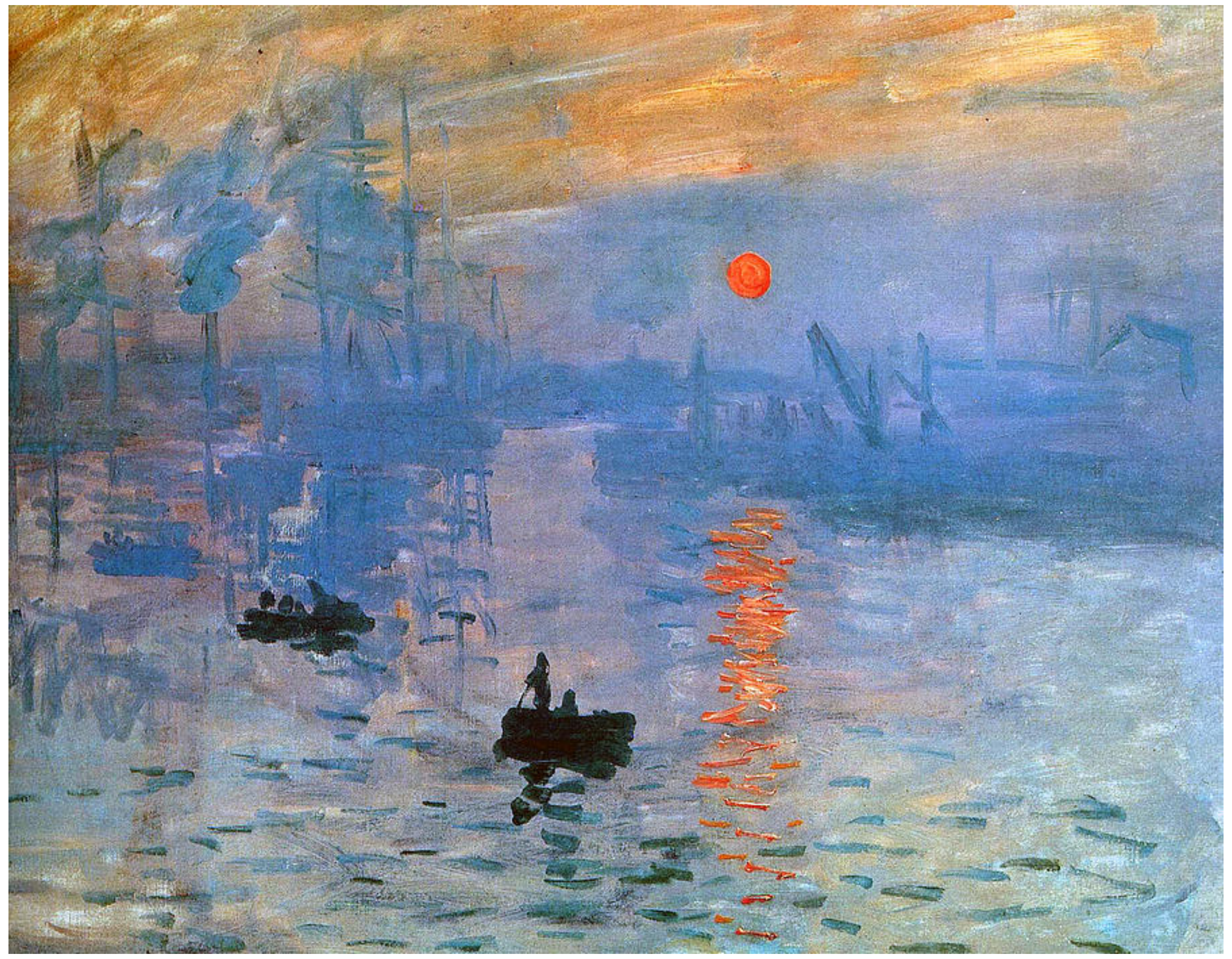
Results

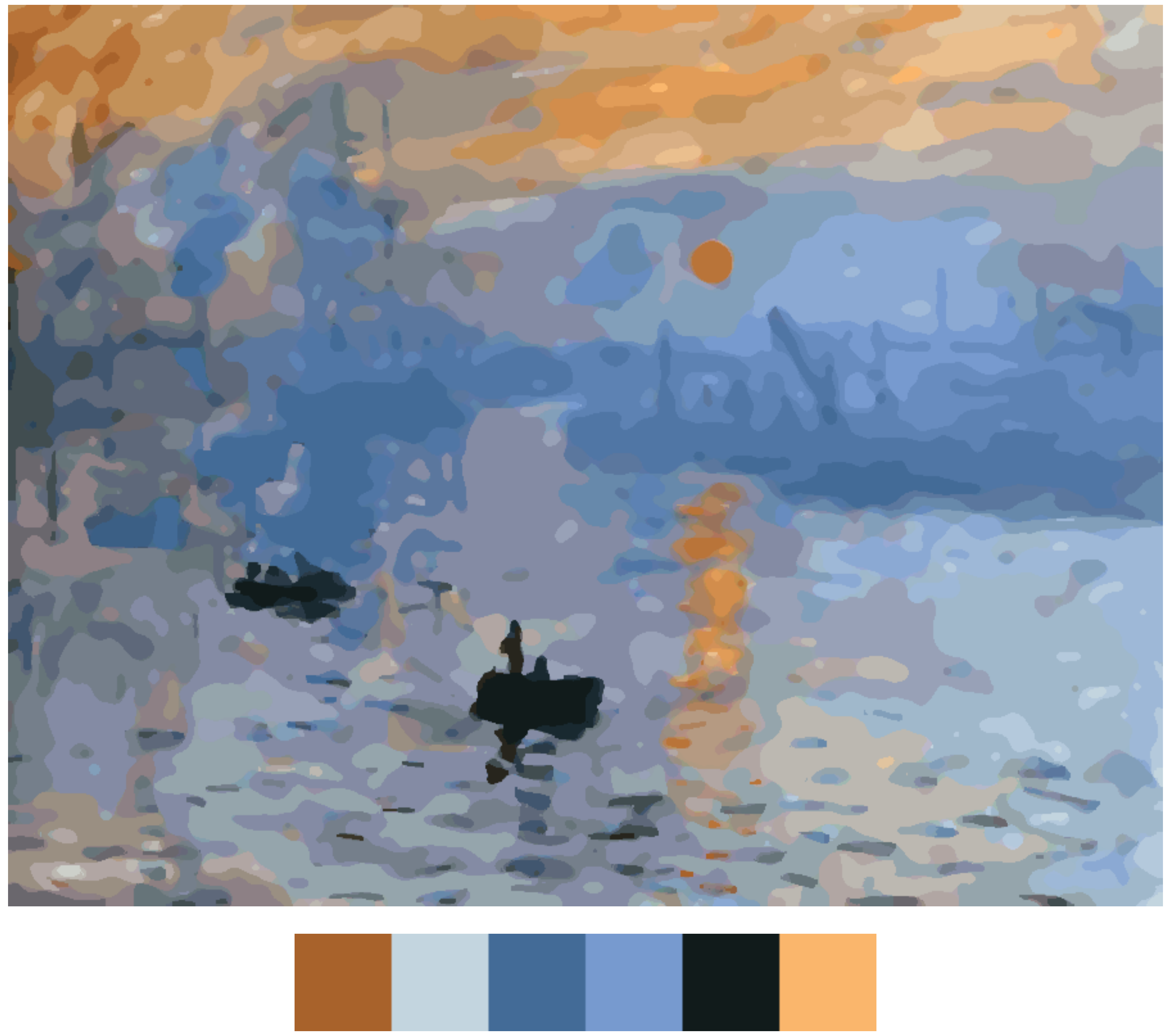


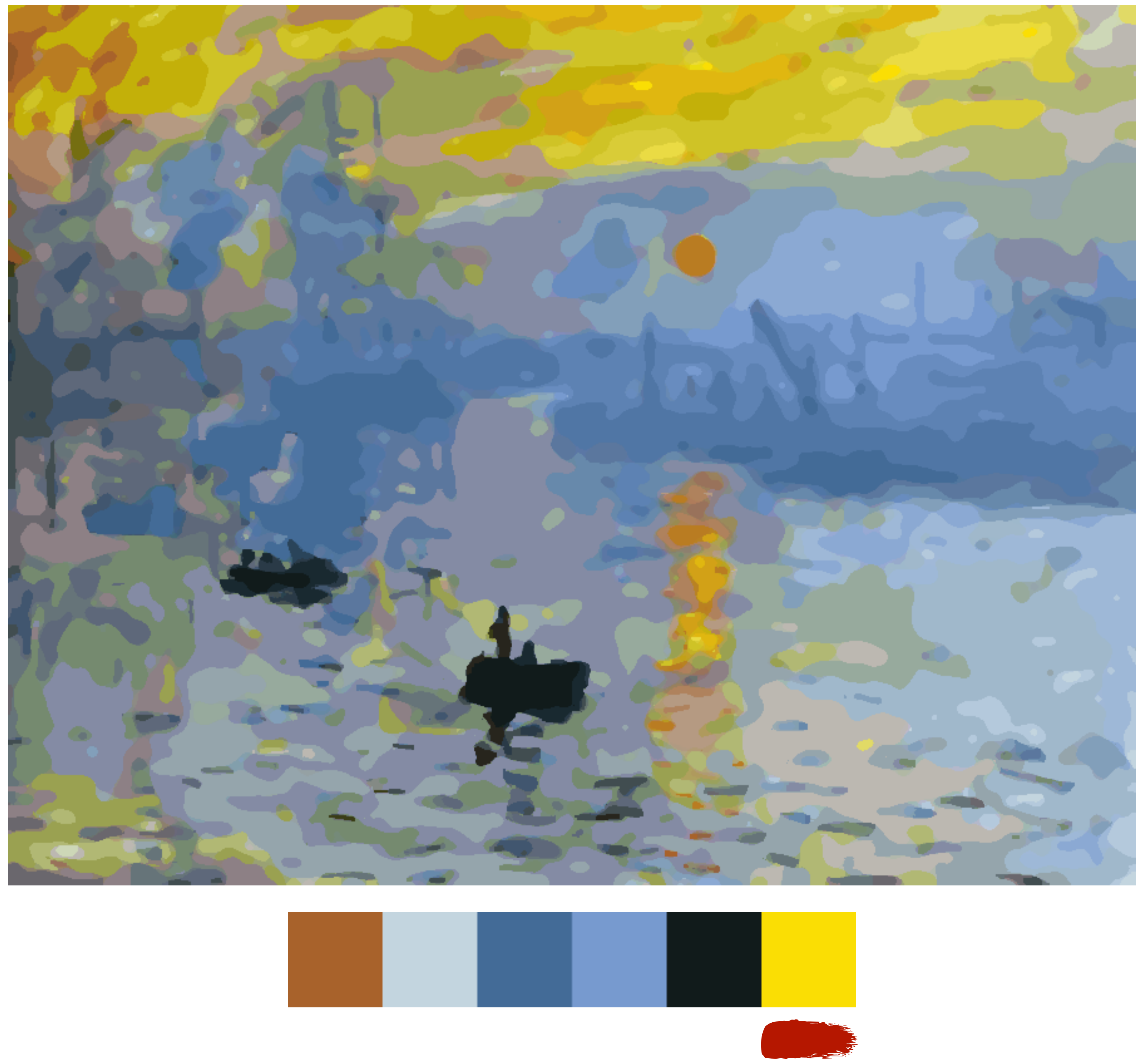


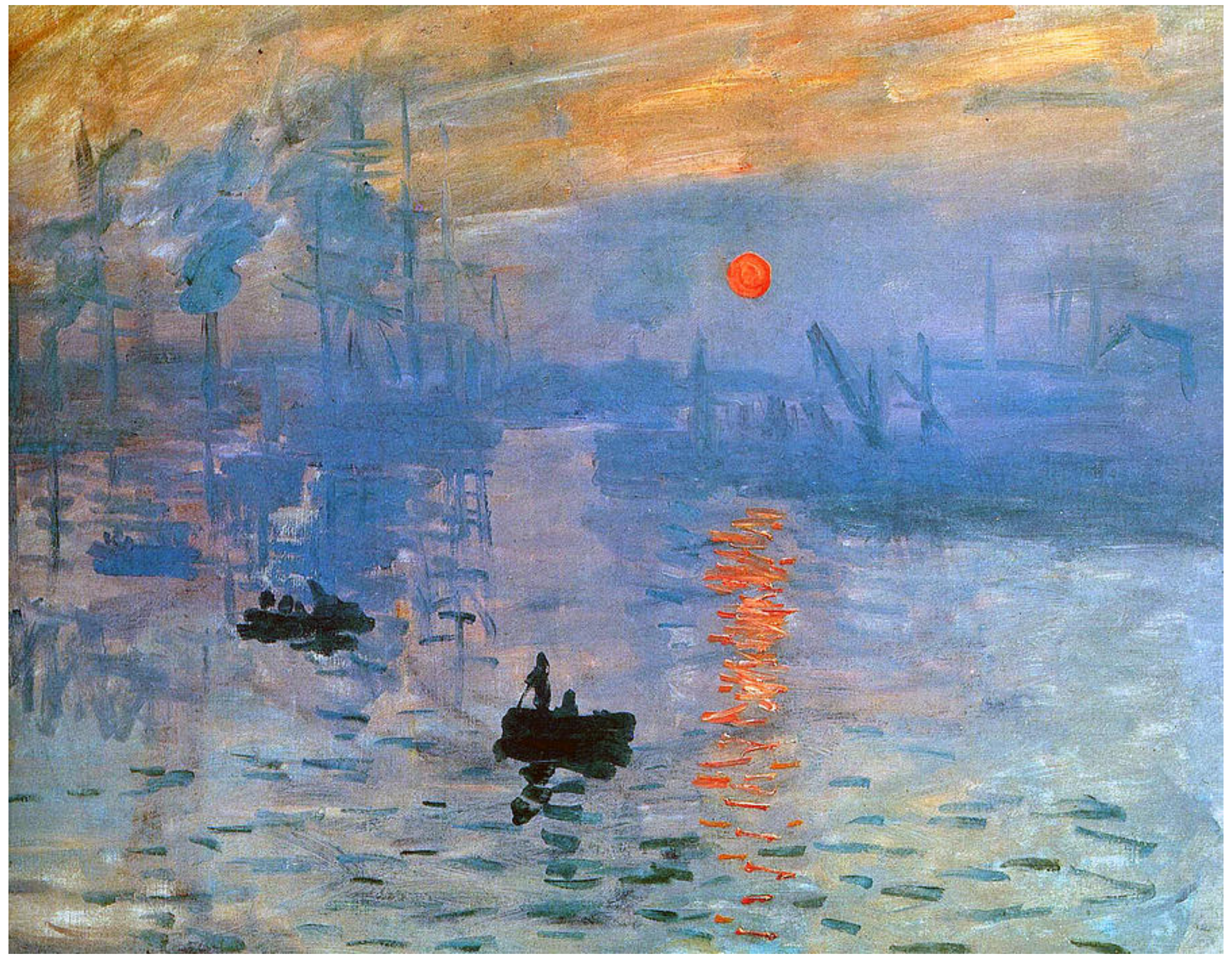


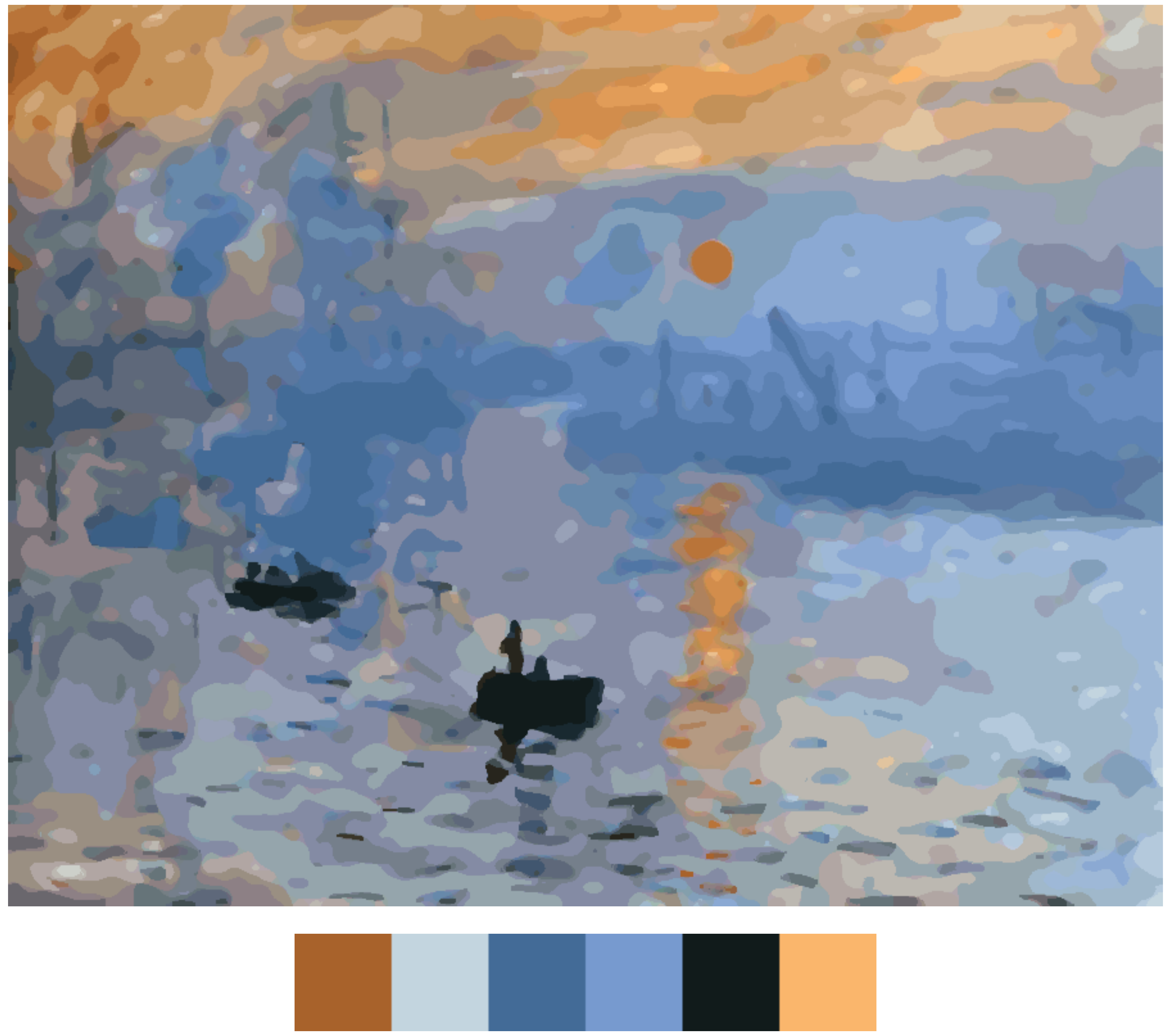


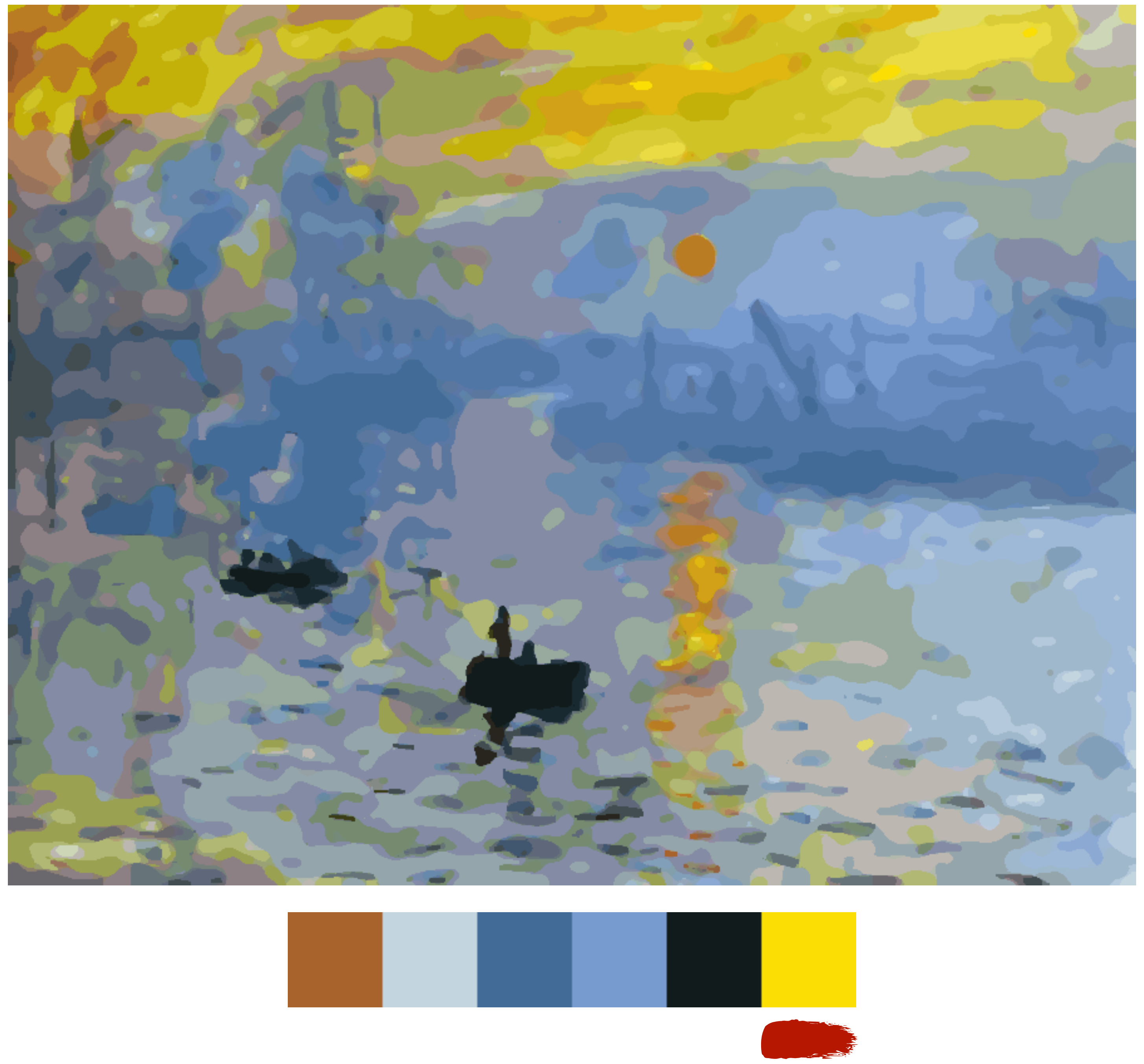


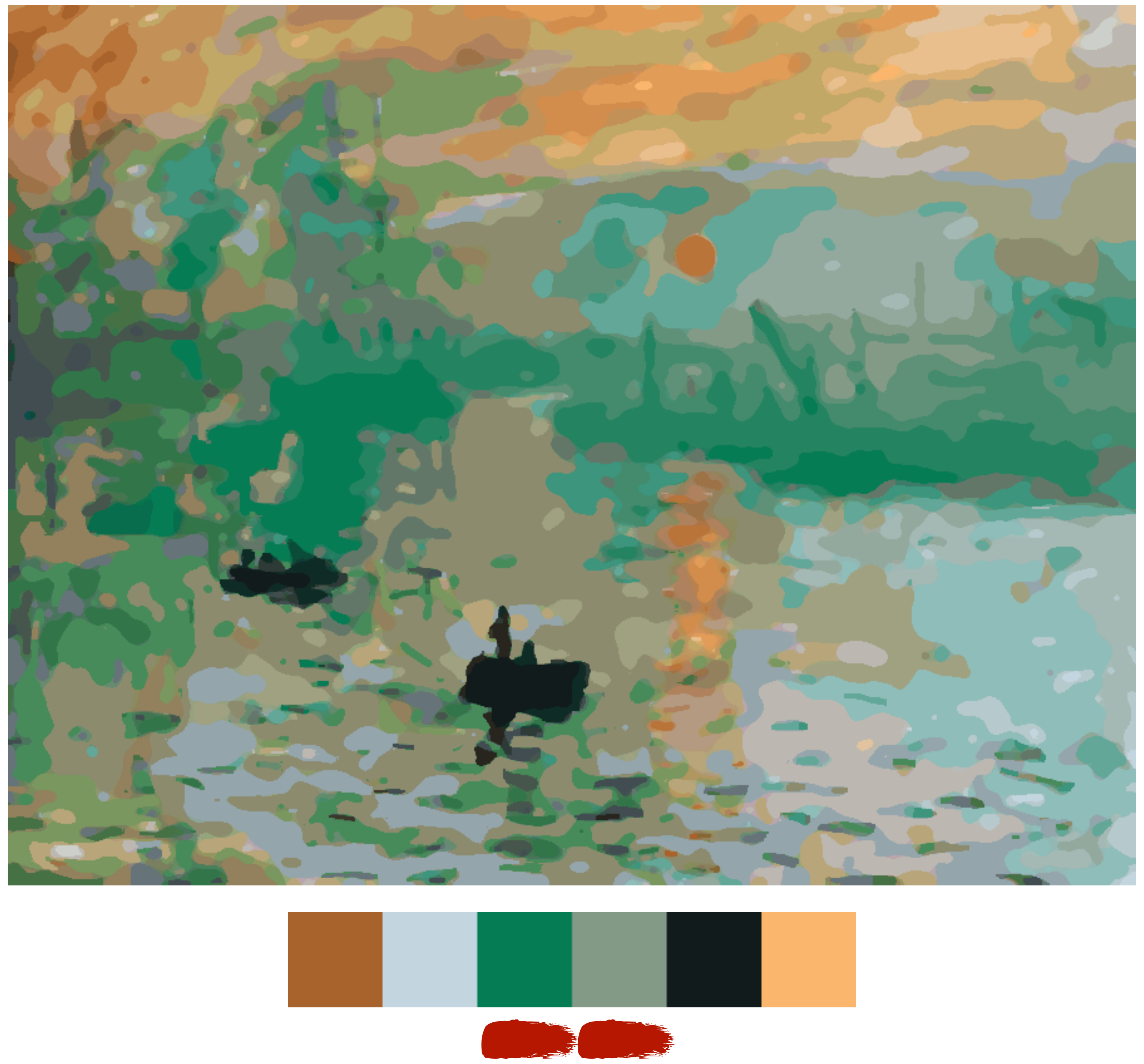












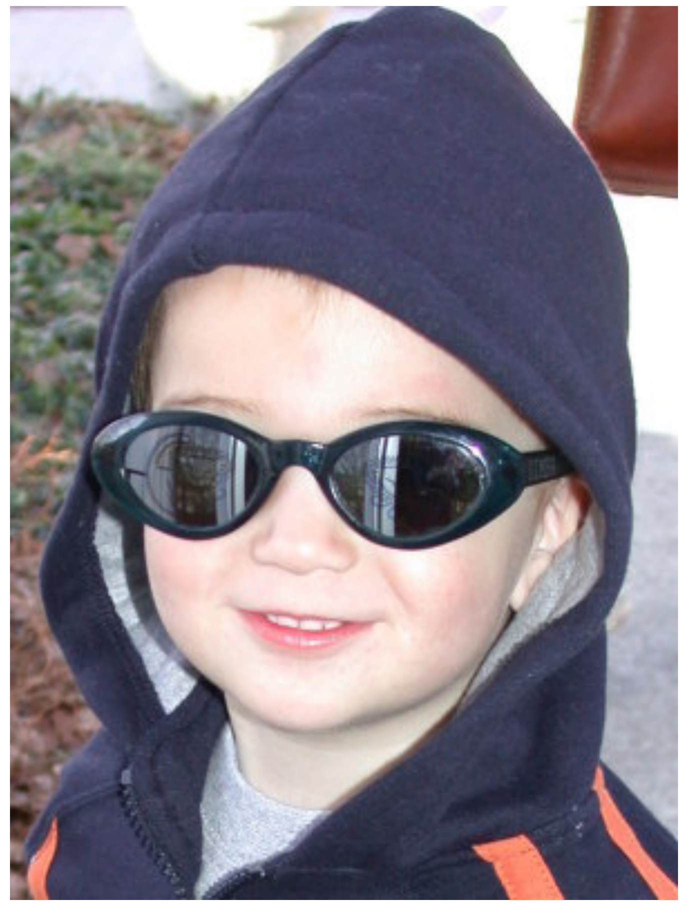
Evaluation

Evaluation

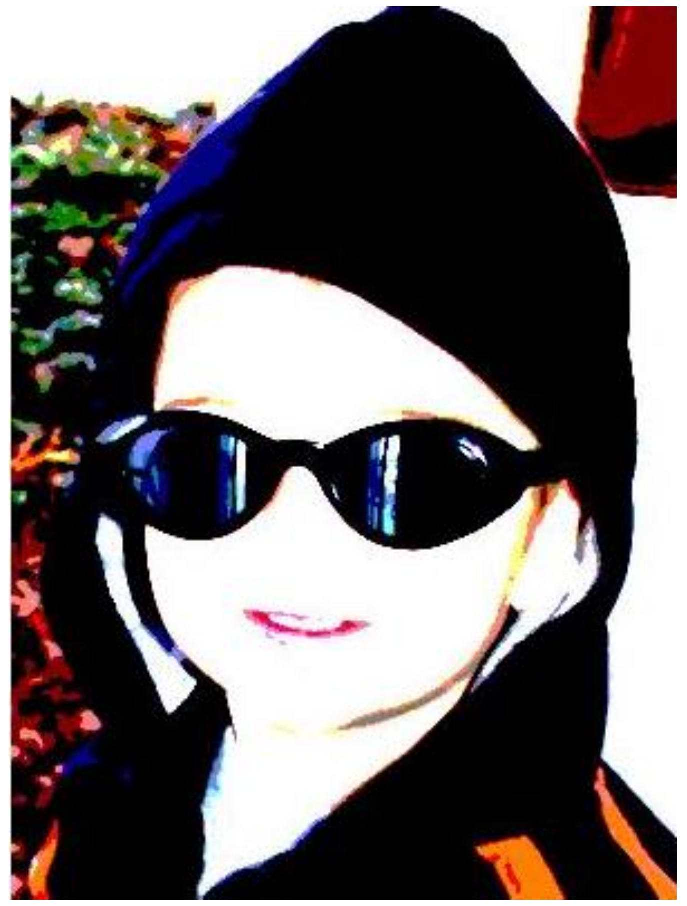
- Comparison to related approaches.

Evaluation

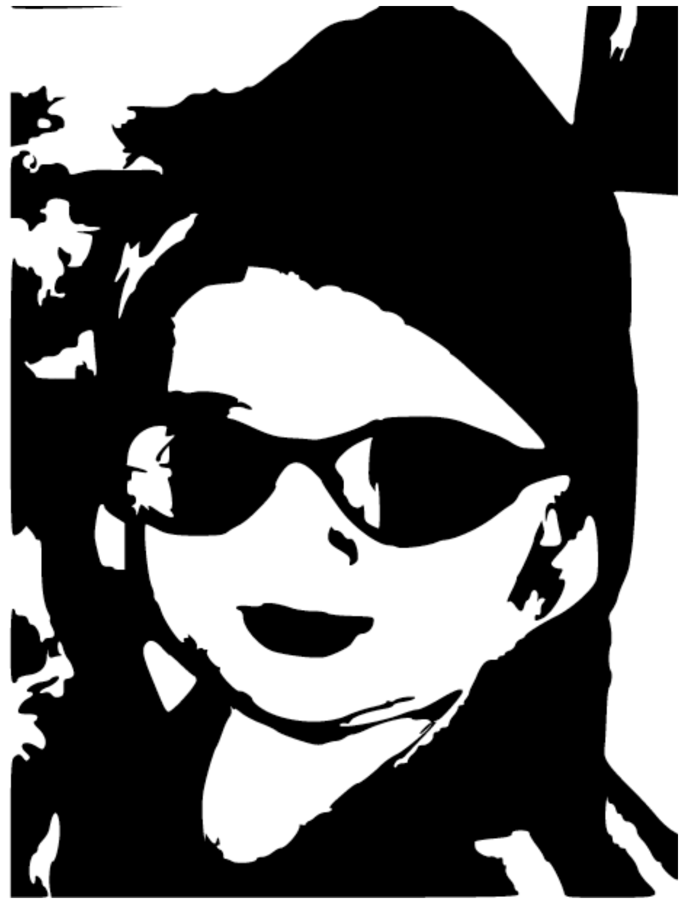
- Comparison to related approaches.



