Pigmento: Pigment-Based Image Analysis and Editing

Jianchao Tan George Mason University

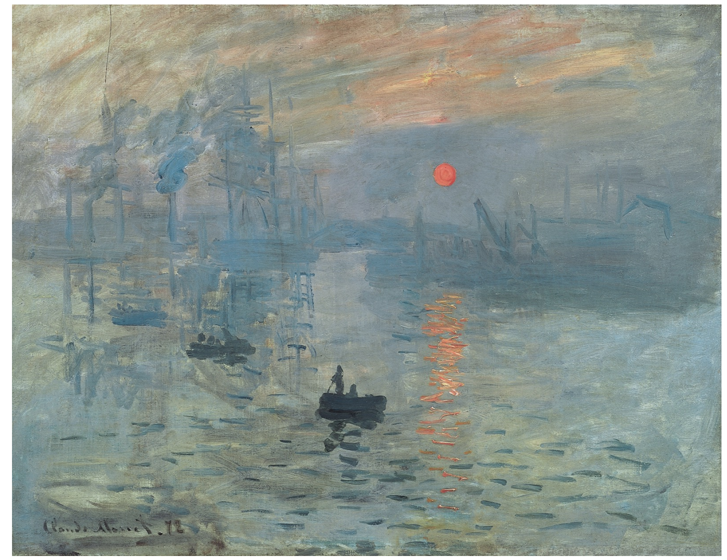
Stephen DiVerdi Adobe Research

Jingwan Lu Adobe Research

Yotam Gingold George Mason University

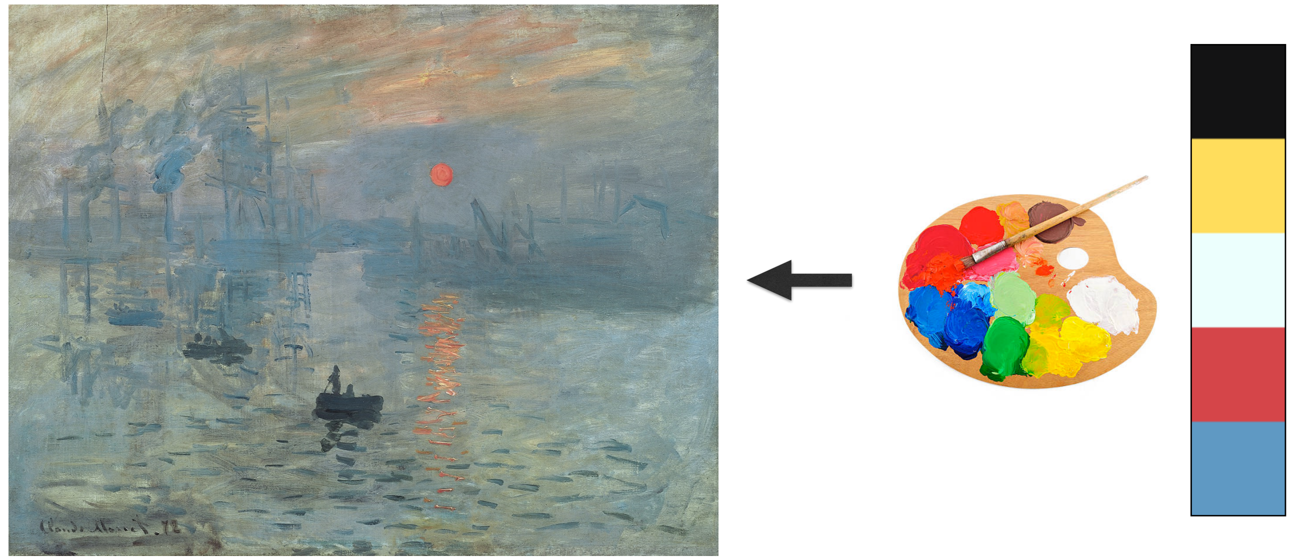
Good afternoon everyone, I am Jianchao Tan, a Ph.D. student in George Mason University Computer Science department. This project is about pigment based image analysis and editing. This work is partially done in my adobe research summer internship.

Background: Physical Painting



Physical paintings are created using real word brushes with some primary pigments, such as oil pigment, watercolor pigment or acrylic pigments. The pigments here are not RGB values used in digital software. They are not mixed with linear blending. These pigments' color perceived by us are the result of a complicated physical model over illuminant, pigment parameters and human eye response.

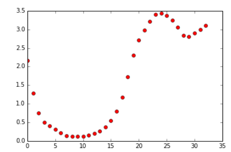
Background: Physical Painting



Physical paintings are created using real word brushes with some primary pigments, such as oil pigment, watercolor pigment or acrylic pigments. The pigments here are not RGB values used in digital software. They are not mixed with linear blending. These pigments’ color perceived by us are the result of a complicated physical model over illuminant, pigment parameters and human eye response.

Background: Kubelka-Munk Model

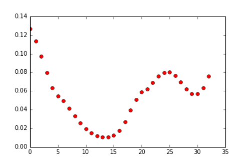
Cyan pigment ground truth data.
33 wavelength, from 380 to 700 nm, every 10 nm.



