Pigmento: Pigment-Based Image Analysis and Editing

Jianchao Tan        George Mason University
Stephen DiVerdi    Adobe Research
Jingwan Lu          Adobe Research
Yotam Gingold      George Mason University
Background: Physical Painting
Background: Physical Painting
Background: Kubelka-Munk Model

Cyan pigment ground truth data.
33 wavelength, from 380 to 700 nm, every 10 nm.
Background: Kubelka-Munk Model

Cyan pigment ground truth data.
33 wavelength, from 380 to 700 nm, every 10 nm.

Absorption Curve (a)
Scattering Curve (s)
Substrate Reflectance ($\xi$)
Kubelka-Munk model (KM)
Reflectance Curve (r)
Background: Kubelka-Munk Model

Cyan pigment ground truth data.
33 wavelength, from 380 to 700 nm, every 10 nm.

Absorption Curve (a)
Scattering Curve (s)
Thickness (t)
Substrate Reflectance (ξ)
Kubelka-Munk model (KM)
Background: Kubelka-Munk Model

Cyan pigment ground truth data. 33 wavelength, from 380 to 700 nm, every 10 nm.

Absorption Curve (a)

Scattering Curve (s)

Thickness (t)

Substrate Reflectance ($\xi$)

Kubelka-Munk model (KM)

Reflectance Curve (r)

KM model $r = km(a, s, t, \xi)$

Illuminant

XYZ to RGB Transform

Gamma Correction

RGB

Color Matching Function
Background: Kubelka-Munk Model

Cyan pigment ground truth data.
33 wavelength, from 380 to 700 nm, every 10 nm.

Absorption Curve (a)
Scattering Curve (s)
Thickness (t)
Substrate Reflectance ($\xi$)

Kubelka-Munk model (KM)

Reflectance Curve (r)

KM model $r = km(a, s, t, \xi)$

Illuminant

XYZ to RGB Transform
Gamma Correction

RGB

Color Rendering $\phi: \mathbb{R}^L \rightarrow \mathbb{R}^3$
Background: Mixing multispectral pigments

$$I_{RGB} = \phi(km(a, s, t = 1, \xi = 1))$$
Background: Mixing multispectral pigments

\[ I_{RGB} = \phi(km(a, s, t = 1, \xi = 1)) \]

\[ a_{mix} = \sum w_i a_i \]
\[ s_{mix} = \sum w_i s_i \]
Background: Mixing multispectral pigments

\[ I_{RGB} = \phi(km(a, s, t = 1, \xi = 1)) \]

\[ a_{mix} = \sum w_ia_i \]
\[ s_{mix} = \sum w_is_i \]
Background: Mixing multispectral pigments

\[ I_{RGB} = \phi(km(a, s, t = 1, \xi = 1)) \]

\[ a_{mix} = \sum w_i a_i \]
\[ s_{mix} = \sum w_i s_i \]
Background: Mixing multispectral pigments

\[ I_{RGB} = \phi(km(a, s, t = 1, \xi = 1)) \]

\[ a_{mix} = \sum w_i a_i \]

\[ s_{mix} = \sum w_i s_i \]
Background: Kubelka-Munk Mixing Model

- Multispectral KM Mixing
- Linear RGB Mixing
Motivation

Painting re-editing

Input
Motivation

- Scattering
- Decomposition
- Absorption

Primary pigments

Mixing weights

Painting re-editing

Input
Related Work

• Digital palette based editing.
  

Decomposing Images into Layers via RGB-space Geometry (Tan et al. 2016)
Related Work

• Kubelka-Munk model based editing.