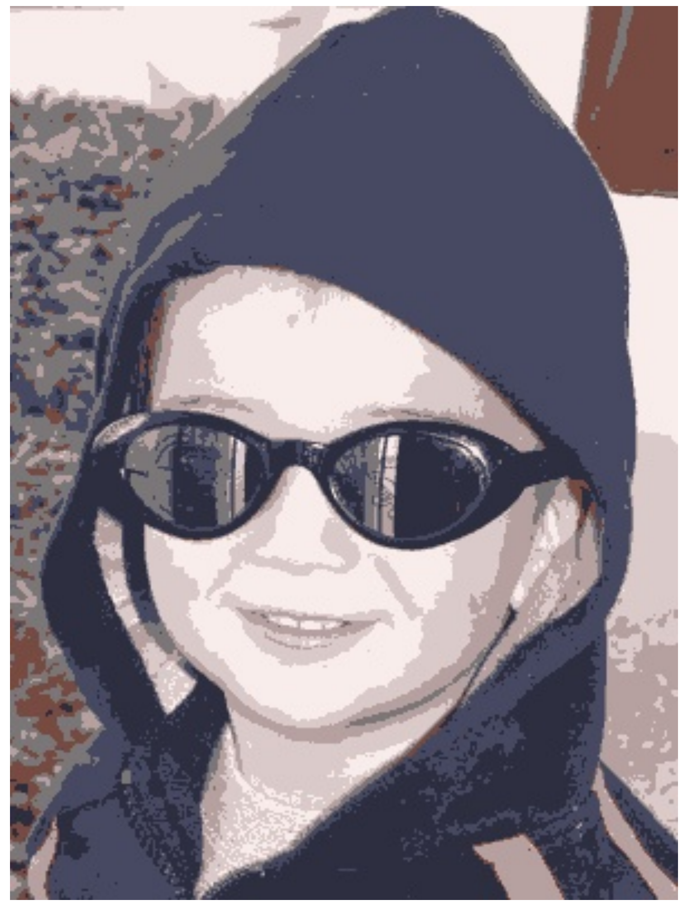
Input



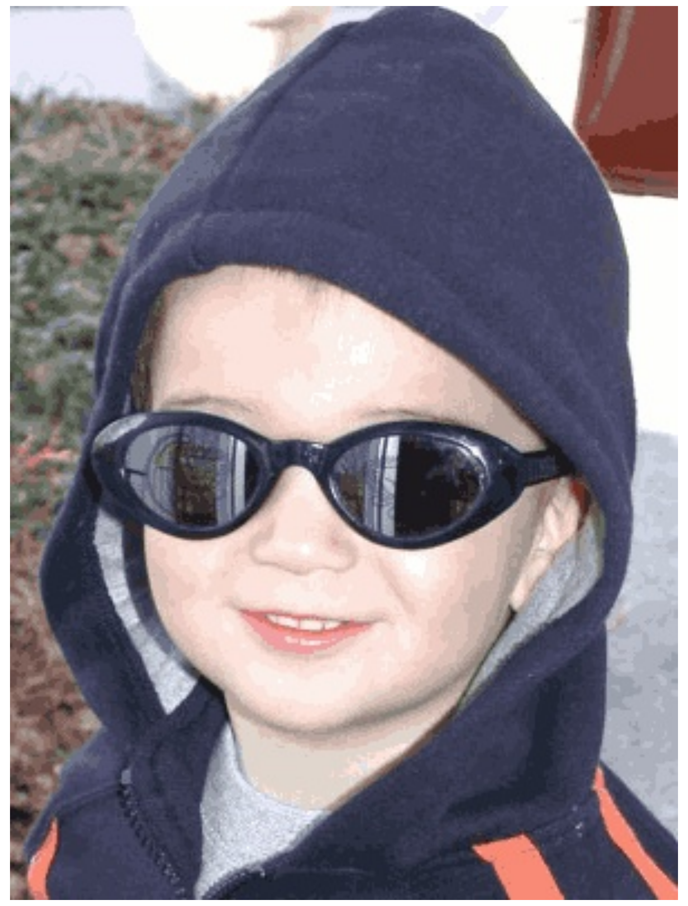
[Afifi 2018]



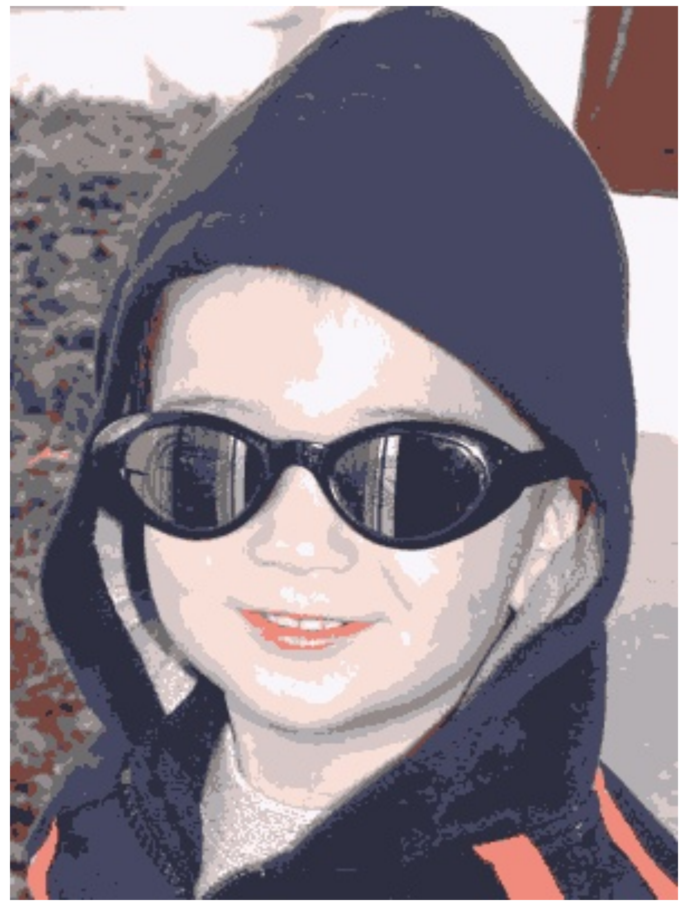
[Xu and Kaplan 2008]



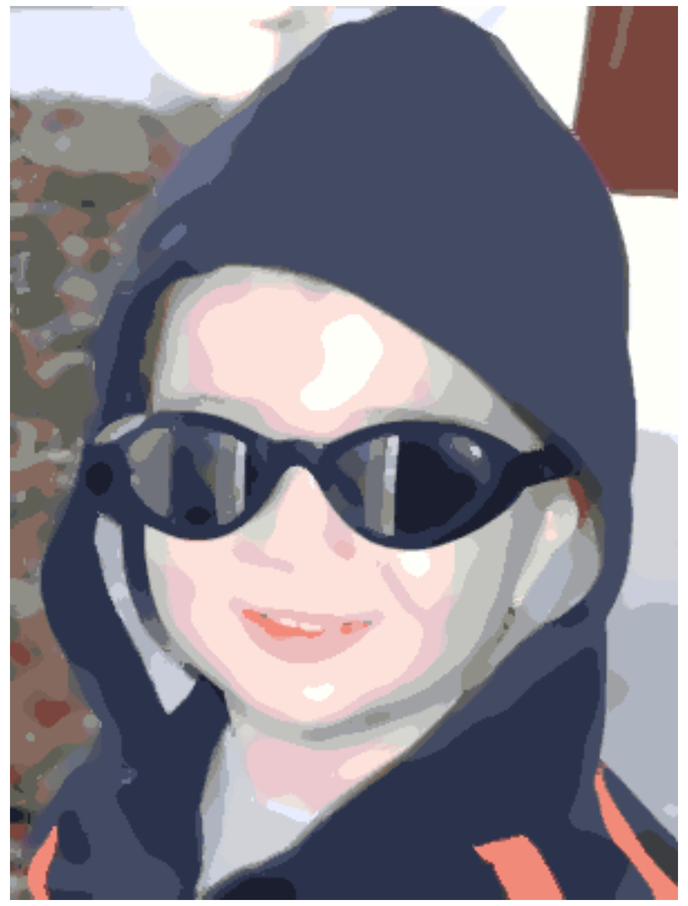
Direct K-means
(*K*=*7* in RGB-space)



Direct K-means
(*K*=*44* in RGB-space)



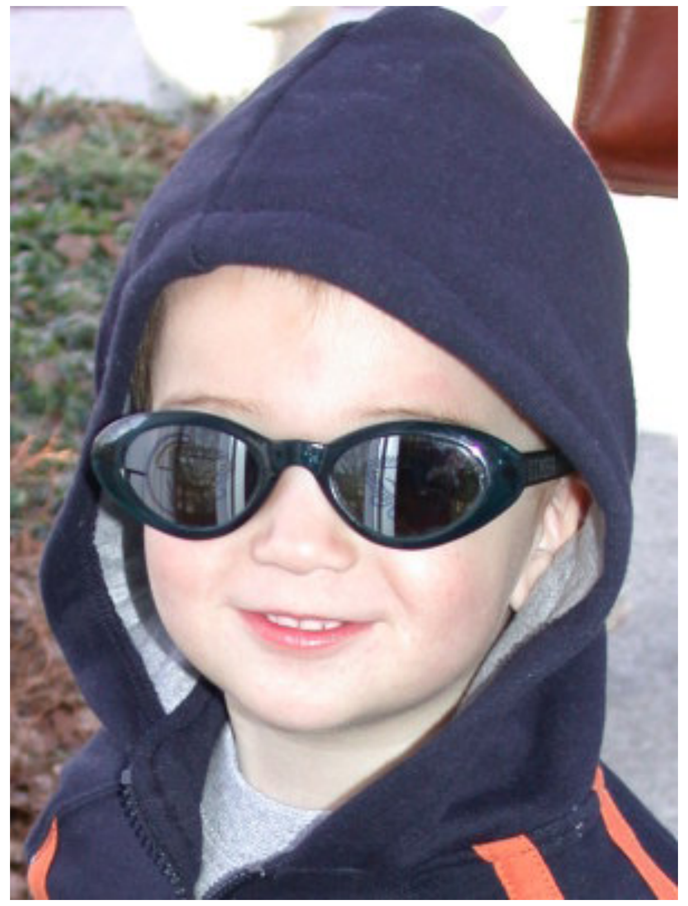
Direct K-means
(*K*=*10* in RGBXY-space)



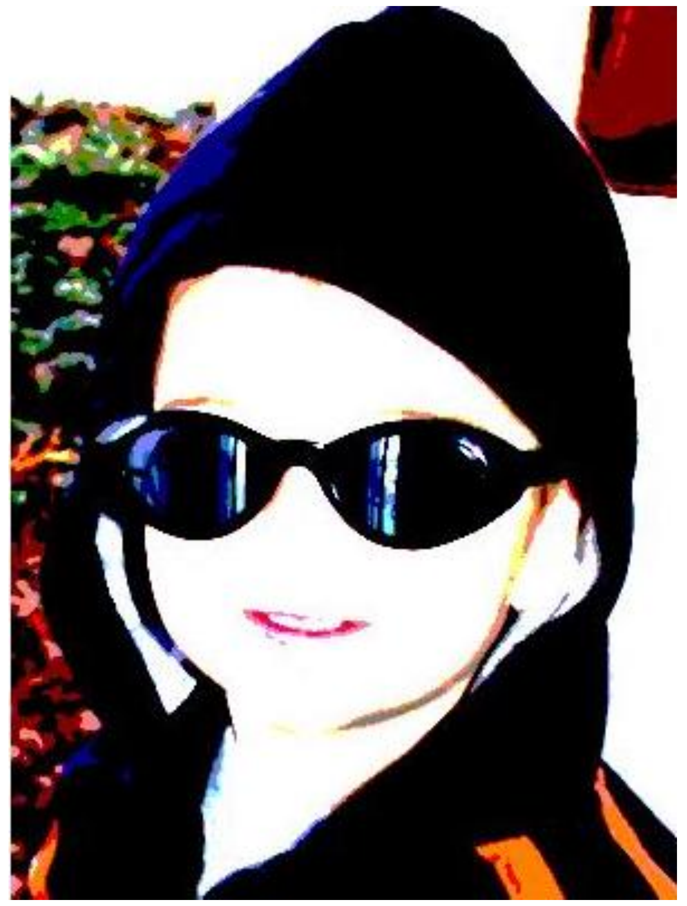
Ours

Evaluation

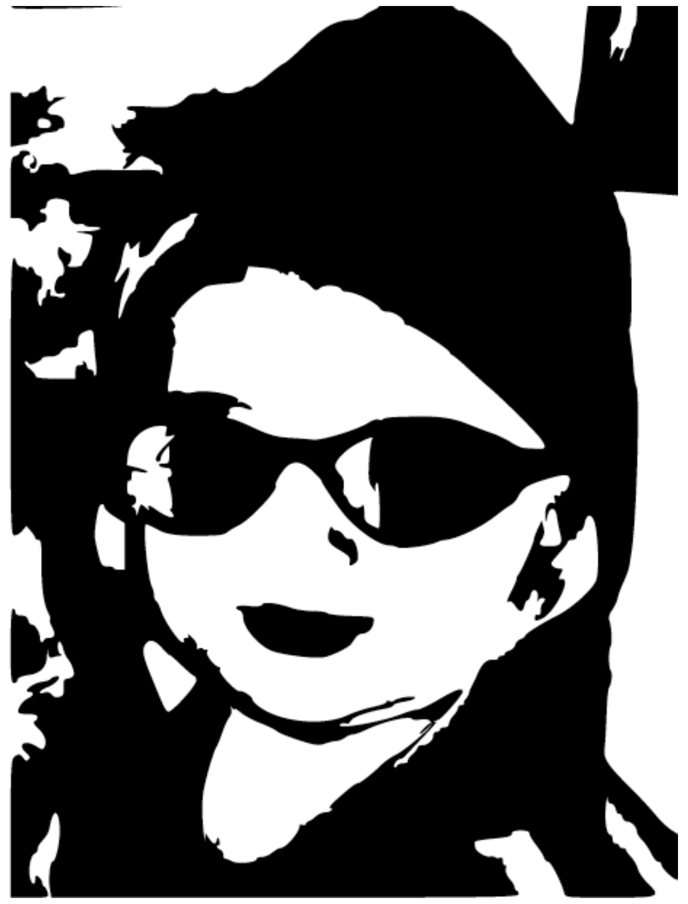
- Comparison to related approaches.