Absorption Curve (a)

Thickness (t)

Substrate Reflectance (ξ)



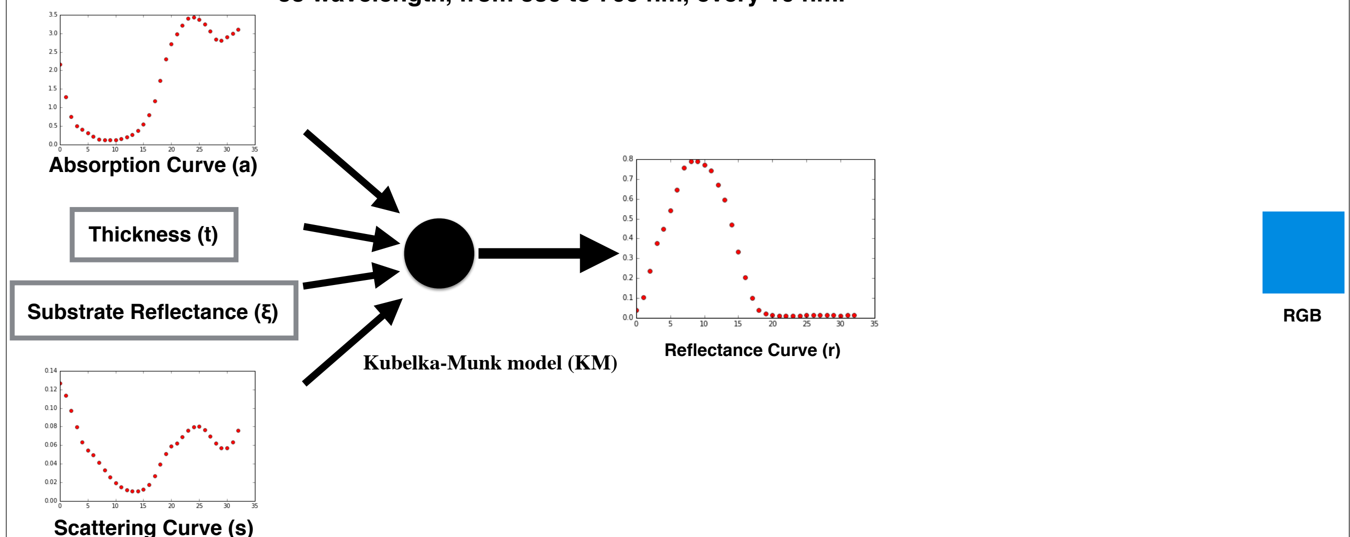
Scattering Curve (s)

RGB

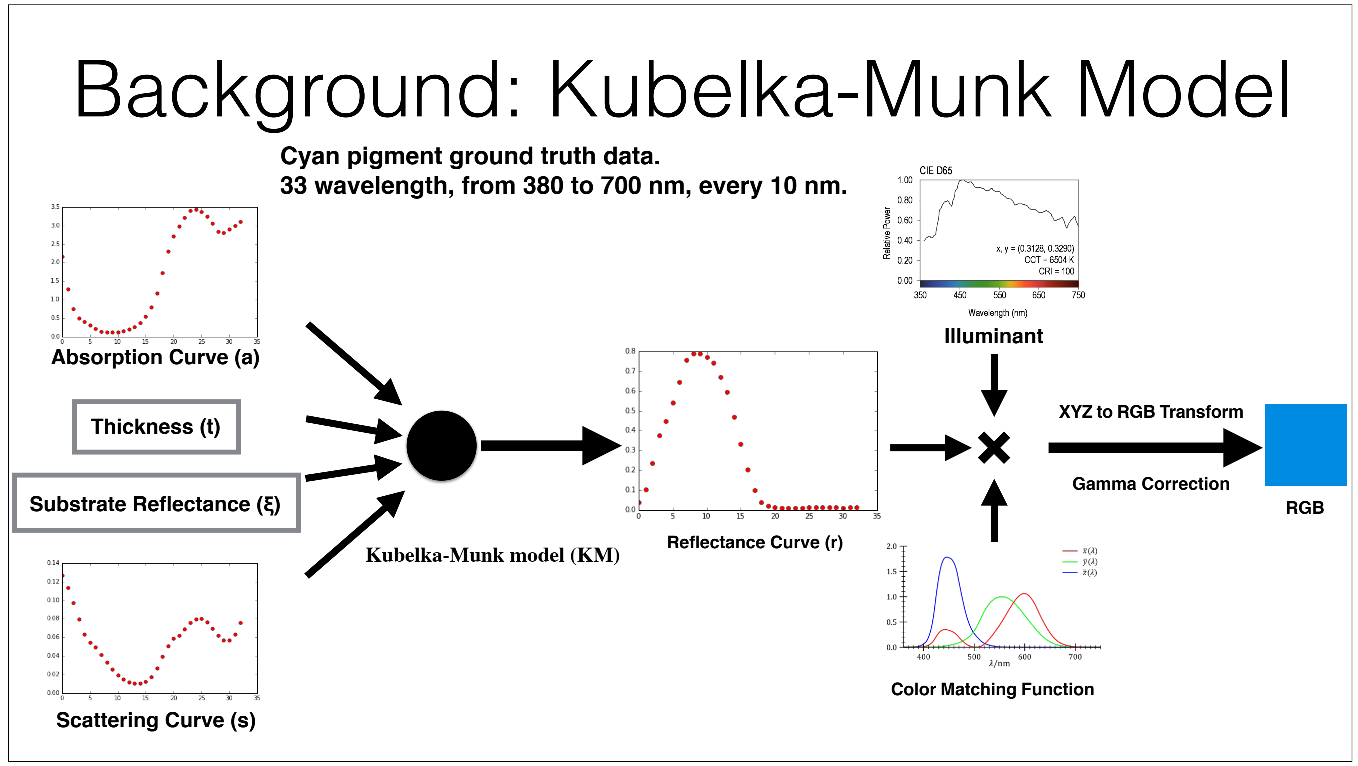
There is a physically inspired color compositing model called Kubelka-Munk model. This model takes pigment absorption, thickness, scattering and substrate reflectance as input to obtain pigment reflectance parameters, which can be feed into an standard imaging system to get final rendered RGB color of this pigment. The KM model and imaging system are both nonlinear transformations, represented by this km function and phi function. The whole color compositing system is nonlinear.