• Curtis et al. 1997; IMPaSTo (Baxter et al. 2004); Okumura et al. 2005; Zhao et al. 2008; RealPigment (Lu et al. 2014); Abed et al. 2014; Tan et al. 2015; Aharoni-Mack et al. 2017

Pigment-Based Recoloring of Watercolor Paintings (Aharoni-Mack et al. 2017)
Problem Statement

Input: Image pixels’ RGB colors: $I$. 
Problem Statement

**Input:** Image pixels’ RGB colors: $\mathbf{I}$.

**Output:** Primary multispectral pigments: $\mathbf{H}=[\mathbf{A}|\mathbf{S}]$. Their per-pixel mixing weights: $\mathbf{W}$.  

![Image of RGB colors and pixel mixing weights]
Problem Statement

Input:  Image pixels’ RGB colors: $I$.


$$I = \phi(km(WH, t, \xi))$$
Problem Statement

Input: Image pixels’ RGB colors: $\mathbf{I}$.  

Output: Primary multispectral pigments: $\mathbf{H}=[\mathbf{A}|\mathbf{S}]$. Their per-pixel mixing weights: $\mathbf{W}$.  

$$\mathbf{I} = \phi(km(\mathbf{WH}, t, \xi = 1))$$
Problem Statement

Input: Image pixels’ RGB colors: \( \mathbf{I} \).

Output: Primary multispectral pigments: \( \mathbf{H} = [\mathbf{A} | \mathbf{S}] \). Their per-pixel mixing weights: \( \mathbf{W} \).

\[
\mathbf{I} = \phi(\mathbf{km}(\mathbf{WH}, t = 1, \xi = 1))
\]
Problem Statement

Input: Image pixels’ RGB colors: $\mathbf{I}$.

Output: Primary multispectral pigments: $\mathbf{H} = [\mathbf{A} | \mathbf{S}]$. Their per-pixel mixing weights: $\mathbf{W}$.

$$\mathbf{I} = \phi(km(\mathbf{WH}))$$
Problem Statement

**Input:** Image pixels’ RGB colors: \( \mathbf{I} \).

**Output:** Primary multispectral pigments: \( \mathbf{H} = [\mathbf{A} | \mathbf{S}] \). Their per-pixel mixing weights: \( \mathbf{W} \).

\[
\| \mathbf{I} - \phi(km(\mathbf{WH})) \|^2
\]
Problem Statement

Input: Image pixels’ RGB colors: $I$.


\[
||I - \phi(km(WH))||^2
\]

It is under-constrained, and there are two additional challenges!
Challenge 1: Metamerism

Metamerism is a Big Effect

Color matching is an important illusion that is understood quantitatively
Solution: Smoothness Regularization
Solution: Smoothness Regularization

Absorption and Scattering curve of each primary pigment should be smooth.
Solution: Smoothness Regularization

Absorption and Scattering curve of each primary pigment should be smooth.

Absorption and Scattering’s division curve should also be smooth.
Solution: Smoothness Regularization

Absorption and Scattering curve of each primary pigment should be smooth.

Absorption and Scattering’s division curve should also be smooth.

Useful for Metamerism problem!
Challenge 2: Solution Space

- Gamut H for 4 color points
- Gamut H1 by scaling H
- Gamut H2 by rotating H
Challenge 2: Solution Space

Gamut H for 4 color points
Gamut H1 by scaling H
Gamut H2 by rotating H

Gamut Q for more points
Gamut Q1 by scaling Q
Gamut Q2 by rotating Q
Good Initial values
Good Initial values
Good Initial values
Divide into two subproblems

Directly solving this problem is hard.
Divide into two subproblems

Directly solving this problem is hard.

We divide it into two subproblems:
Divide into two subproblems

Directly solving this problem is hard.

We divide it into two subproblems:

1. Primary pigments extraction
Divide into two subproblems

Directly solving this problem is hard.