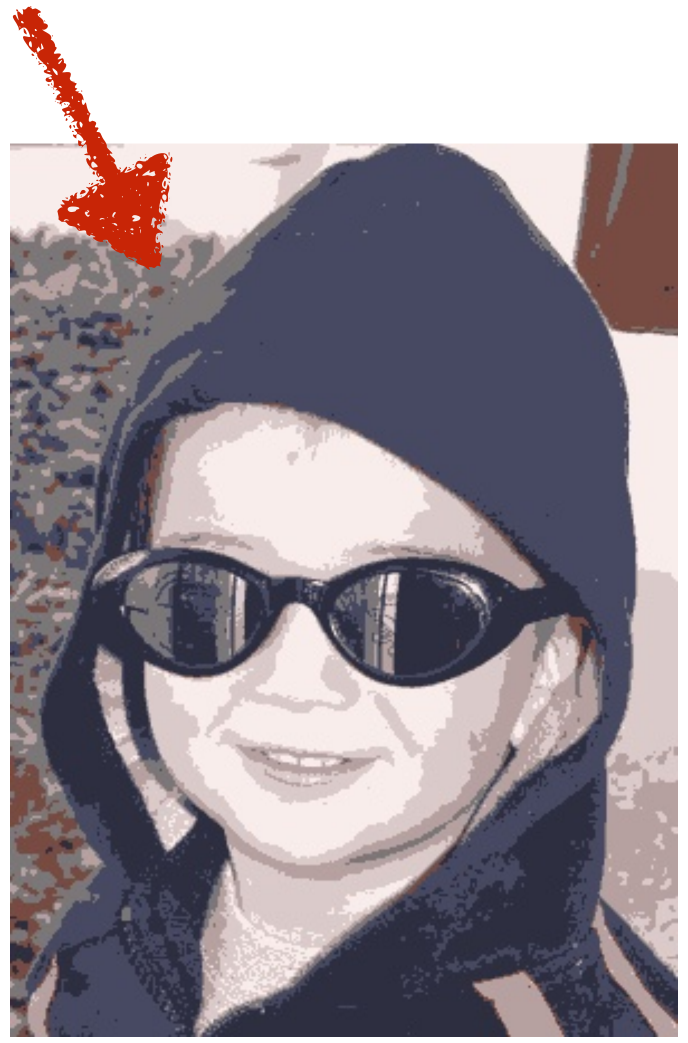
Input



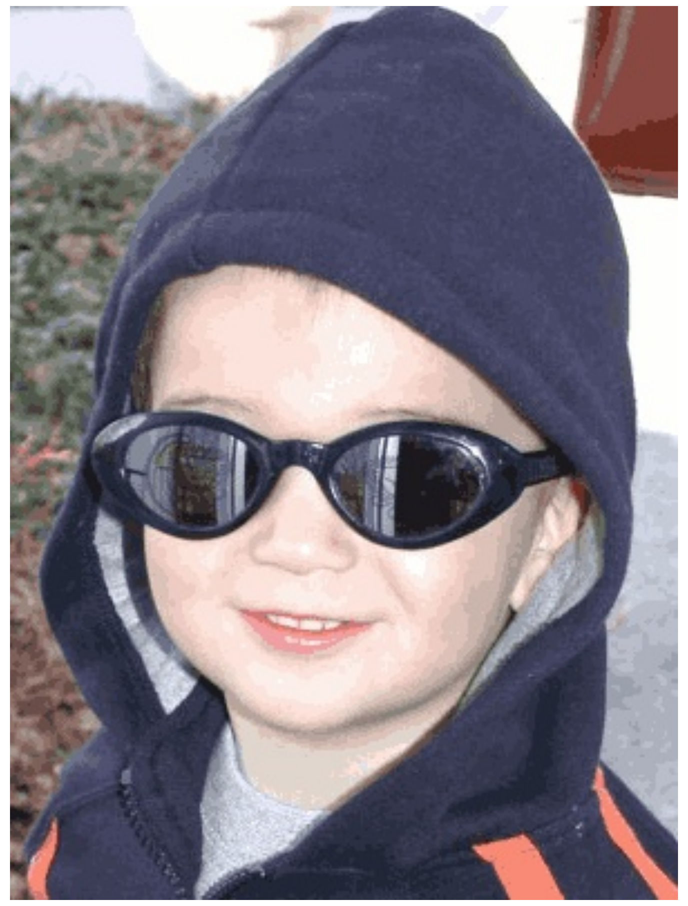
[Afifi 2018]



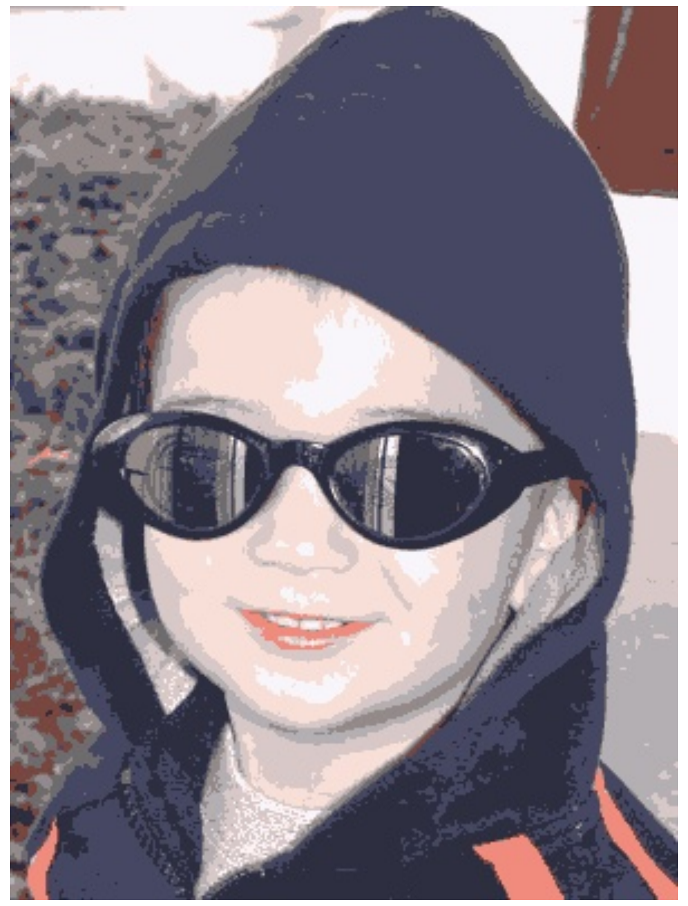
[Xu and Kaplan 2008]



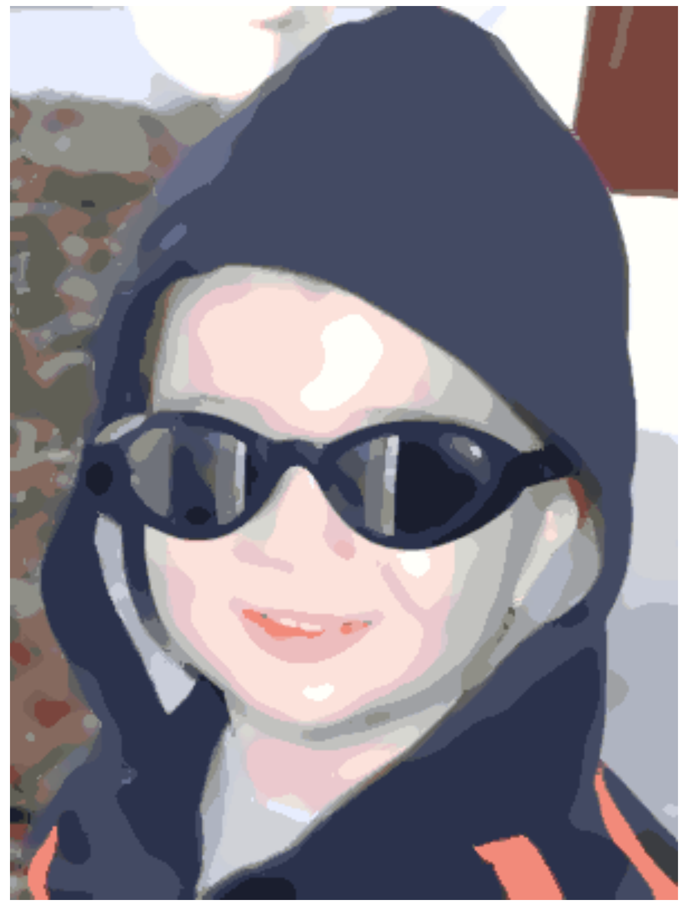
Direct K-means
(*K=7* in RGB-space)



Direct K-means
(*K=44* in RGB-space)



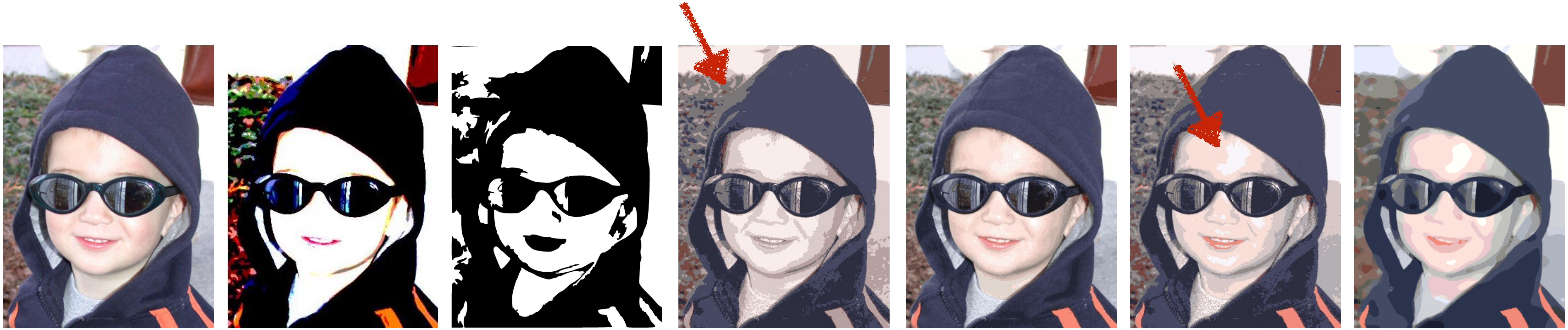
Direct K-means
(*K=10* in RGBXY-space)



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)

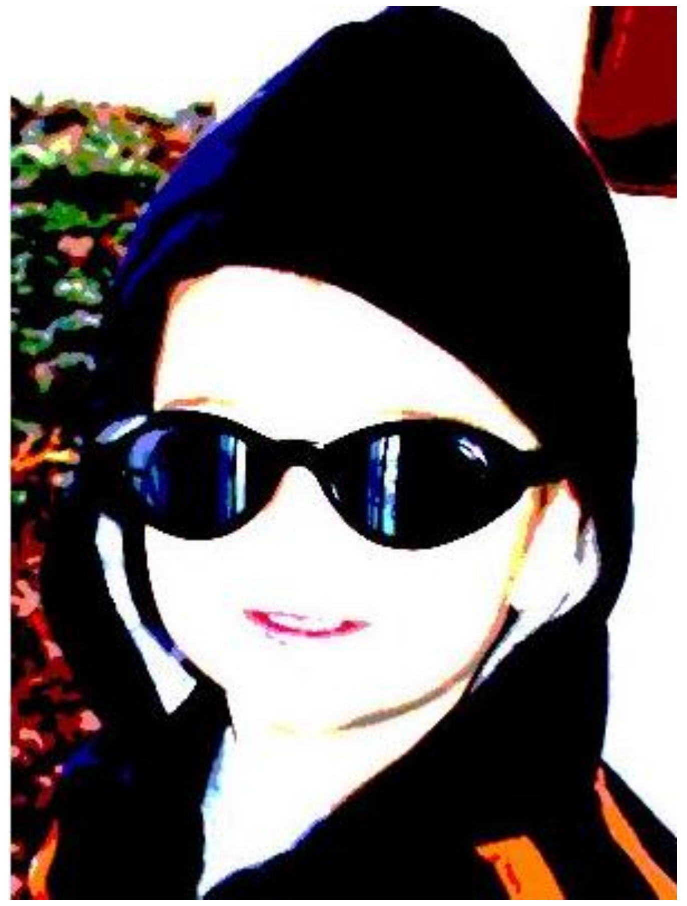
Direct K-means
(*K=10* in RGBXY-space)

Ours

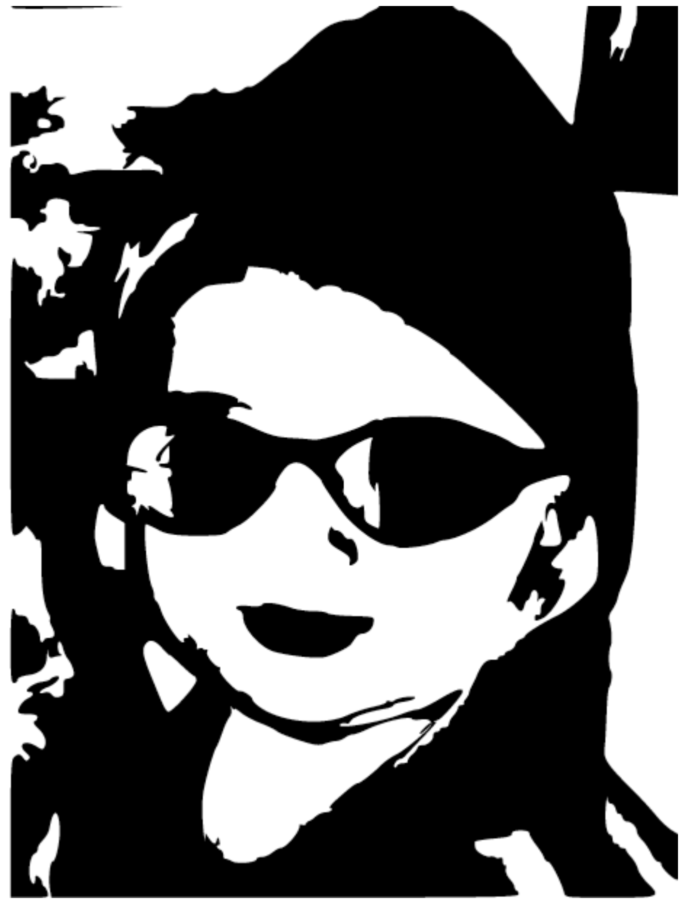
Evaluation

- Comparison to related approaches.

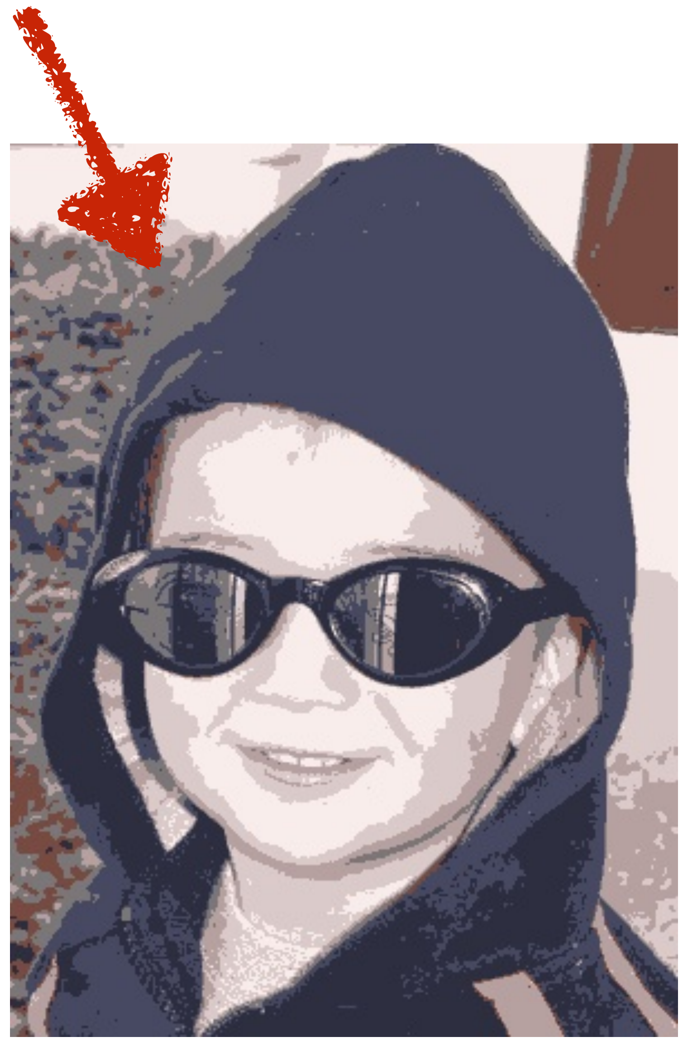
Input



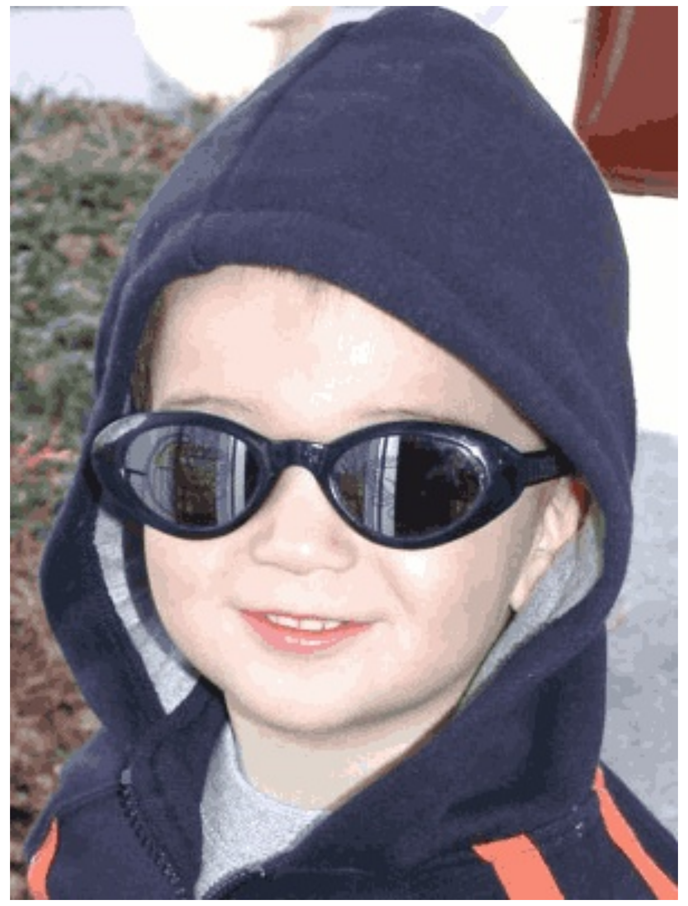
[Afifi 2018]



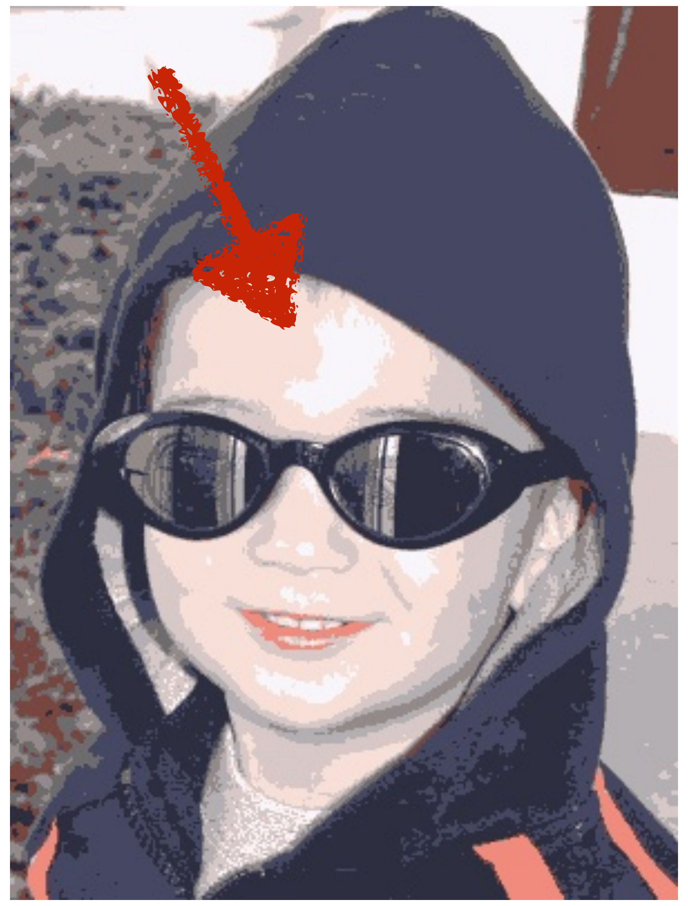
[Xu and Kaplan 2008]



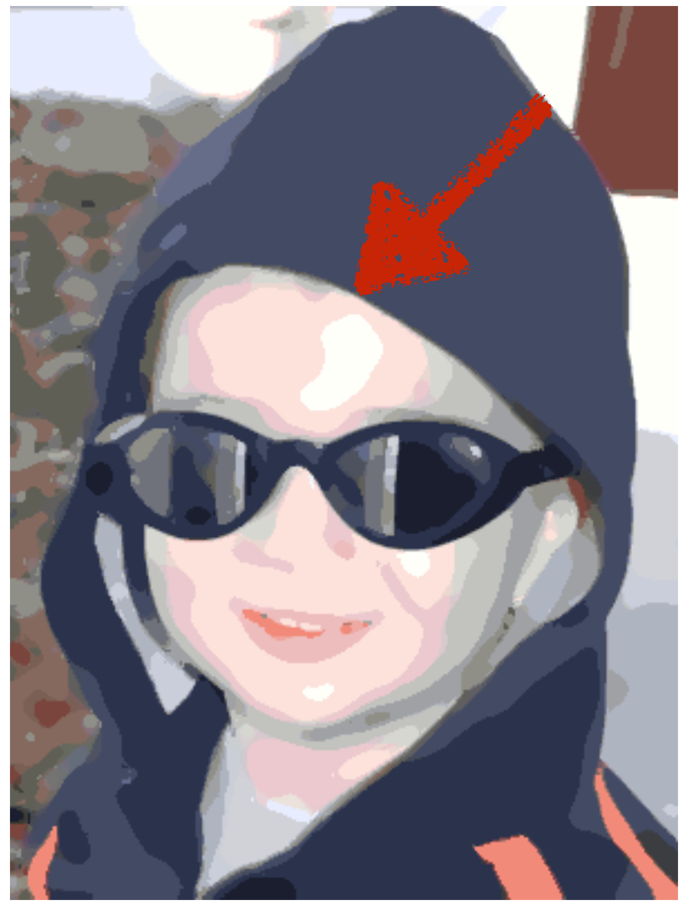
Direct K-means
(*K=7* in RGB-space)



Direct K-means
(*K=44* in RGB-space)



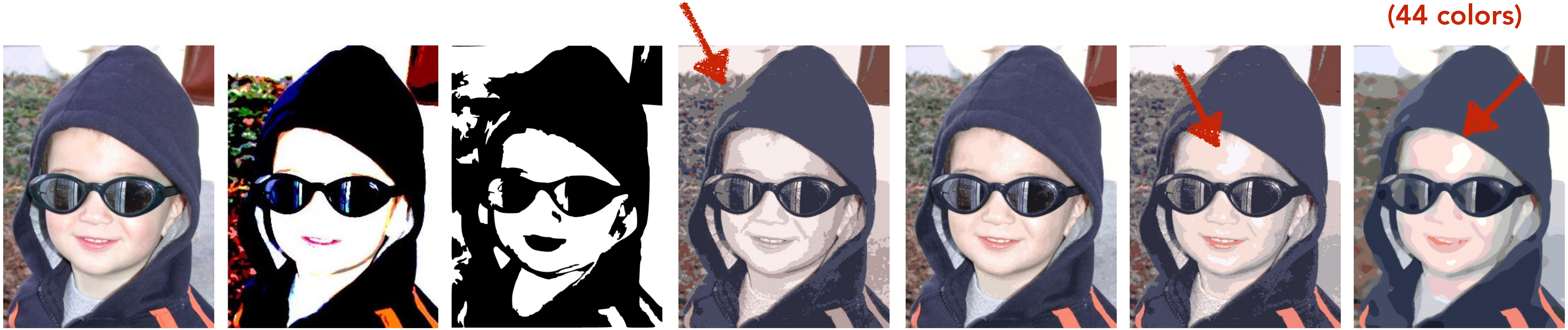
Direct K-means
(*K=10* in RGBXY-space)



Ours

Evaluation

- Comparison to related approaches.



(44 colors)

Input

[Afifi 2018]

[Xu and Kaplan 2008]

Direct K-means
(*K=7* in RGB-space)

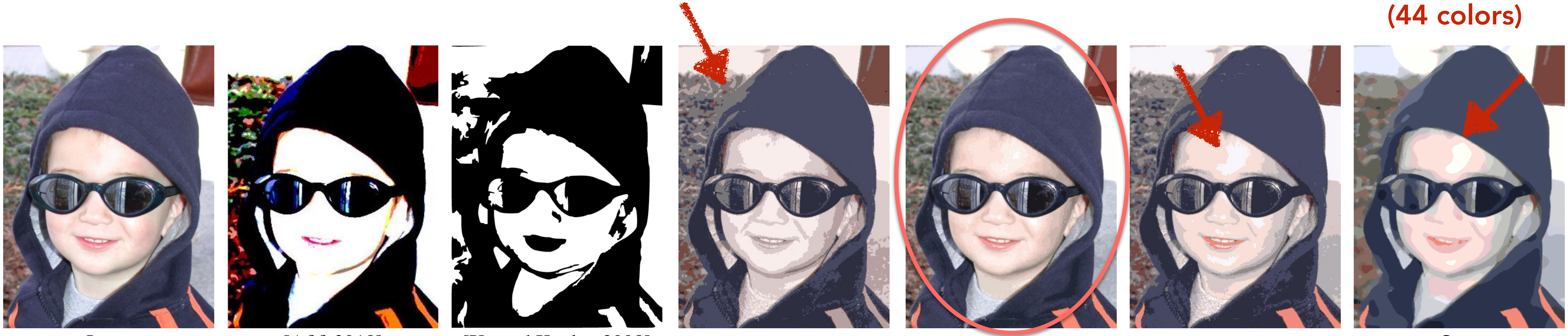
Direct K-means
(*K=44* in RGB-space)

Direct K-means
(*K=10* in RGBXY-space)

Ours

Evaluation

- Comparison to related approaches.



(44 colors)

Input

[Afifi 2018]

[Xu and Kaplan 2008]

Direct K-means
(*K=7* in RGB-space)

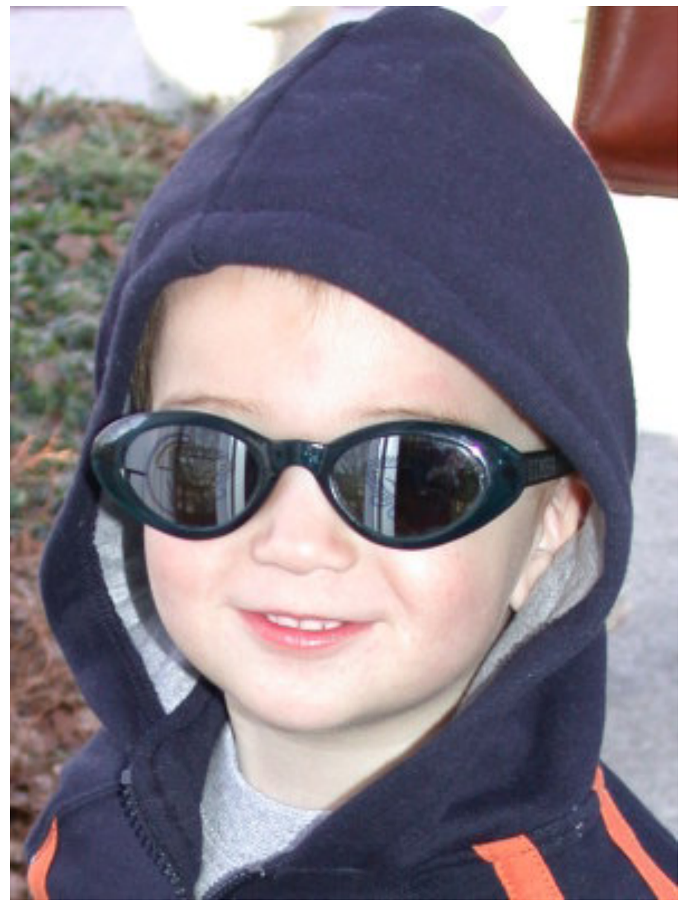
Direct K-means
(*K=44* in RGB-space)

Direct K-means
(*K=10* in RGBXY-space)

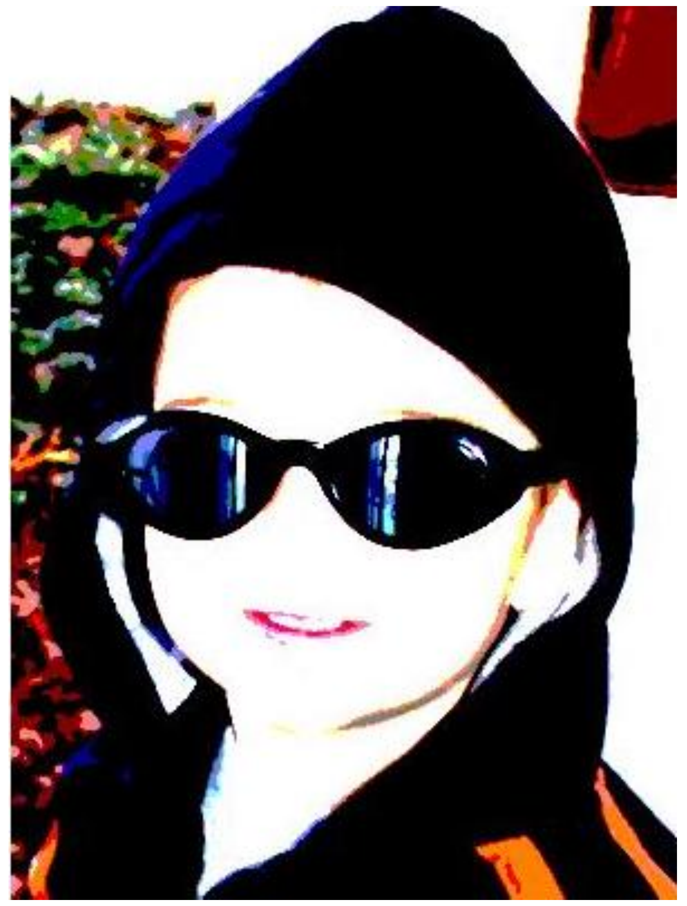
Ours

Evaluation

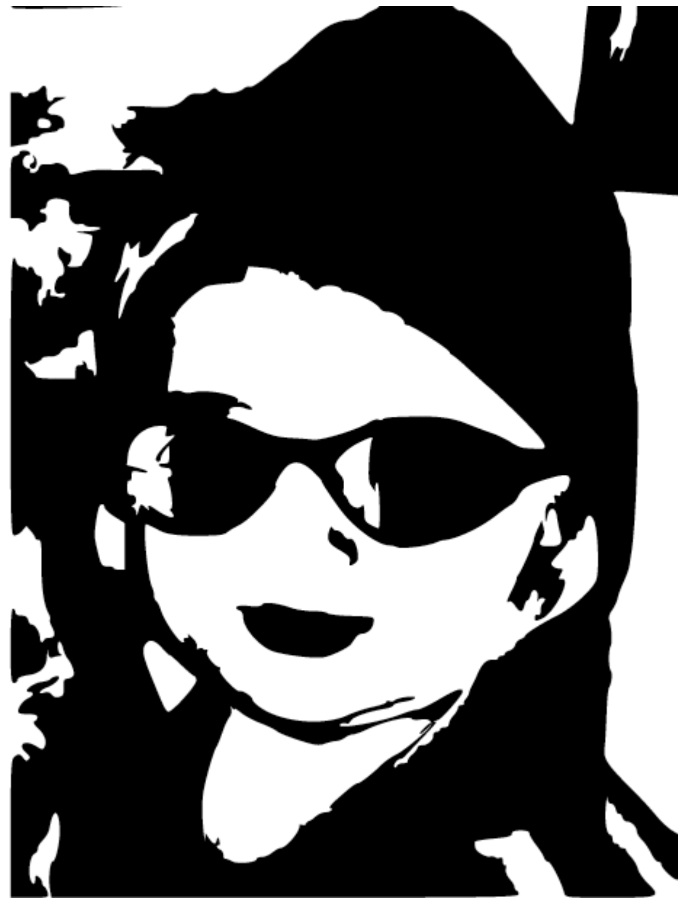
- Comparison to related approaches.
- Expert study with professional artists.



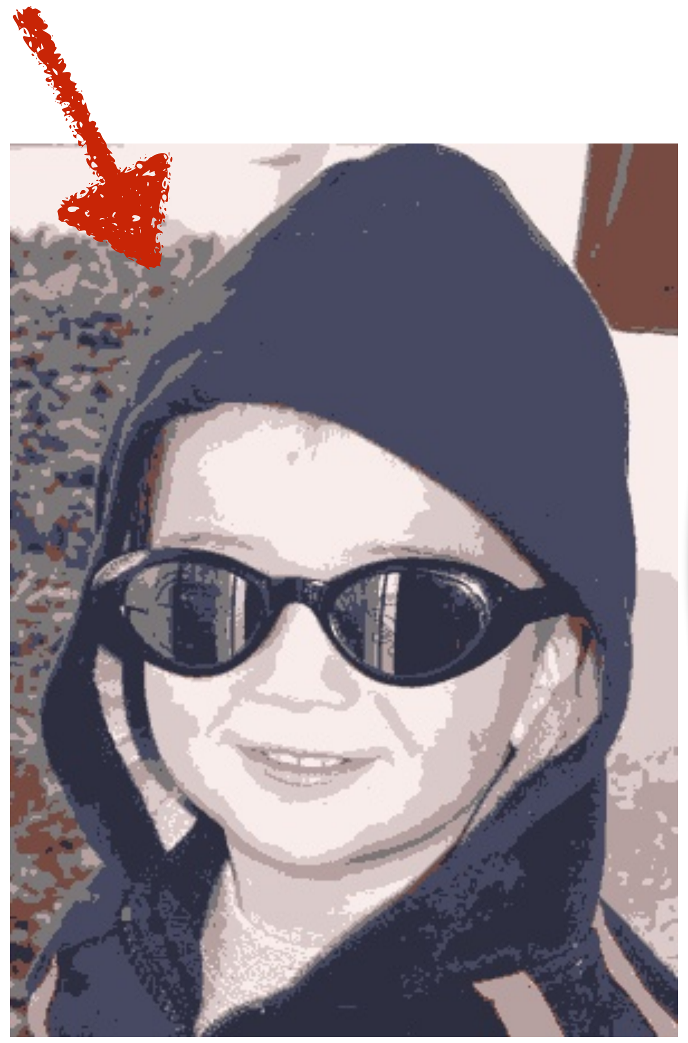
Input



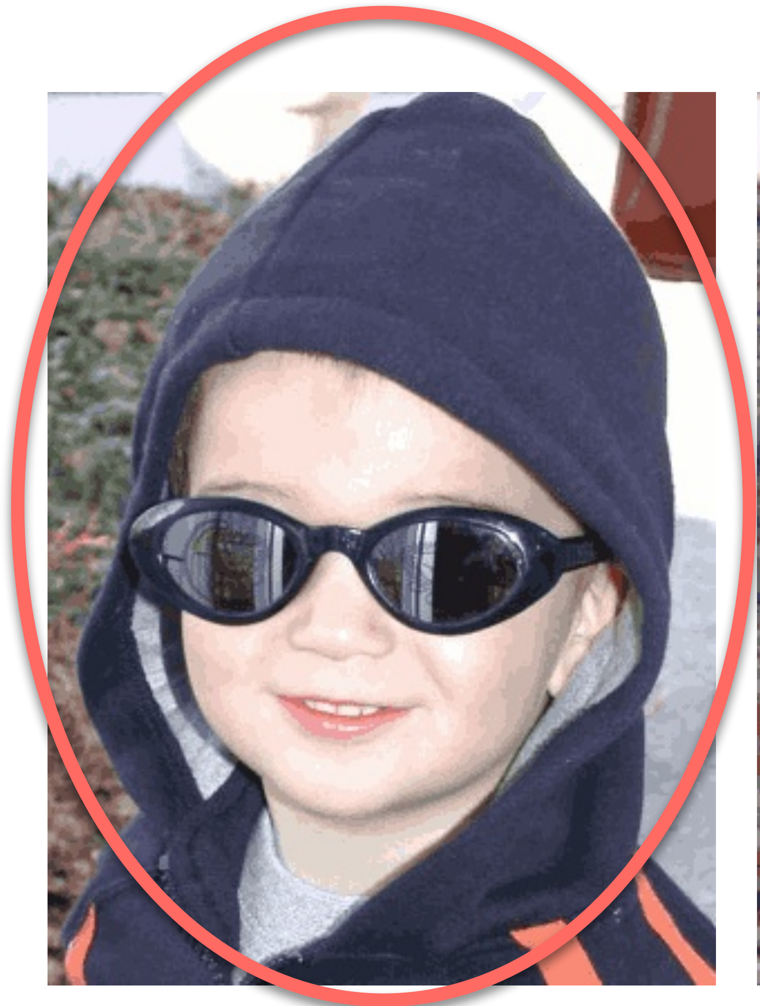
[Afifi 2018]



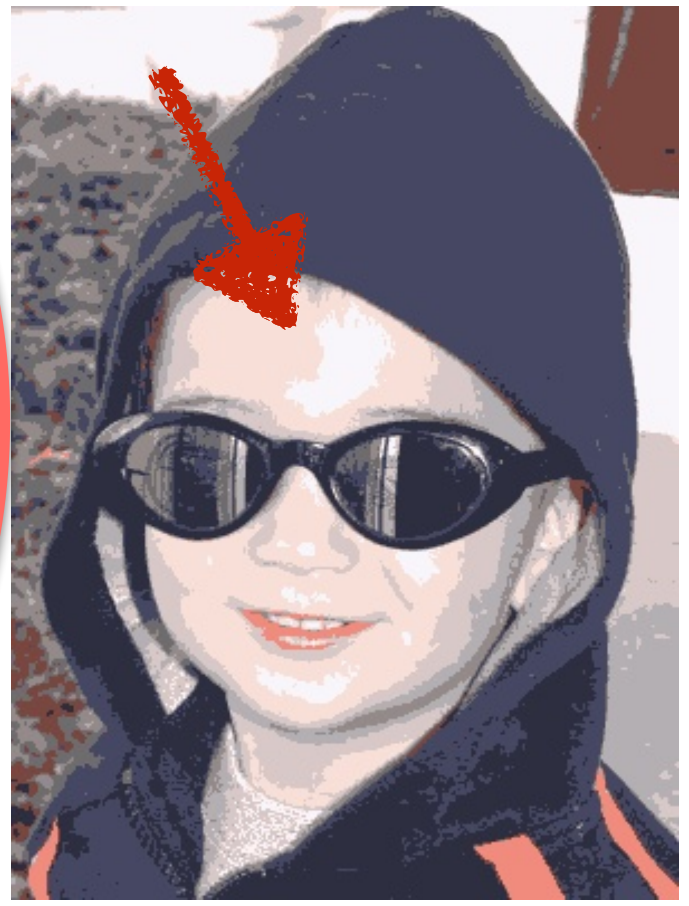
[Xu and Kaplan 2008]



Direct K-means
(*K*=*7* in RGB-space)

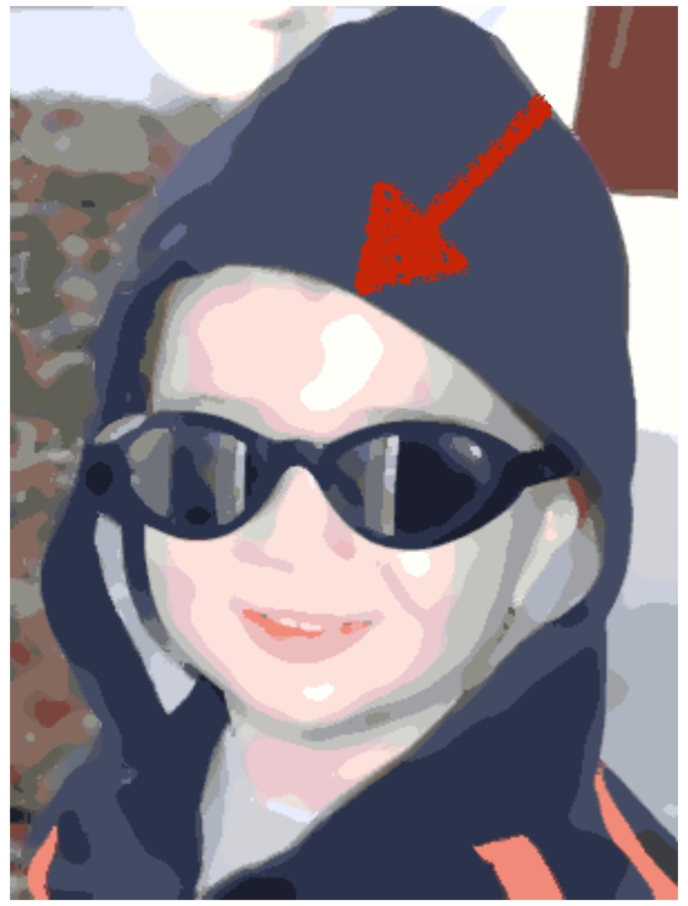


Direct K-means
(*K*=*44* in RGB-space)



Direct K-means
(*K*=*10* in RGBXY-space)

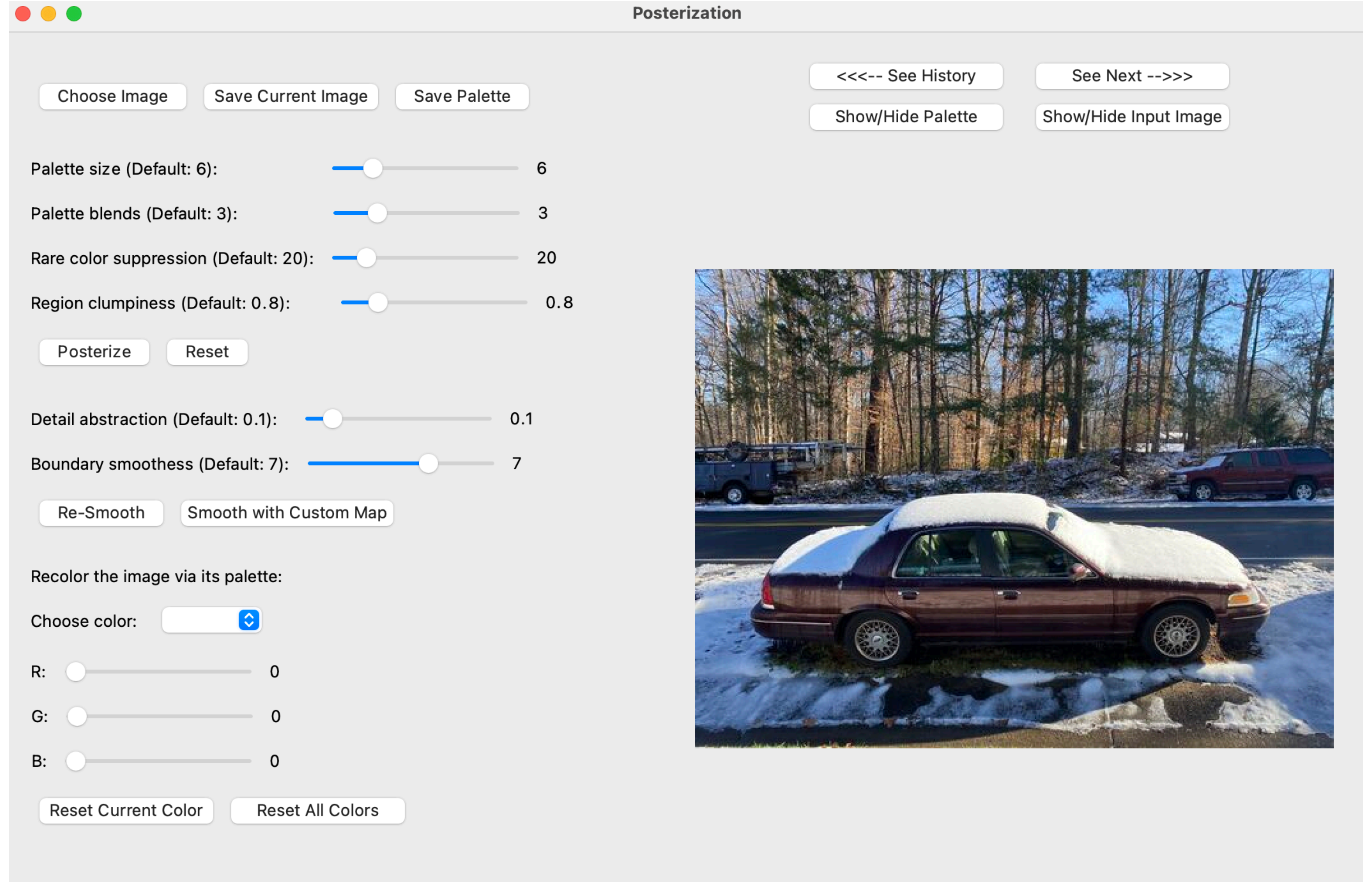
(44 colors)