We divide it into two subproblems:

1. Primary pigments extraction

2. Mixing weights extraction
Pigments Extraction
Pigments Extraction

Tan et al. 2016
Pigments Extraction

Tan et al. 2016

Okumura et al. 2005

Multispectral Initial H0
Pigments Extraction

Tan et al. 2016

Convex hull vertices

Okumura et al. 2005

Multispectral Initial H0

Representative pixels
Pigments Extraction

\[ \| I - \phi(km(WH)) \|^2 \]

- Fix H, solve W
- Fix W, solve H

+ Pigment Smoothness

RGB

Okumura et al. 2005

Multispectral Initial H0

Convex hull vertices

Representative pixels

ANLS

Tan et al. 2016
Pigments Extraction

\[ \|I - \phi(km(WH))\|^2 + \text{Pigment Smoothness} \]

Fix H, solve W
Fix W, solve H

ANLS

Multispectral Final \( H^* \)

Representative pixels

\[ \text{Convex hull vertices} \]

Tan et al. 2016

Okumura et al. 2005

RGB

Multispectral Initial \( H_0 \)

Multispectral

Final \( H^* \)
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

**Smoothness**: Each primary pigment’s mixing weights map is spatially smooth.
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

**Smoothness**: Each primary pigment’s mixing weights map is spatially smooth

**Sparsity**: Each pixel’s color is a mixing of smallest subset of primary pigments
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

**Smoothness**: Each primary pigment’s mixing weights map is spatially smooth

**Sparsity**: Each pixel’s color is a mixing of smallest subset of primary pigments
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

**Smoothness**: Each primary pigment’s mixing weights map is spatially smooth

**Sparsity**: Each pixel’s color is a mixing of smallest subset of primary pigments
Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

**Smoothness**: Each primary pigment’s mixing weights map is spatially smooth.

**Sparsity**: Each pixel’s color is a mixing of smallest subset of primary pigments.
Our results
Compare to results from other models

Tan et al. 2016
4 pigments
RMSE: 23.4

Ours
4 pigments
RMSE: 5.2

Tan et al. 2016
4 pigments
RMSE: 4.7

Aksoy et al. 2017
7 layers
RMSE: ~0

Chang et al. 2015
4

Tan et al. 2016
6 pigments
RMSE: 4.5

Aksoy et al. 2017
6

Chang et al. 2015
5

Tan et al. 2016
5

Chang et al. 2015
7
Recoloring comparison

Original  blue pigment -> green (ours)  blue RGB -> green (Tan2016)

Original  red pigment -> blue (ours)  red RGB -> blue (Tan2016)
Recoloring comparison

Ours

Tan et al. 2016

Chang et al. 2015

original
Applications
Recoloring by modifying pigment weights
Modify weights of black/white pigment

- Increase the mixing weight of black pigment
- Decrease brightness

- Increase all weights
- Increase the mixing weight of white pigment
- Increase brightness
Modify pigment scattering parameters

- Original
- Increase scattering
- Decrease scattering
Mask Selection

Rectangle Input
Grabcut on KM layer
Grabcut on RGB
Copy-Paste in pigment space
Palette Summarization - Photos

 ours
 Tan2016
 Color CC
 Chang2015
Edge detection and enhancement

- on weights map
- on RGB
- original
- Enhancement
Conclusion
Conclusion

• Provide an efficient optimization framework to extract multispectral pigments and their per-pixel mixing weights from given RGB painting image.
Conclusion

• Provide an efficient optimization framework to extract multispectral pigments and their per-pixel mixing weights from given RGB painting image.

• Enable many paint-like edits of the painting, which are beyond RGB space.
Conclusion

• Provide an efficient optimization framework to extract multispectral pigments and their per-pixel mixing weights from given RGB painting image.

• Enable many paint-like edits of the painting, which are beyond RGB space.