Ours

Evaluation

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



Palette size (Default: 6): 6

Palette blends (Default: 3): 3

Rare color suppression (Default: 20): 20

Region clumpiness (Default: 0.8): 0.8

Detail abstraction (Default: 0.1): 0.1

Boundary smoothess (Default: 7): 7

Recolor the image via its palette:

Choose color:

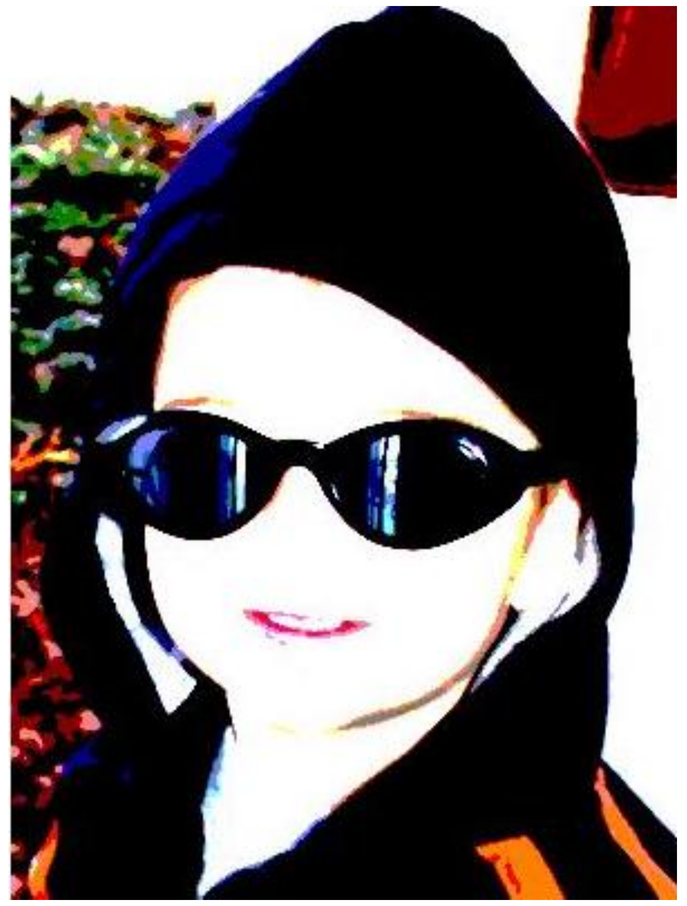
R: 0

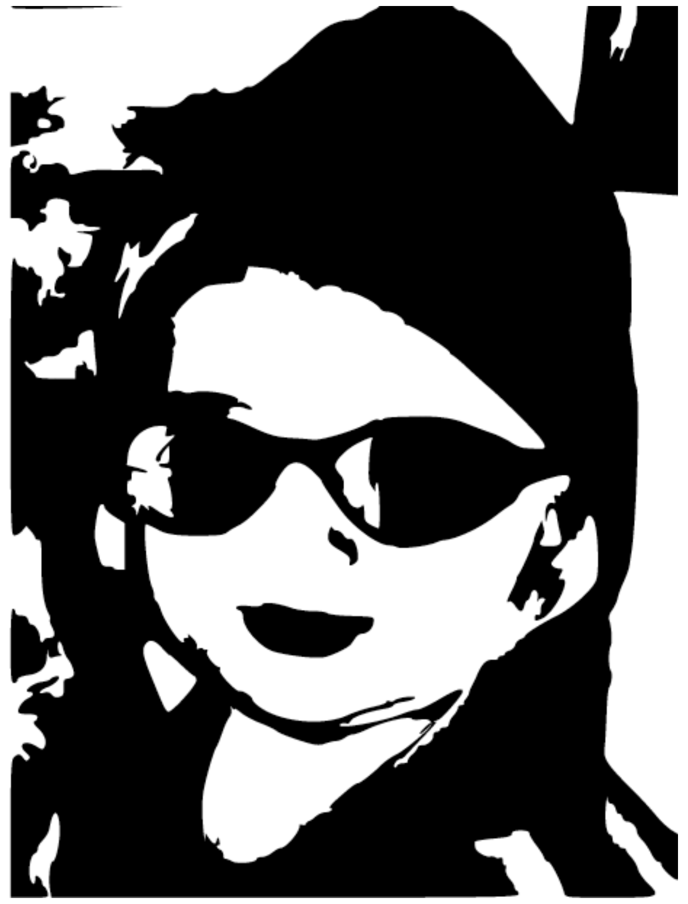
G: 0

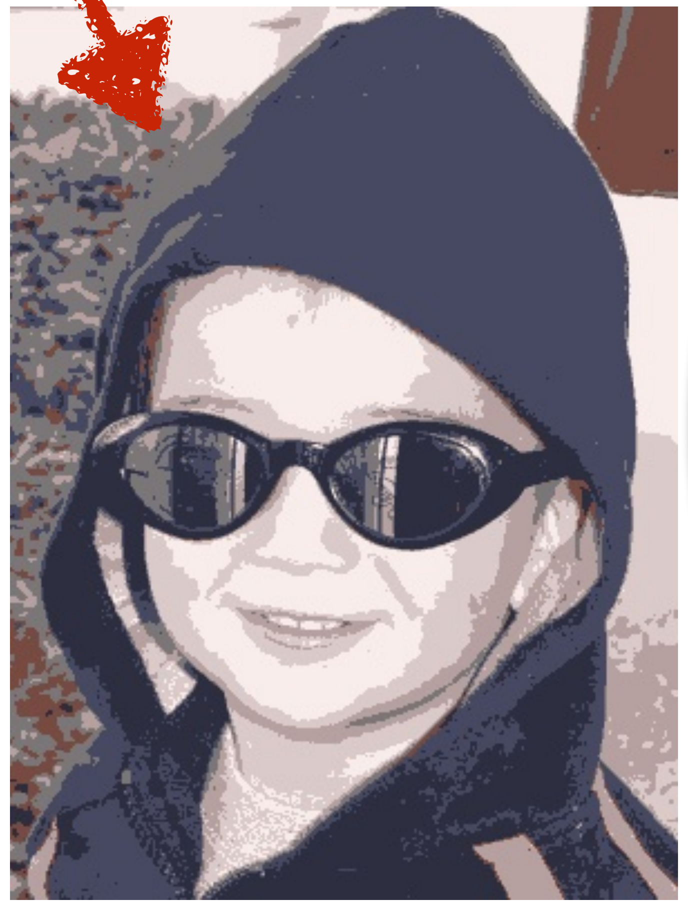
B: 0

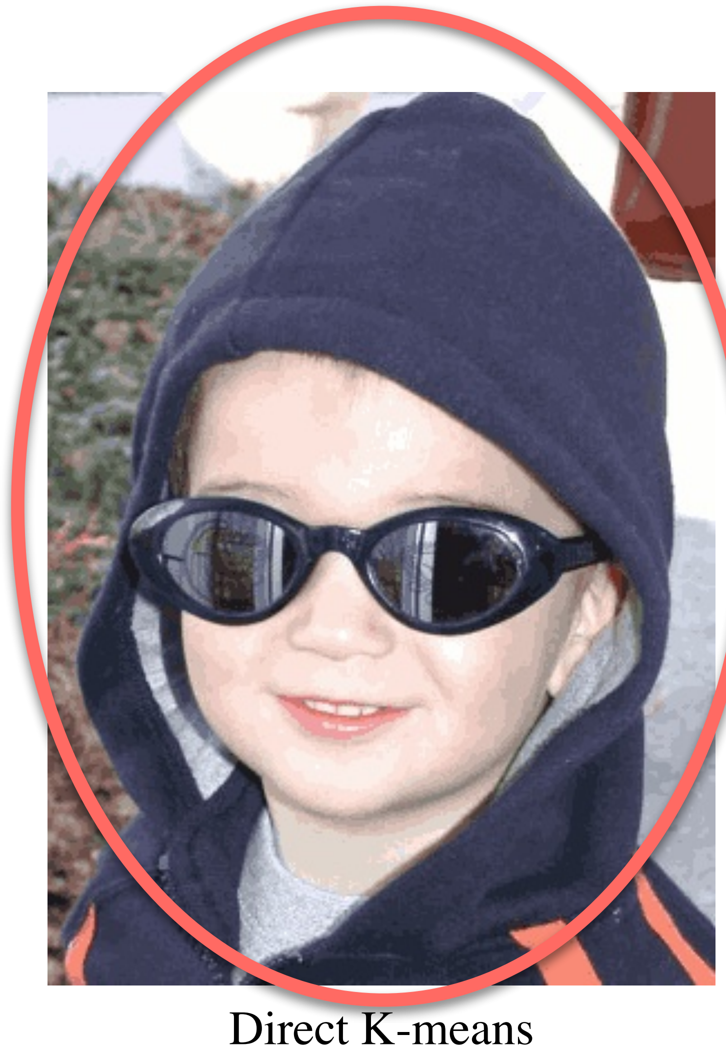
(44 colors)

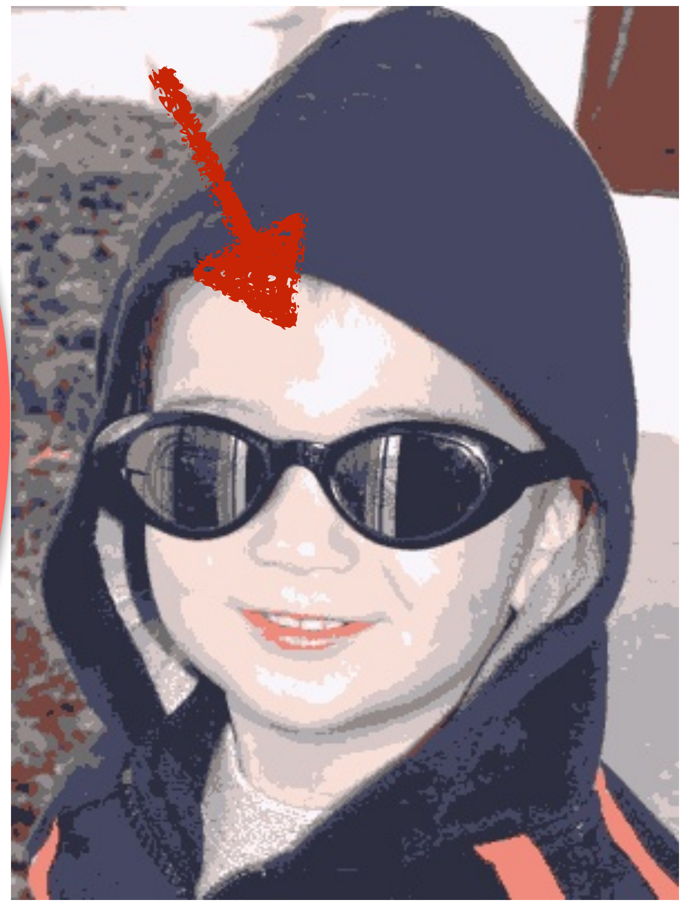
Input

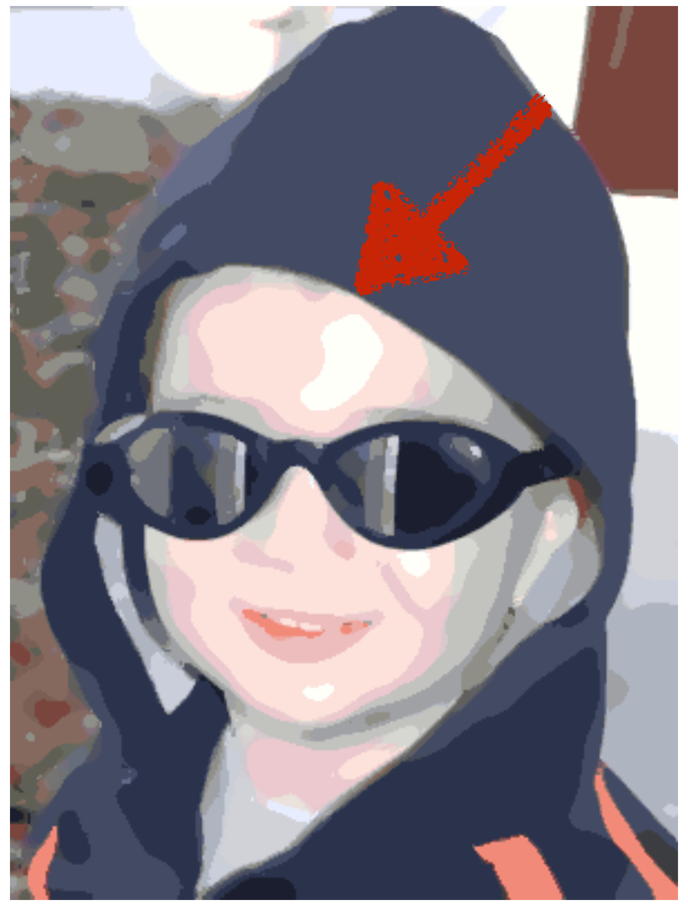

[Afifi 2018]


[Xu and Kaplan 2008]


Direct K-means
(*K=7* in RGB-space)

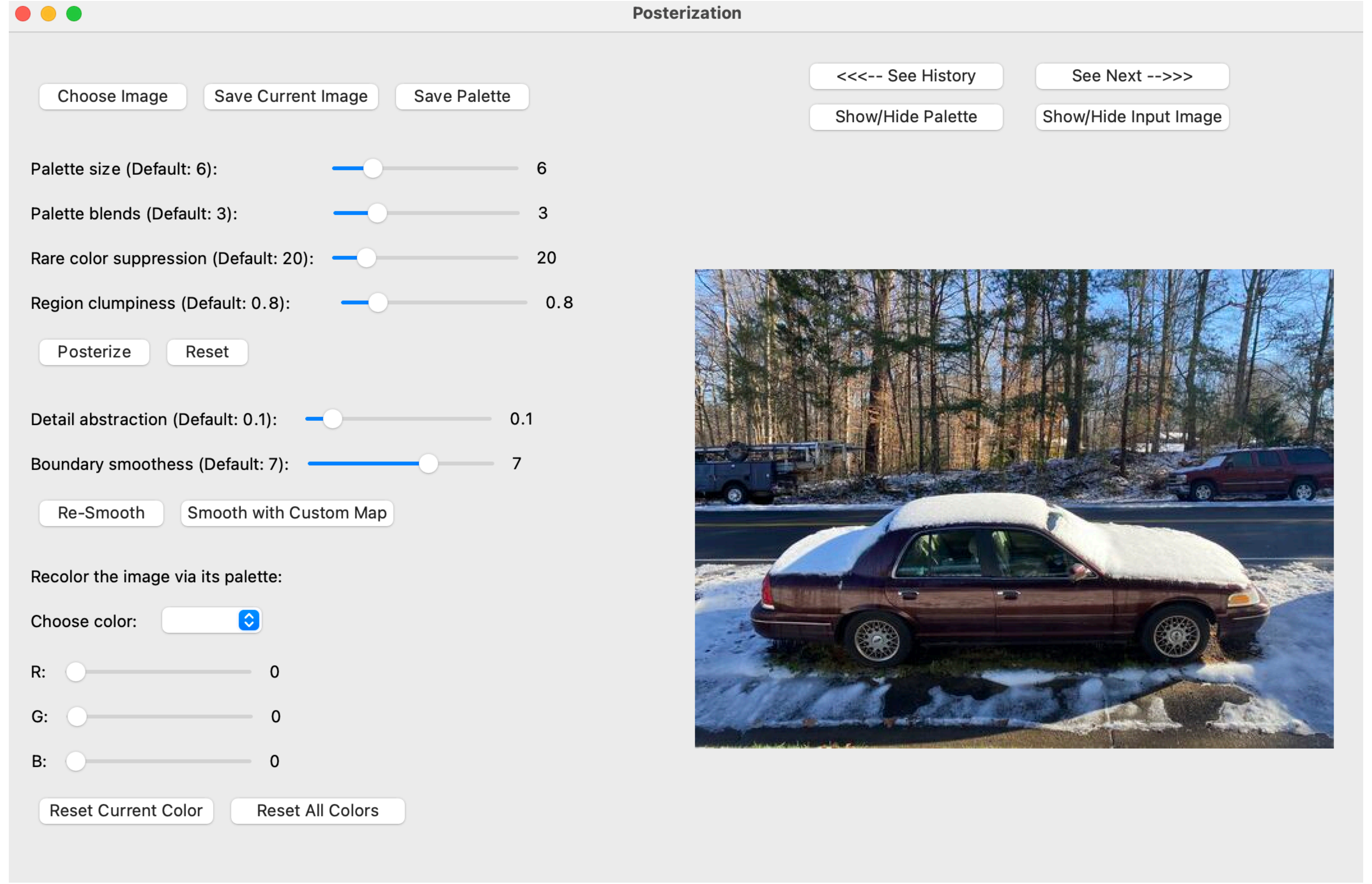

Direct K-means
(*K=44* in RGB-space)


Direct K-means
(*K=10* in RGBXY-space)


Ours

Evaluation

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



Palette size (Default: 6): 6

Palette blends (Default: 3): 3

Rare color suppression (Default: 20): 20

Region clumpiness (Default: 0.8): 0.8

Detail abstraction (Default: 0.1): 0.1

Boundary smoothess (Default: 7): 7

Recolor the image via its palette:

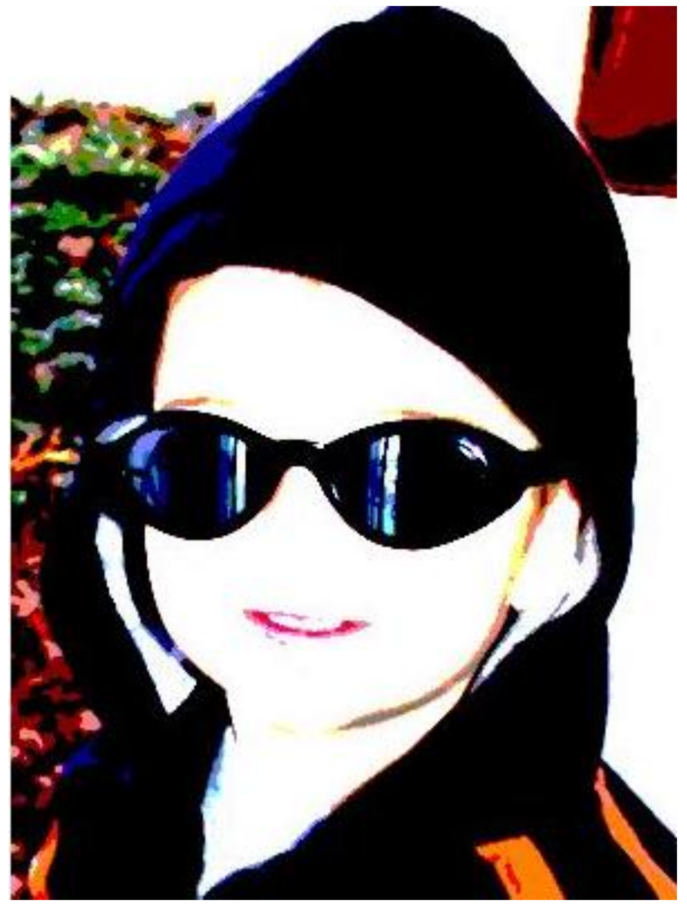
Choose color:

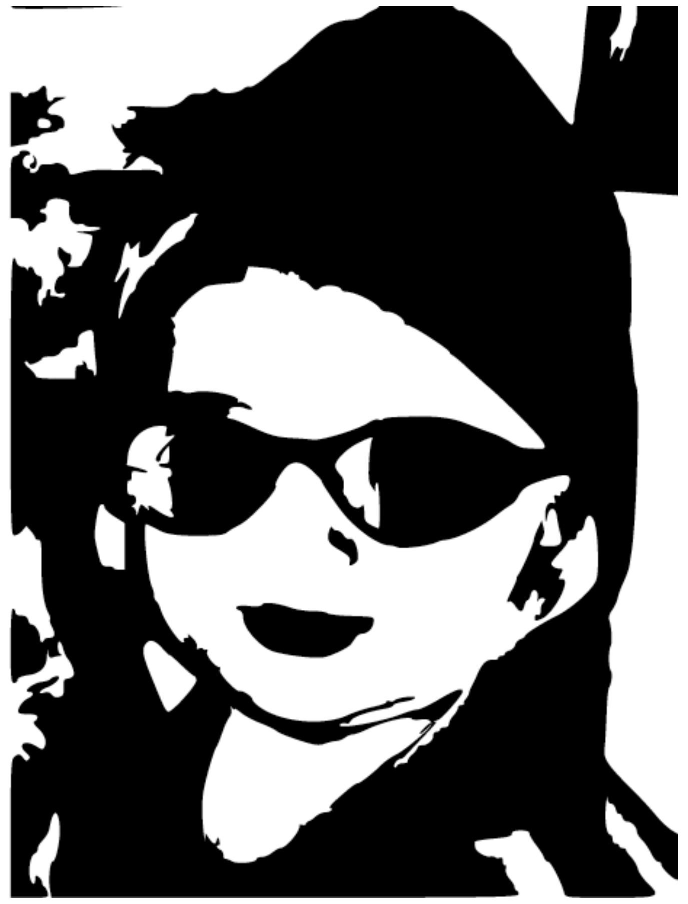
R: 0

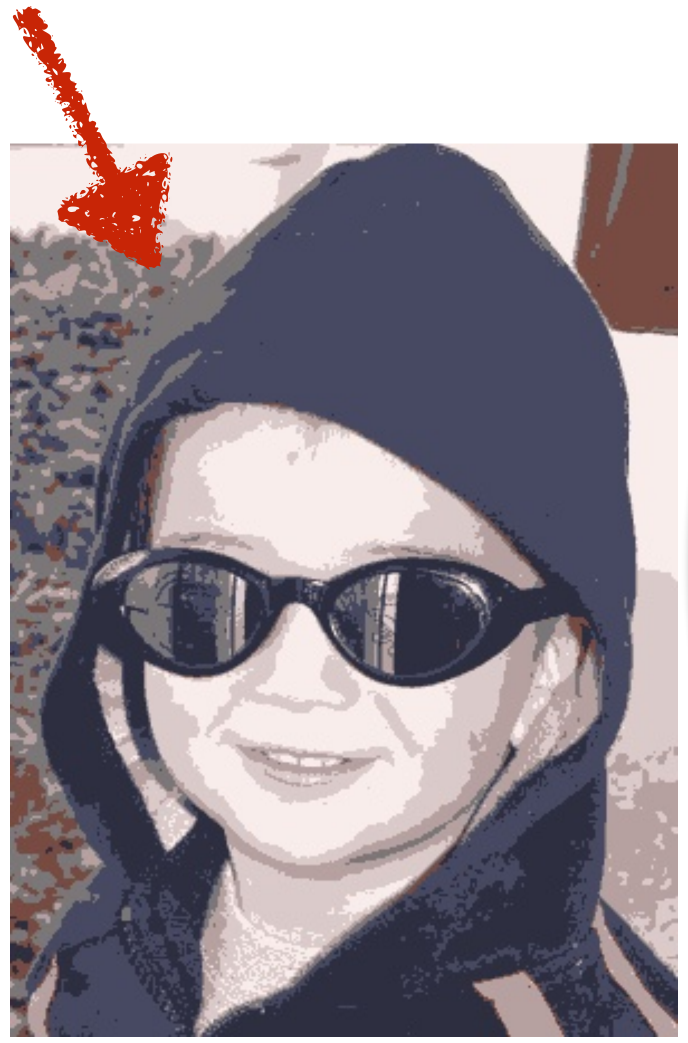
G: 0

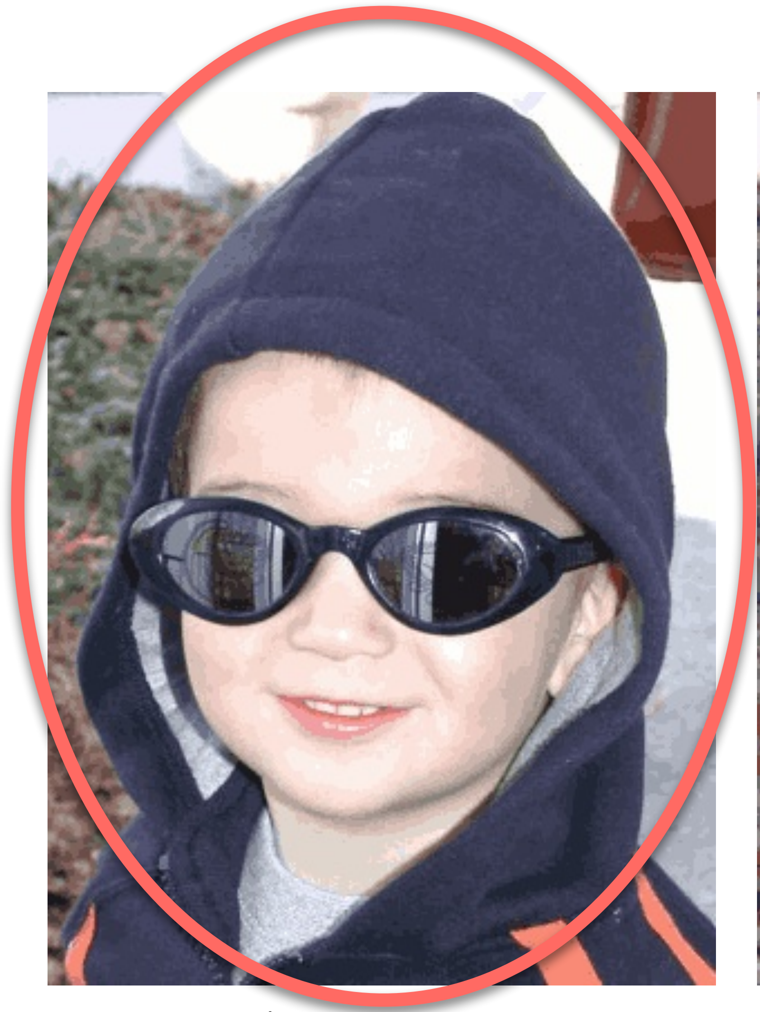
B: 0

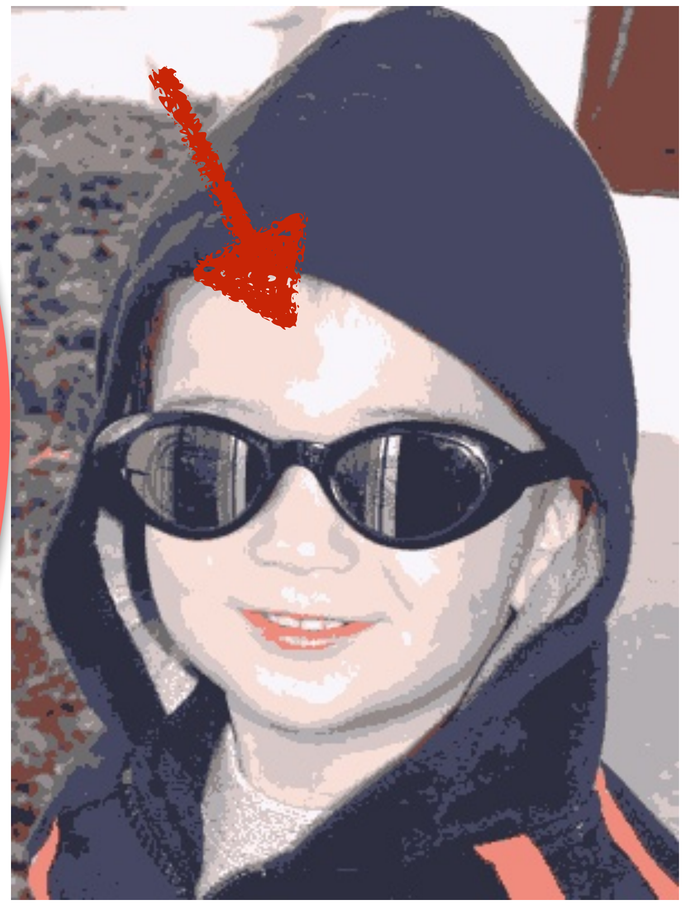
Input


[Afifi 2018]

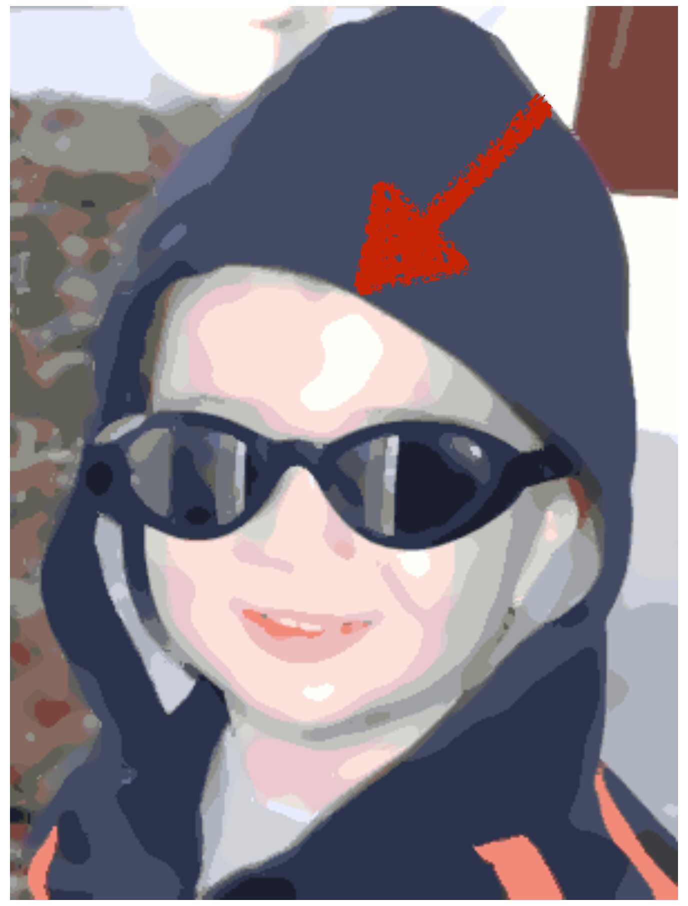

[Xu and Kaplan 2008]


Direct K-means
(K=7 in RGB-space)


Direct K-means
(K=44 in RGB-space)


Direct K-means
(K=10 in RGBXY-space)

(44 colors)


Ours

Conclusion

Conclusion

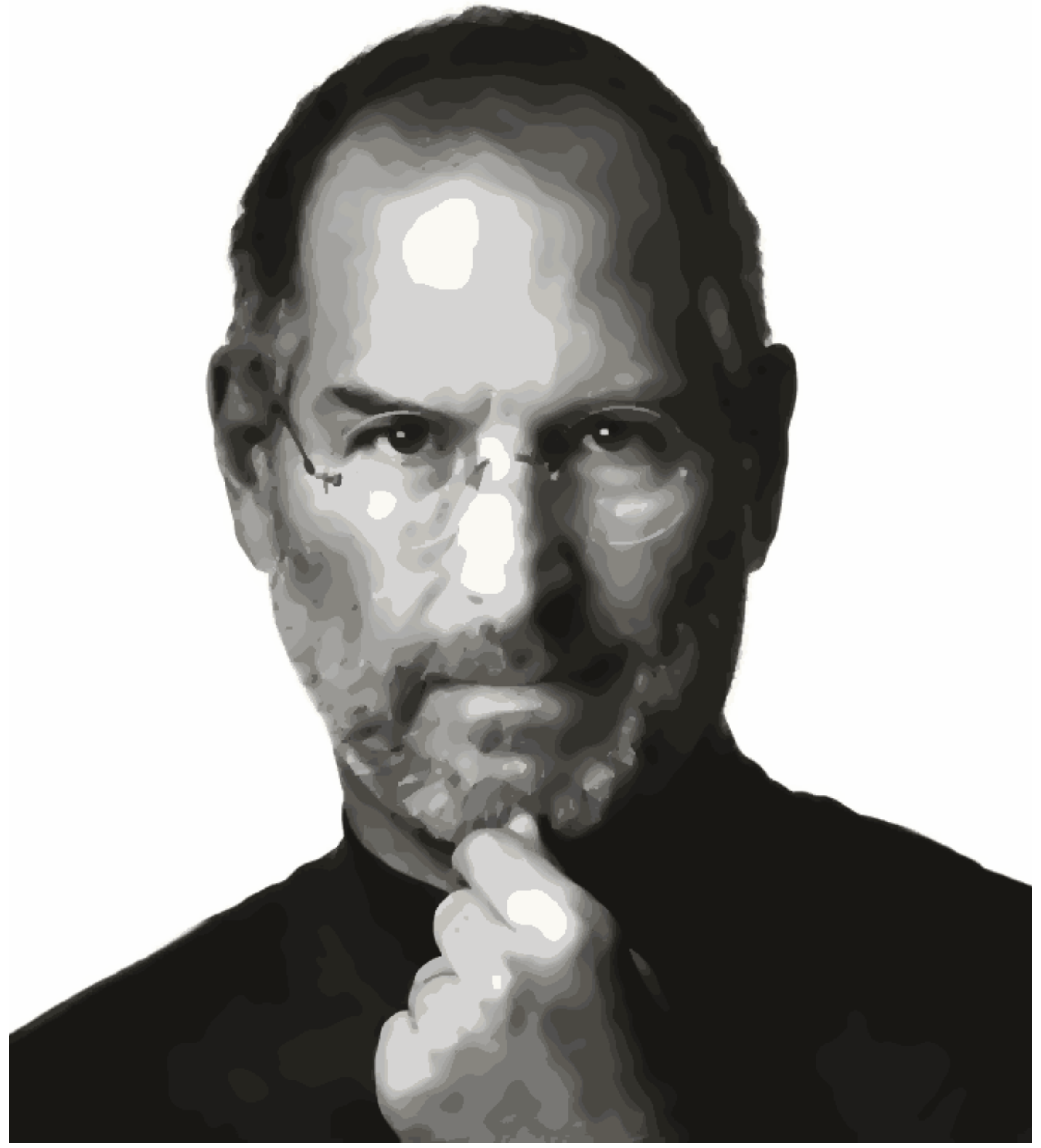
- PosterChild shows:

Conclusion

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

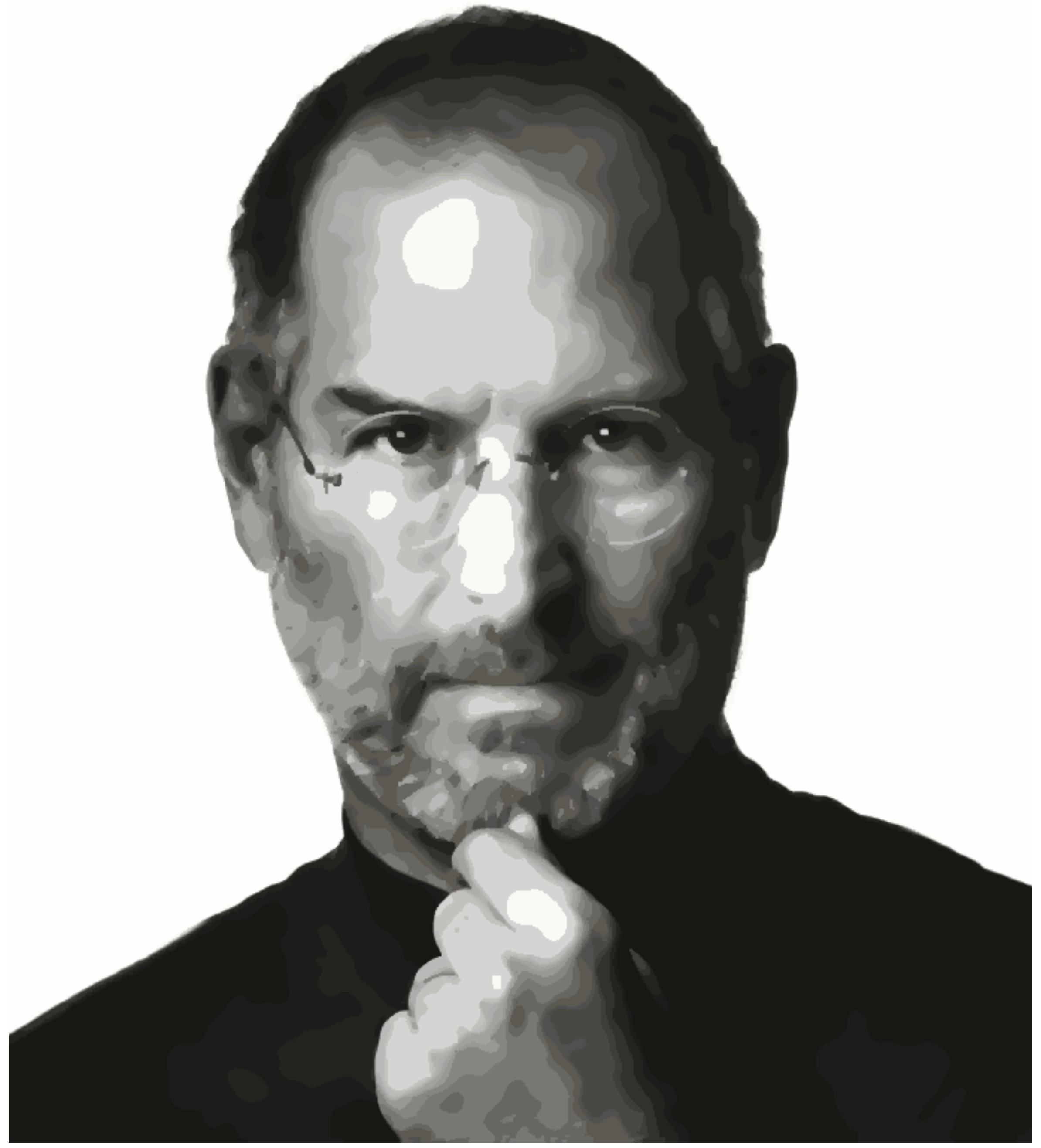
Conclusion

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



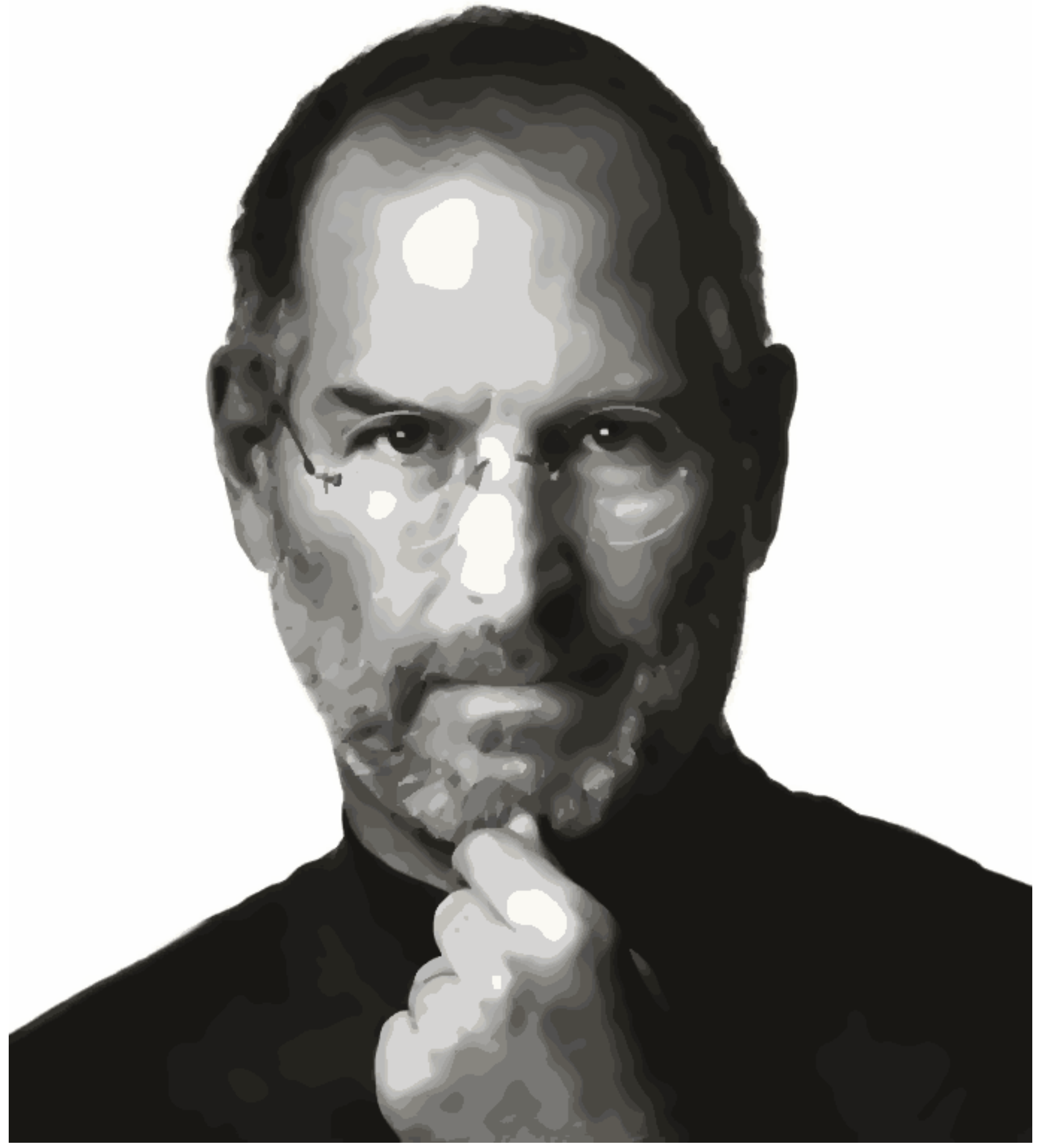
Conclusion

- 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.



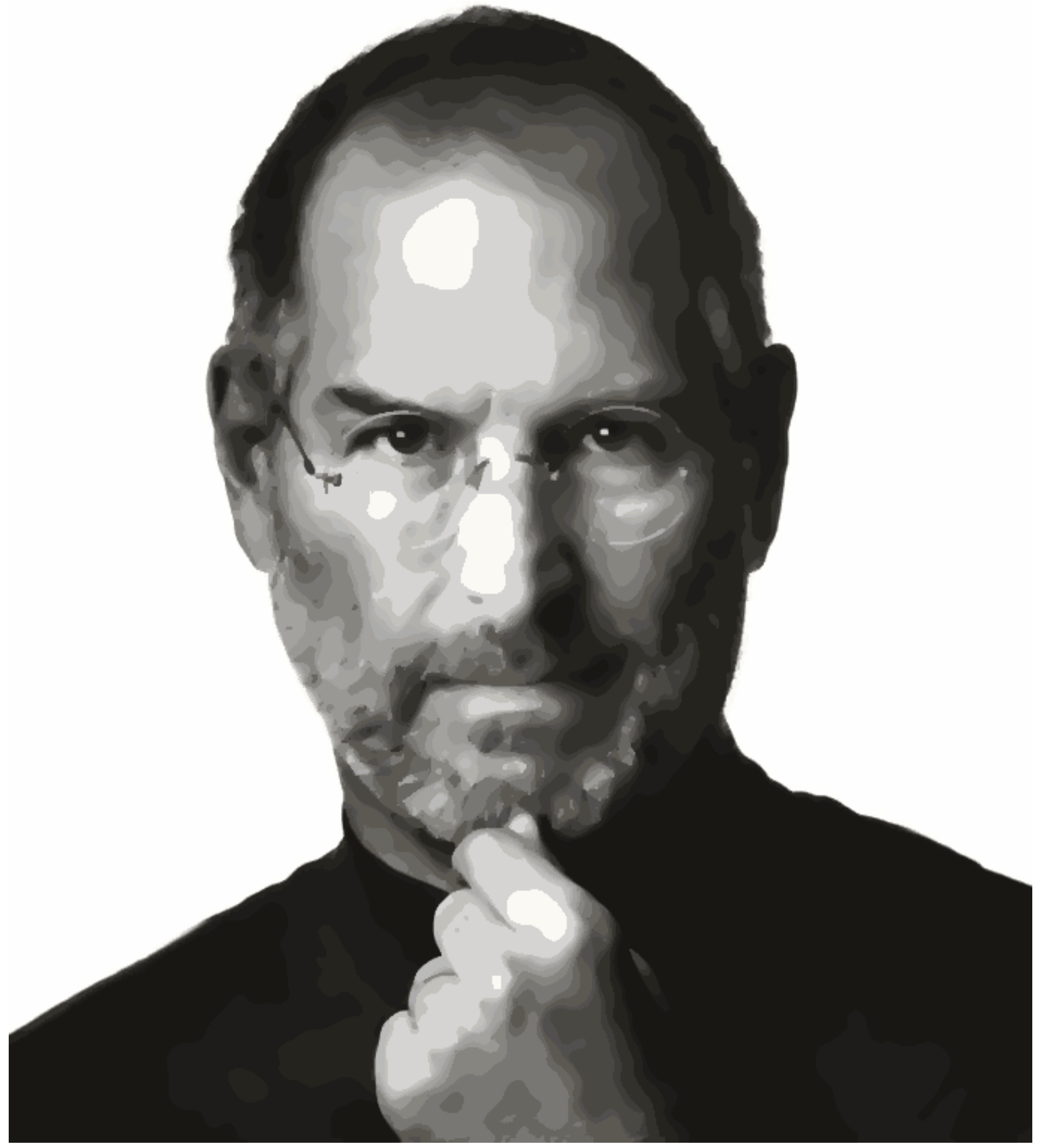
Conclusion

- 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.