• Our discussion of gamut problem and several regularization terms used in our optimization are useful in other similar problems.
Limitation and future work
Limitation and future work

- Using prior acyclic pigment database as initial value may cause overfitting problem.
Limitation and future work

• Using prior acyclic pigment database as initial value may cause overfitting problem.
  • We do not have other datasets (e.g. watercolor pigment) to verify it.
Limitation and future work

- Using prior acyclic pigment database as initial value may cause overfitting problem.
  - We do not have other datasets (e.g. watercolor pigment) to verify it.
- We assume constant paint thickness to simplify optimization.
Limitation and future work

- Using prior acyclic pigment database as initial value may cause overfitting problem.
  - We do not have other datasets (e.g. watercolor pigment) to verify it.
- We assume constant paint thickness to simplify optimization.
- We may want to estimate pigment layers instead of just mixtures, then layer order is needed.
Limitation and future work

- Using prior acyclic pigment database as initial value may cause overfitting problem.
  - We do not have other datasets (e.g. watercolor pigment) to verify it.
- We assume constant paint thickness to simplify optimization.
- We may want to estimate pigment layers instead of just mixtures, then layer order is needed.
- Use our decomposition results to help extract brushstroke-level structure from painting images.
Limitation and future work

- Using prior acyclic pigment database as initial value may cause overfitting problem.
  - We do not have other datasets (e.g. watercolor pigment) to verify it.
- We assume constant paint thickness to simplify optimization.
- We may want to estimate pigment layers instead of just mixtures, then layer order is needed.
- Use our decomposition results to help extract brushstroke-level structure from painting images.
Thank You!

• **Contact Information:**
  • Jianchao Tan: jtan8@gmu.edu
  • Stephen DiVerdi: diverdi@adobe.com
  • Jingwan Lu: jlu@adobe.com
  • Yotam Gingold: ygingold@gmu.edu

• **Project Website:** [https://cragl.cs.gmu.edu/pigmento/](https://cragl.cs.gmu.edu/pigmento/)


• **Artists:**
  • MontMarteArt, Jan Ironside, Graham Gercken, Nel Jansen, Cathleen Rehfeld, Patty Baker, John Larriva, Pamela Gatens, Mark Adam Webster, Patti Mollica, Jan Ironside.

• **Sponsors:**
  • United States National Science Foundation, Adobe Research.
Extra Slides
## Performance Information

<table>
<thead>
<tr>
<th>Examples</th>
<th>Image size</th>
<th>Pigments number</th>
<th>CPU</th>
<th>KM primary pigments extraction Time (sec)</th>
<th>KM mixing weights extraction Time (sec)</th>
<th>KM original image reconstruction RMSE (0-255)</th>
</tr>
</thead>
<tbody>
<tr>
<td>soleil</td>
<td>600*467</td>
<td>6</td>
<td>core i7</td>
<td>35</td>
<td>155</td>
<td>1.9</td>
</tr>
<tr>
<td>autumn</td>
<td>600*458</td>
<td>5</td>
<td>xeon</td>
<td>16</td>
<td>225</td>
<td>6.0</td>
</tr>
<tr>
<td>four_colors_2</td>
<td>600*598</td>
<td>4</td>
<td>core i7</td>
<td>9</td>
<td>211</td>
<td>5.2</td>
</tr>
<tr>
<td>Impasto_flower2</td>
<td>595*600</td>
<td>6</td>
<td>xeon</td>
<td>44</td>
<td>615</td>
<td>5.1</td>
</tr>
<tr>
<td>Landscape4</td>
<td>600*479</td>
<td>5</td>
<td>xeon</td>
<td>26</td>
<td>256</td>
<td>4.7</td>
</tr>
<tr>
<td>Portrait2</td>
<td>600*441</td>
<td>6</td>
<td>xeon</td>
<td>29</td>
<td>243</td>
<td>4.4</td>
</tr>
<tr>
<td>tree</td>
<td>600*492</td>
<td>4</td>
<td>core i7</td>
<td>14</td>
<td>151</td>
<td>4.0</td>
</tr>
</tbody>
</table>
Pigment smoothness and thickness

- **Ground truth**
- **Synthetic image**
- **Thickness per-pixel**
- **Pigments 1 & 2**