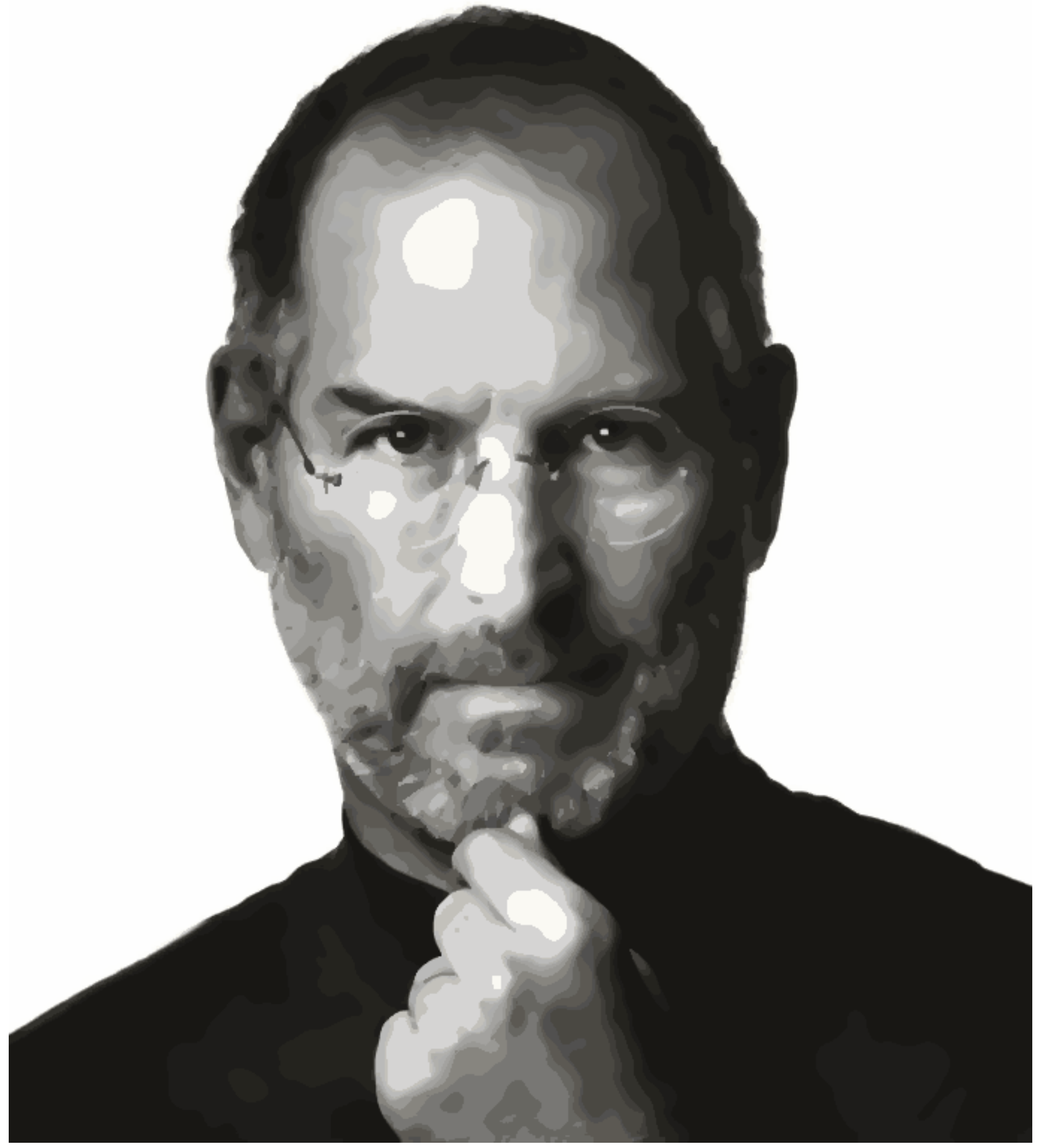
Conclusion

- 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:



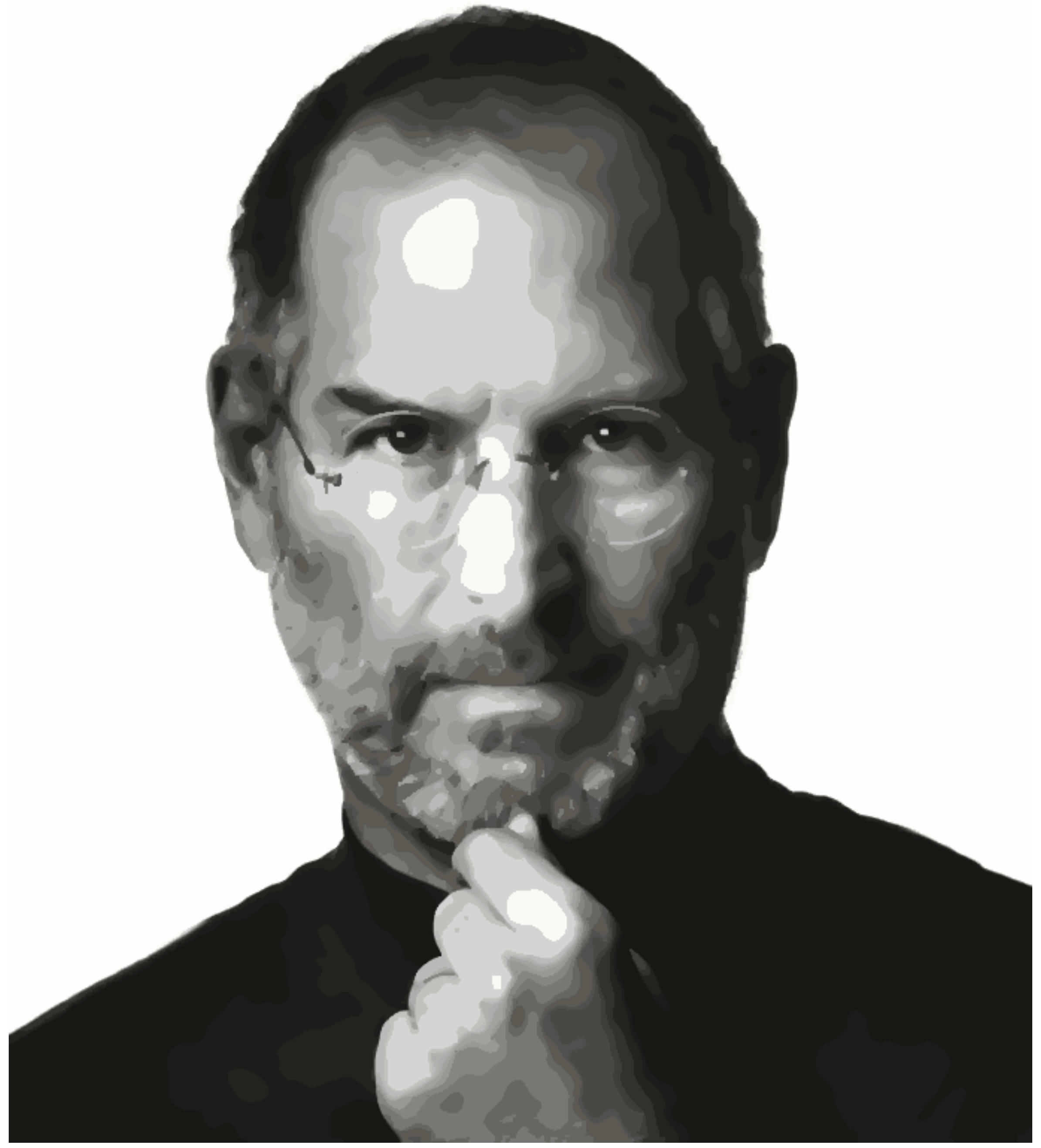
Conclusion

- 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.



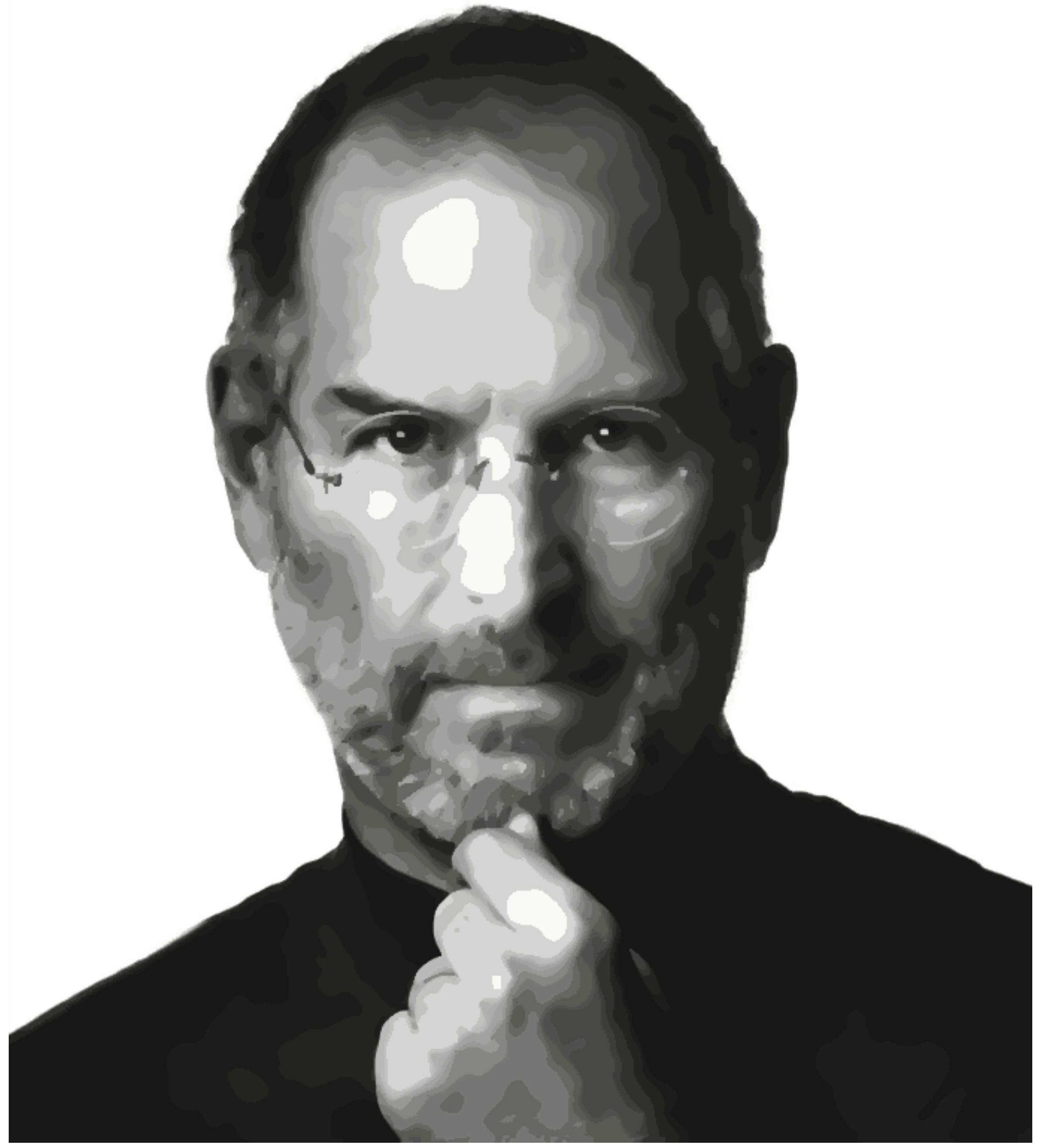
Conclusion

- 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

Step 1: Convex-hull based palette extraction

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

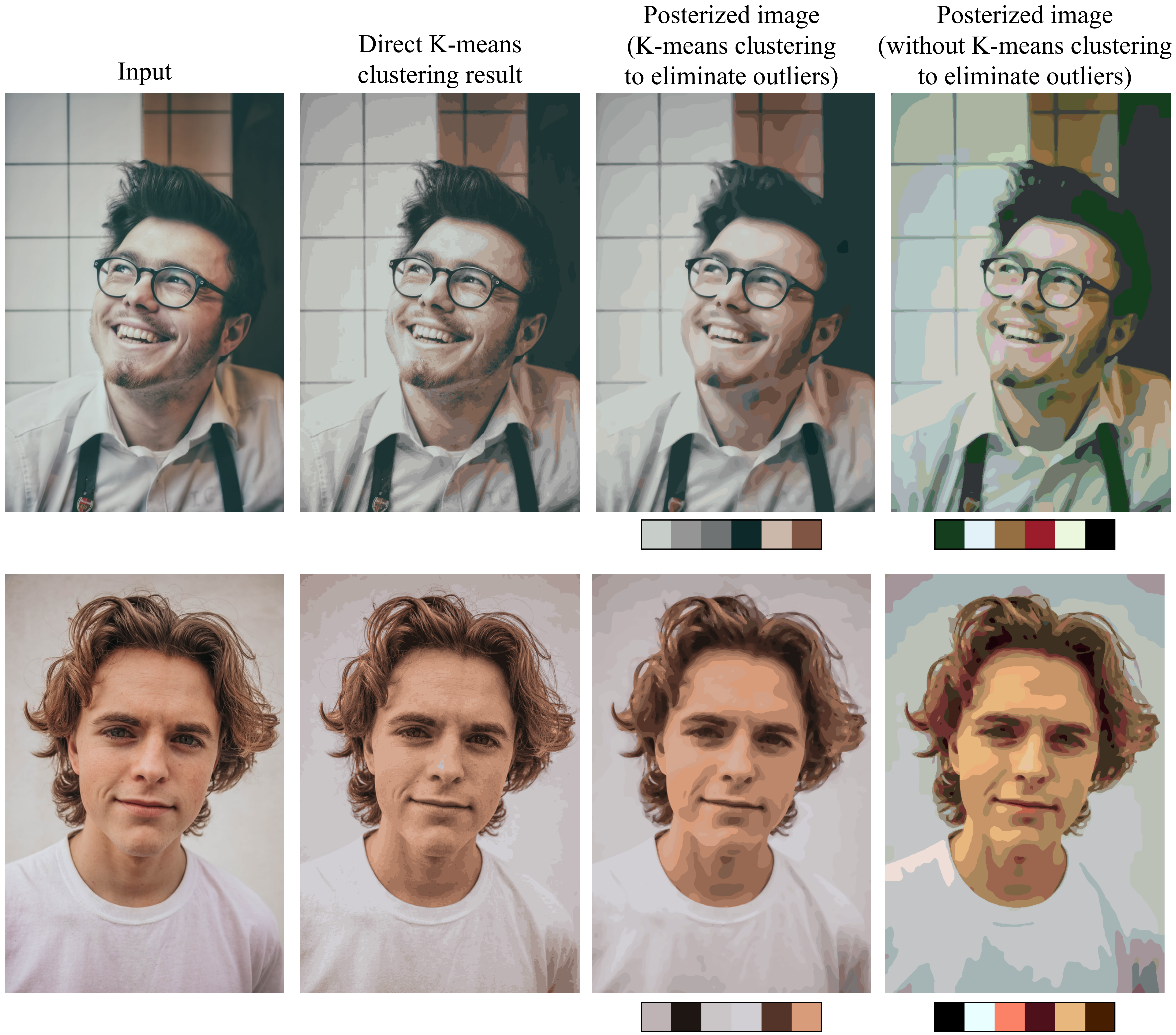
Step 1: Convex-hull based palette extraction

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

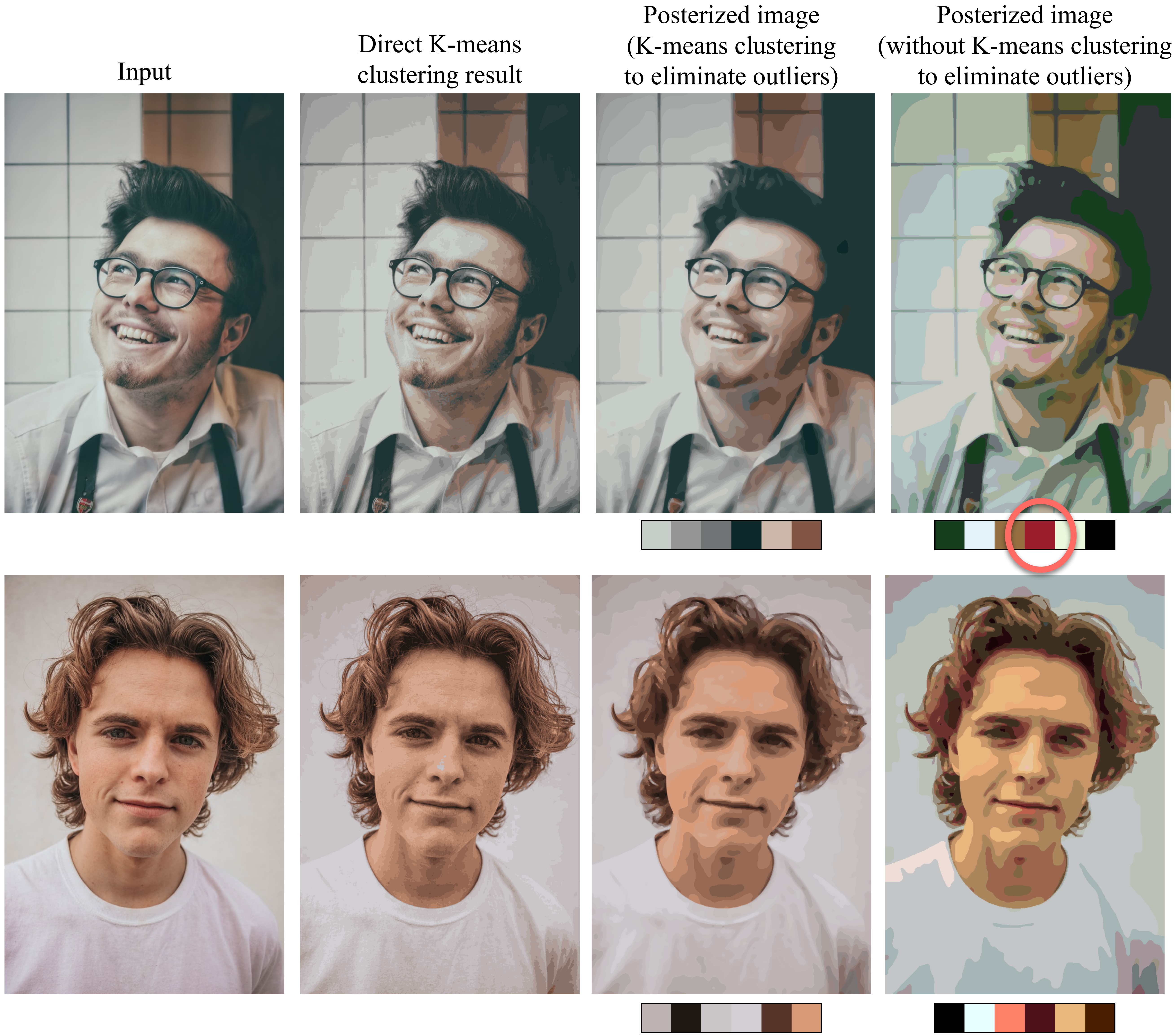
Step 1: Convex-hull based palette extraction

Step 1: Convex-hull based palette extraction

Step 1: Convex-hull based palette extraction


Input
Direct K-means clustering result
Posterized image (K-means clustering to eliminate outliers)
Posterized image (without K-means clustering to eliminate outliers)