**Reflectance**

**Absorption**

**Scattering**

**Absorption Scattering**

- **Constant thickness with a/s smoothness**
  - RGB RMSE: 1.24

- **Constant thickness w/o a/s smoothness**
  - RGB RMSE: 1.39

- **Varying thickness with a/s smoothness**
  - RGB RMSE: 0.68

- **Varying thickness w/o a/s smoothness**
  - RGB RMSE: 0.58

Wavelength
Pigment smoothness and thickness

- **Ground truth**
- **Reflectance**
- **Absorption**
- **Scattering**

**Constant thickness with a/s smoothness**
RGB RMSE: 1.24

**Varying thickness with a/s smoothness**
RGB RMSE: 0.68

**Constant thickness w/o a/s smoothness**
RGB RMSE: 1.39

**Varying thickness w/o a/s smoothness**
RGB RMSE: 0.58
Pigment number influence
Wavelength influence
Wavelength influence

8 wavelength recovery   original   3 wavelength recovery

1.9   soleil   6.5

6.0   autumn   11.0

4.4   portrait2   8.5

4.7   landscape4   6.3

5.1   Impasto_flower4   7.3

4.0   tree   5.2

5.2   four_colors_2   8.1
Wavelength influence
Primary pigment estimation convergence

![Total energy over iterations for different images](image)
Primary pigment estimation convergence

Reconstruction error vs. Iterations for different datasets.
Compare to results from other models

Ours
4 pigments
RMSE: 4.0

Tan et al. 2016
4 pigments
RMSE: 10.1

Tan et al. 2016
6 pigments
RMSE: 4.5

Aksoy et al. 2017
6 layers
RMSE: ~0

Chang et al. 2015
4 pigments
RMSE: 5.2

Tan et al. 2016
6 pigments
RMSE: 4.7

Aksoy et al. 2017
7 layers
RMSE: ~0

Chang et al. 2015
5 pigments
RMSE: 10.1
Aksoy et al. 2017 results
Ground Truth Test
### Ground truth test information

<table>
<thead>
<tr>
<th>Experiments</th>
<th>RMSE for recovering pigments parameters H (A / S)</th>
<th>RMSE for recovering pigments Reflectance R</th>
<th>RMSE for weights recovering using recovered pigments</th>
<th>RMSE for weights recovering using ground truth pigments</th>
<th>RMSE for image recovering using recovered pigments</th>
<th>RMSE for image recovering using ground truth pigments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>6.2 / 1.2</td>
<td>0.3</td>
<td>29</td>
<td>15.2</td>
<td>4.8</td>
<td>5.9</td>
</tr>
<tr>
<td>Exp2</td>
<td>1.4 / 0.9</td>
<td>0.3</td>
<td>19.8</td>
<td>11.8</td>
<td>6.8</td>
<td>4.3</td>
</tr>
<tr>
<td>Exp3</td>
<td>4.5 / 0.5</td>
<td>0.7</td>
<td>63</td>
<td>21.4</td>
<td>6.7</td>
<td>5.9</td>
</tr>
<tr>
<td>Exp4</td>
<td>7.1 / 1.2</td>
<td>0.6</td>
<td>42.3</td>
<td>14.1</td>
<td>8.5</td>
<td>6</td>
</tr>
<tr>
<td>Exp5</td>
<td>1.0 / 0.7</td>
<td>0.3</td>
<td>16.6</td>
<td>10.4</td>
<td>5.8</td>
<td>5.2</td>
</tr>
<tr>
<td>Mean</td>
<td>4.0 / 0.9</td>
<td>0.4</td>
<td>34.14</td>
<td>14.58</td>
<td>6.52</td>
<td>5.46</td>
</tr>
<tr>
<td>Std</td>
<td>2.7 / 0.3</td>
<td>0.2</td>
<td>18.97</td>
<td>4.25</td>
<td>1.37</td>
<td>0.72</td>
</tr>
</tbody>
</table>
Kubelka-Munk Layer Model

\[ R = R_1 + \frac{T_1^2 R_2}{1 - R_1 R_2} \]

\( R \) is Reflectance

\( T \) is Transmittance