Step 1: Convex-hull based palette extraction



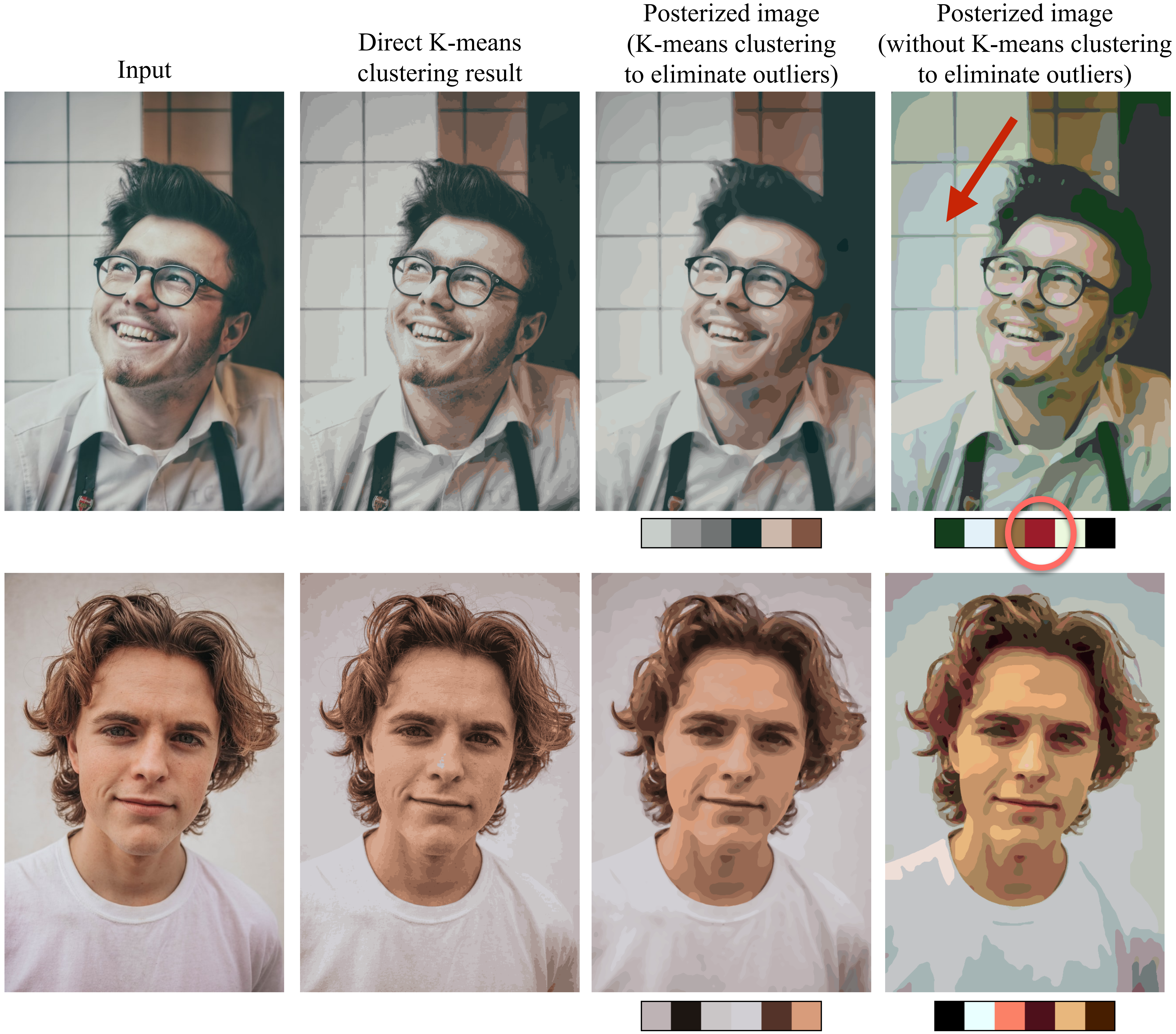
Input

Direct K-means clustering result

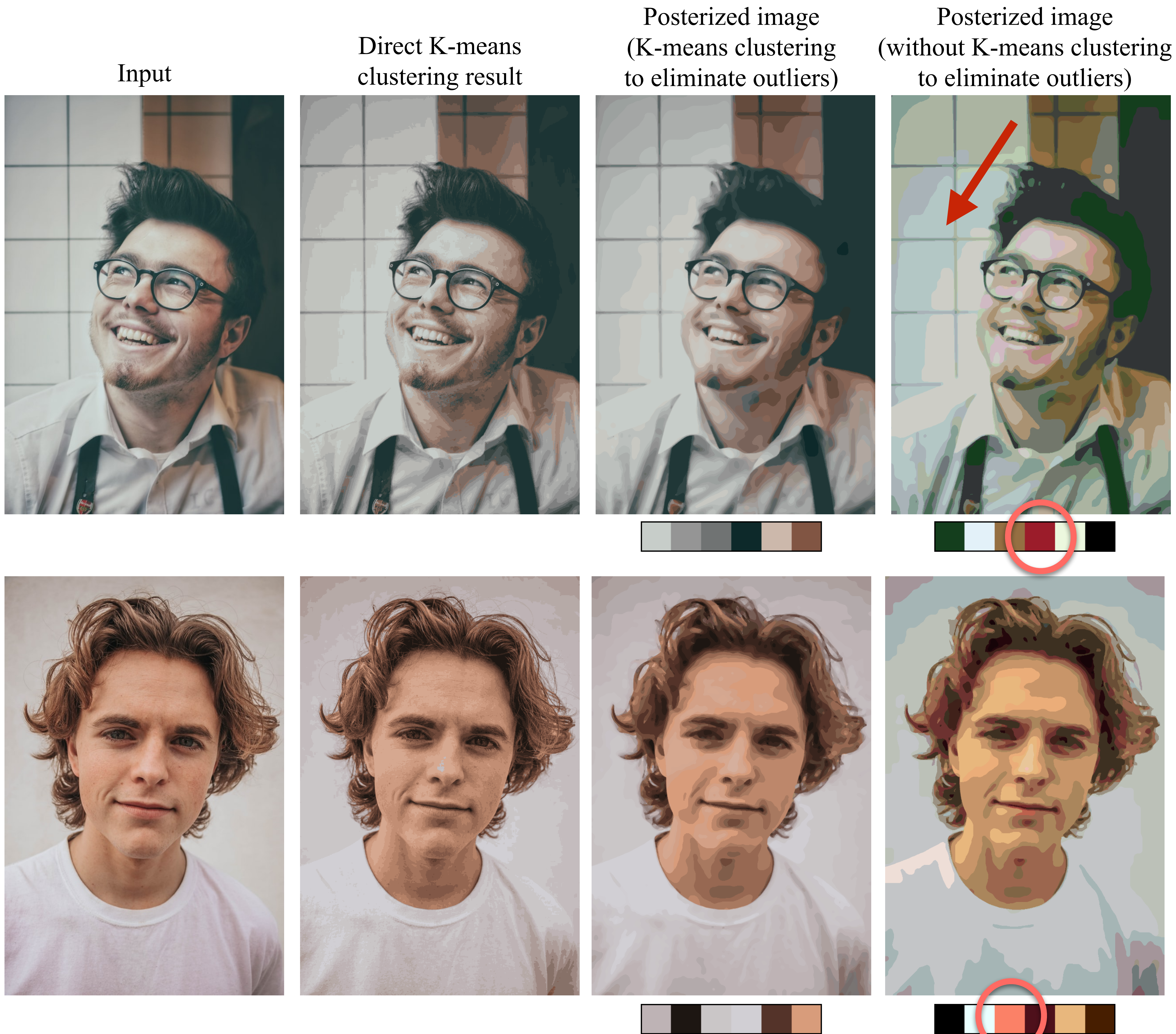
Posterized image (K-means clustering to eliminate outliers)

Posterized image (without K-means clustering to eliminate outliers)

Step 1: Convex-hull based palette extraction


Input
Direct K-means clustering result
Posterized image (K-means clustering to eliminate outliers)
Posterized image (without K-means clustering to eliminate outliers)

Step 1: Convex-hull based palette extraction



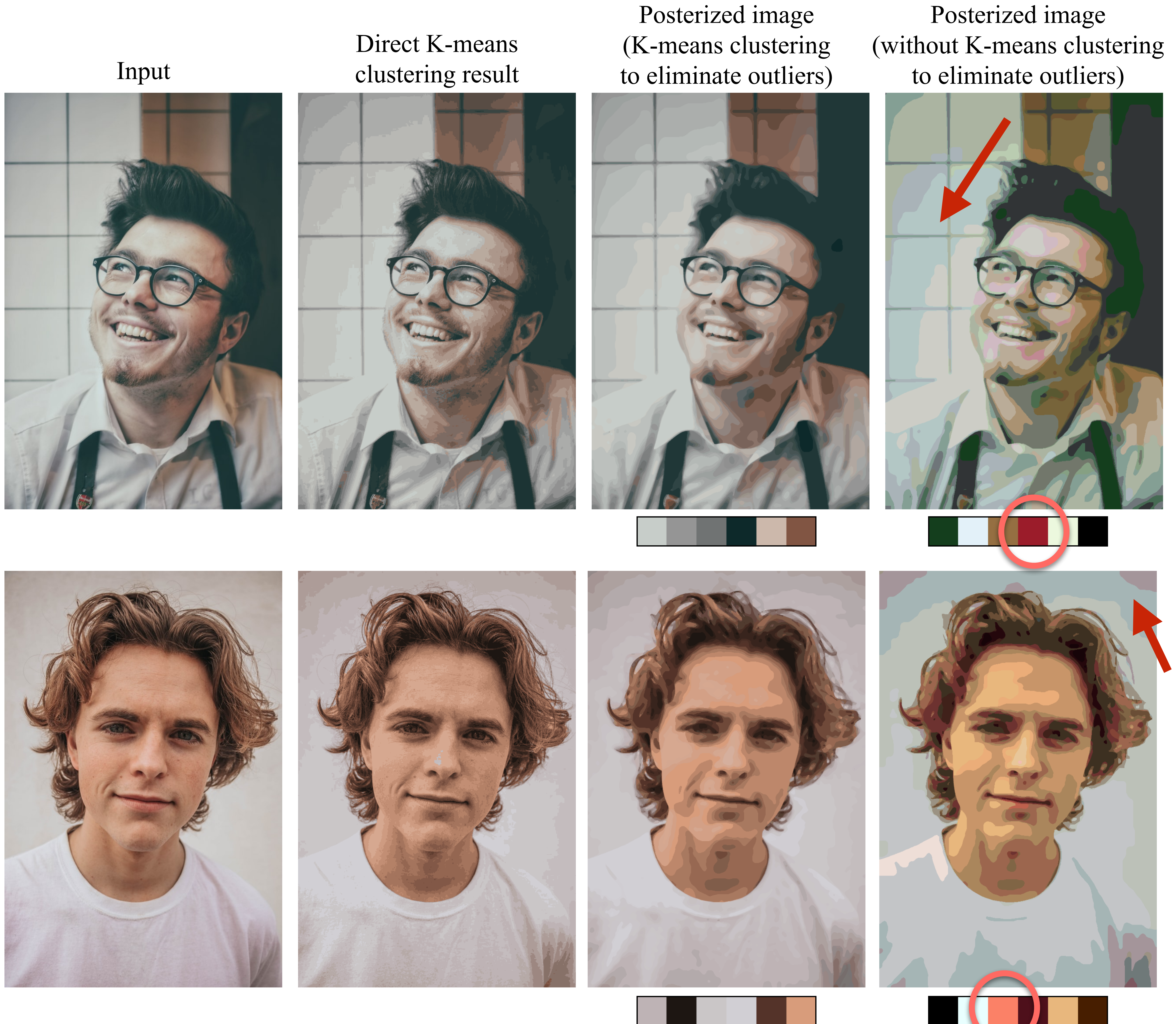
Input

Direct K-means clustering result

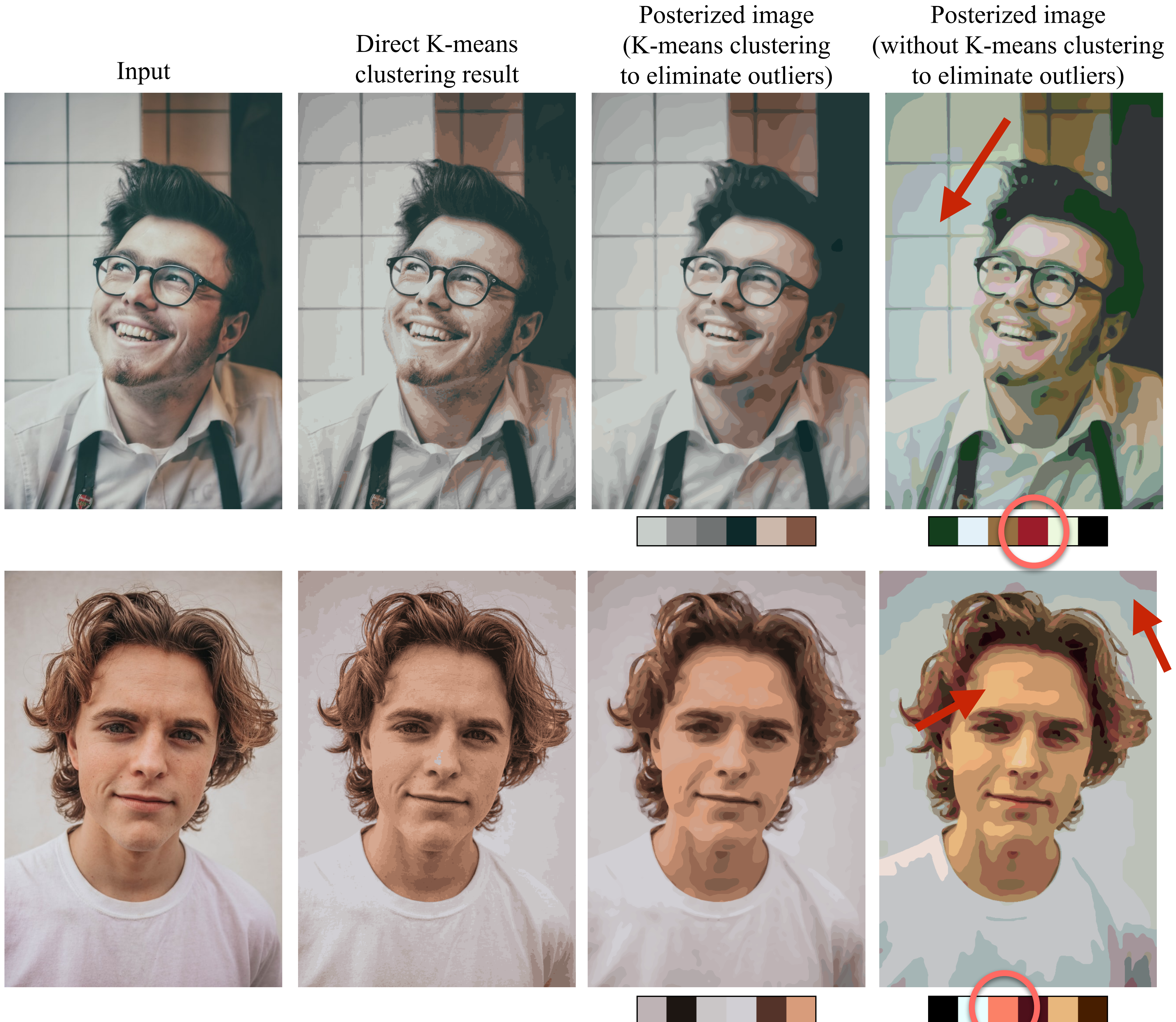
Posterized image (K-means clustering to eliminate outliers)

Posterized image (without K-means clustering to eliminate outliers)

Step 1: Convex-hull based palette extraction

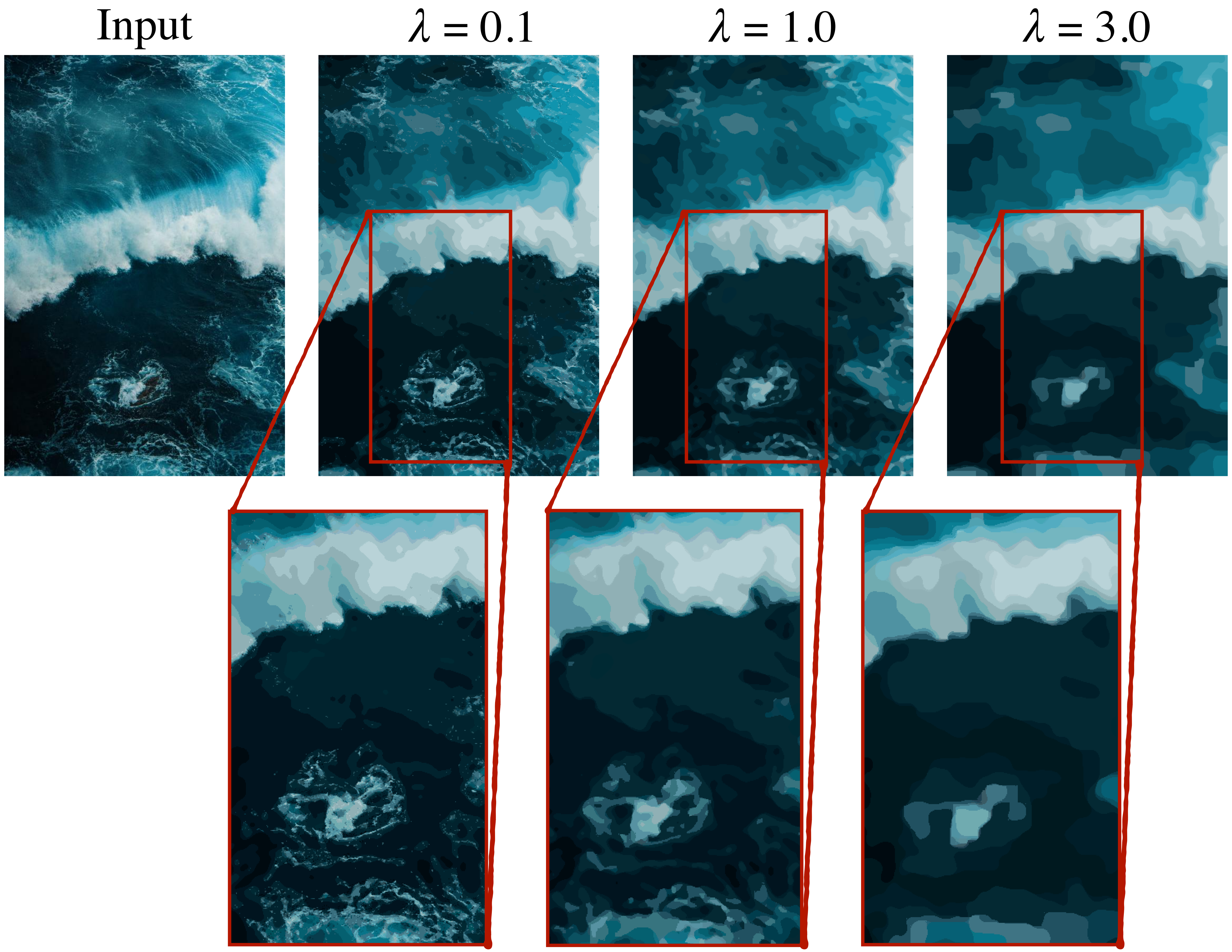

Input
Direct K-means clustering result
Posterized image (K-means clustering to eliminate outliers)
Posterized image (without K-means clustering to eliminate outliers)

Step 1: Convex-hull based palette extraction

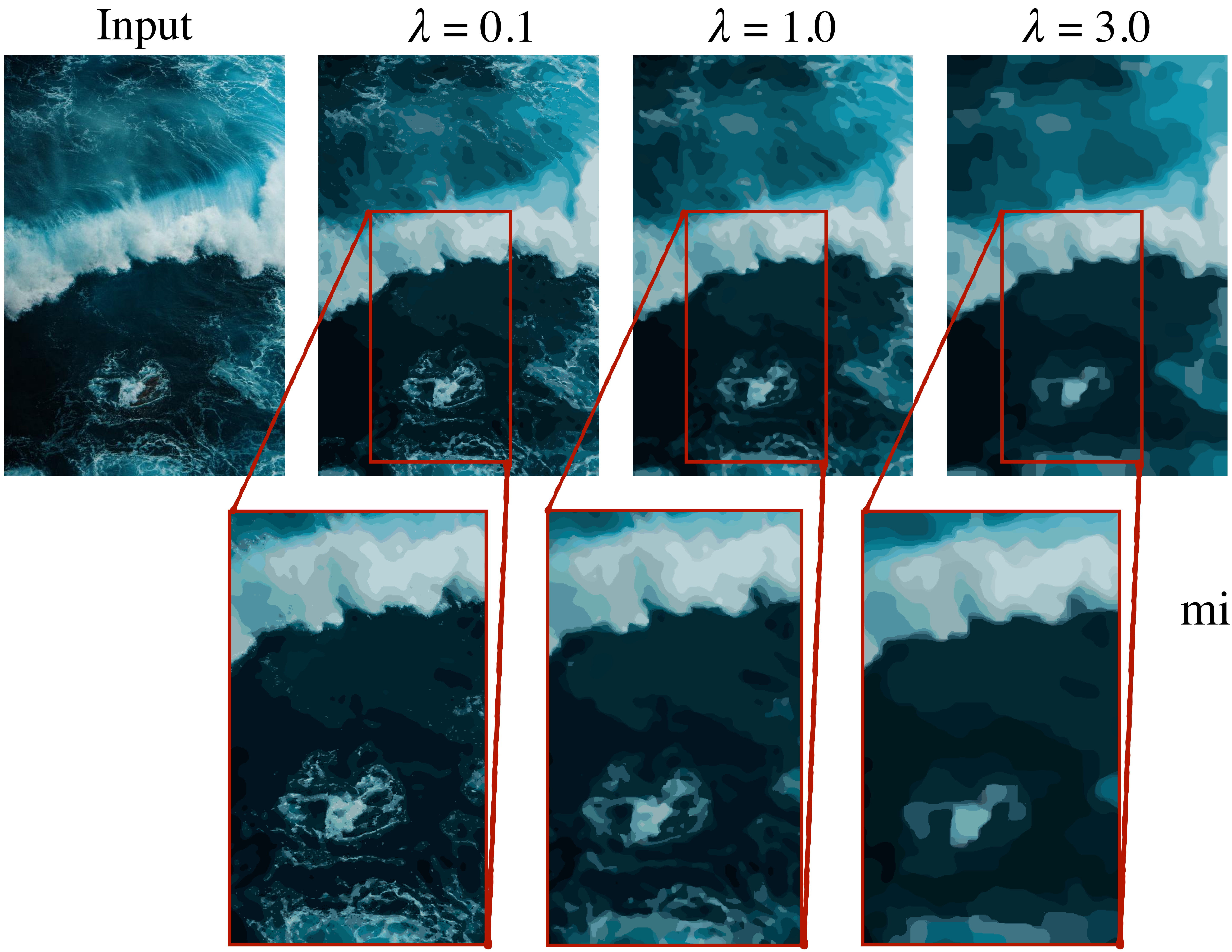

Input
Direct K-means clustering result
Posterized image (K-means clustering to eliminate outliers)
Posterized image (without K-means clustering to eliminate outliers)

Step 2: Rough region and color assignment

Step 2: Rough region and color assignment

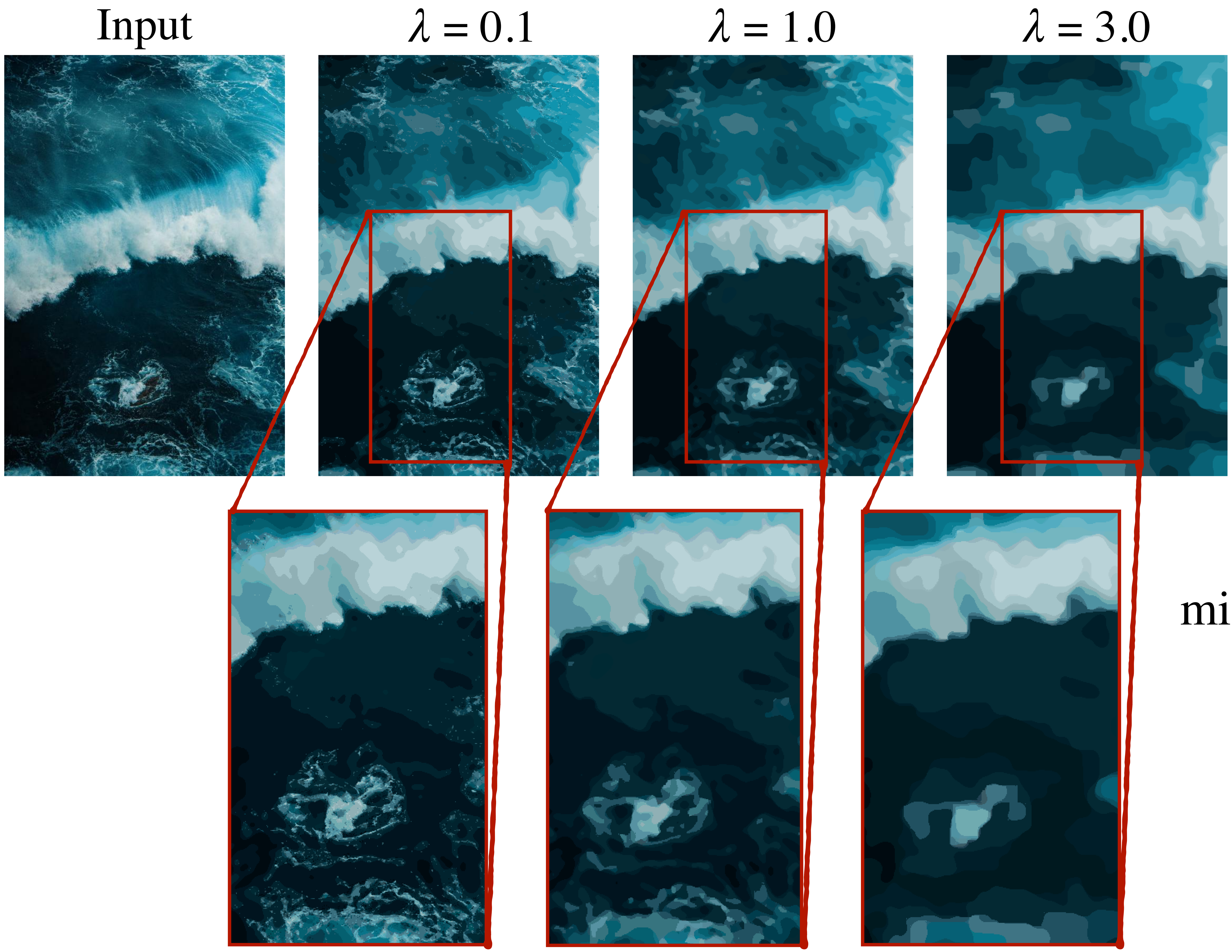


Step 2: Rough region and color assignment



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

Step 2: Rough region and color assignment



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