Background: Kubelka-Munk Model

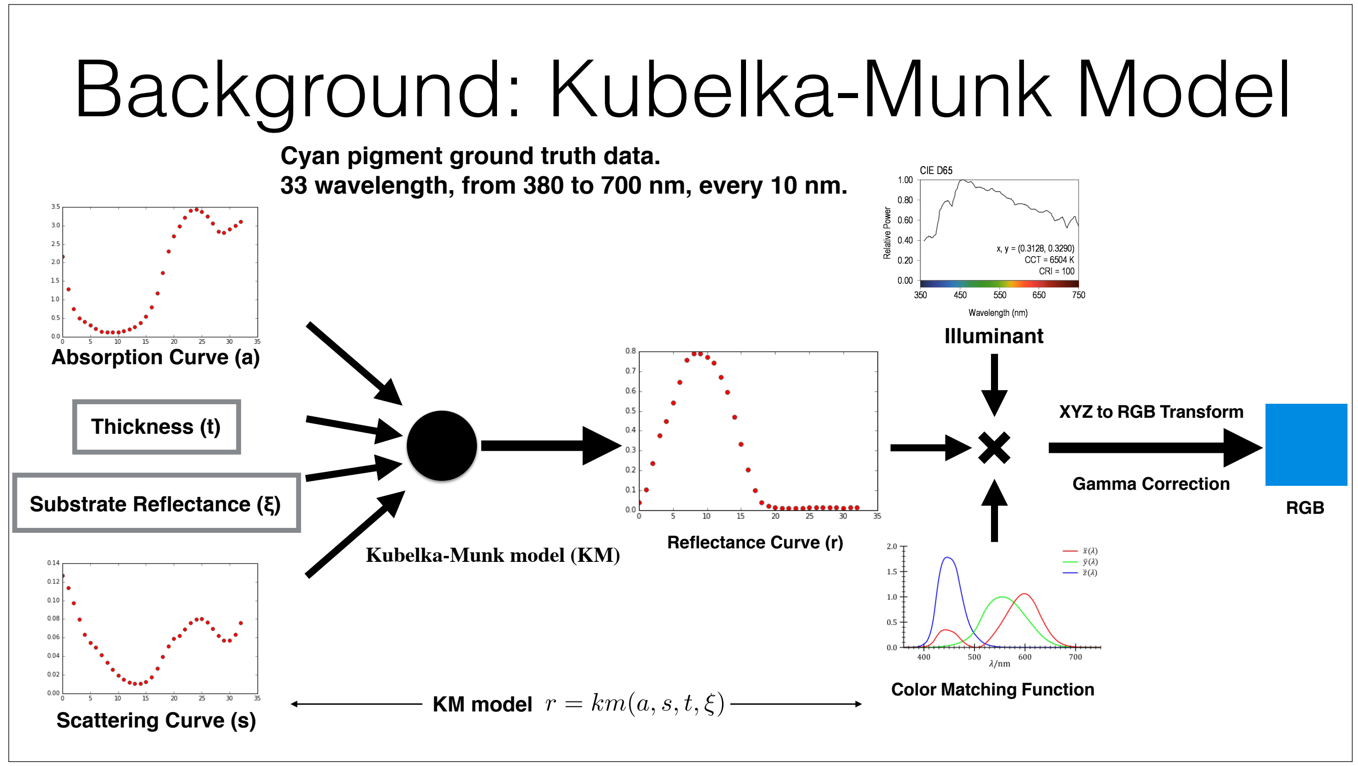
Cyan pigment ground truth data.
33 wavelength, from 380 to 700 nm, every 10 nm.



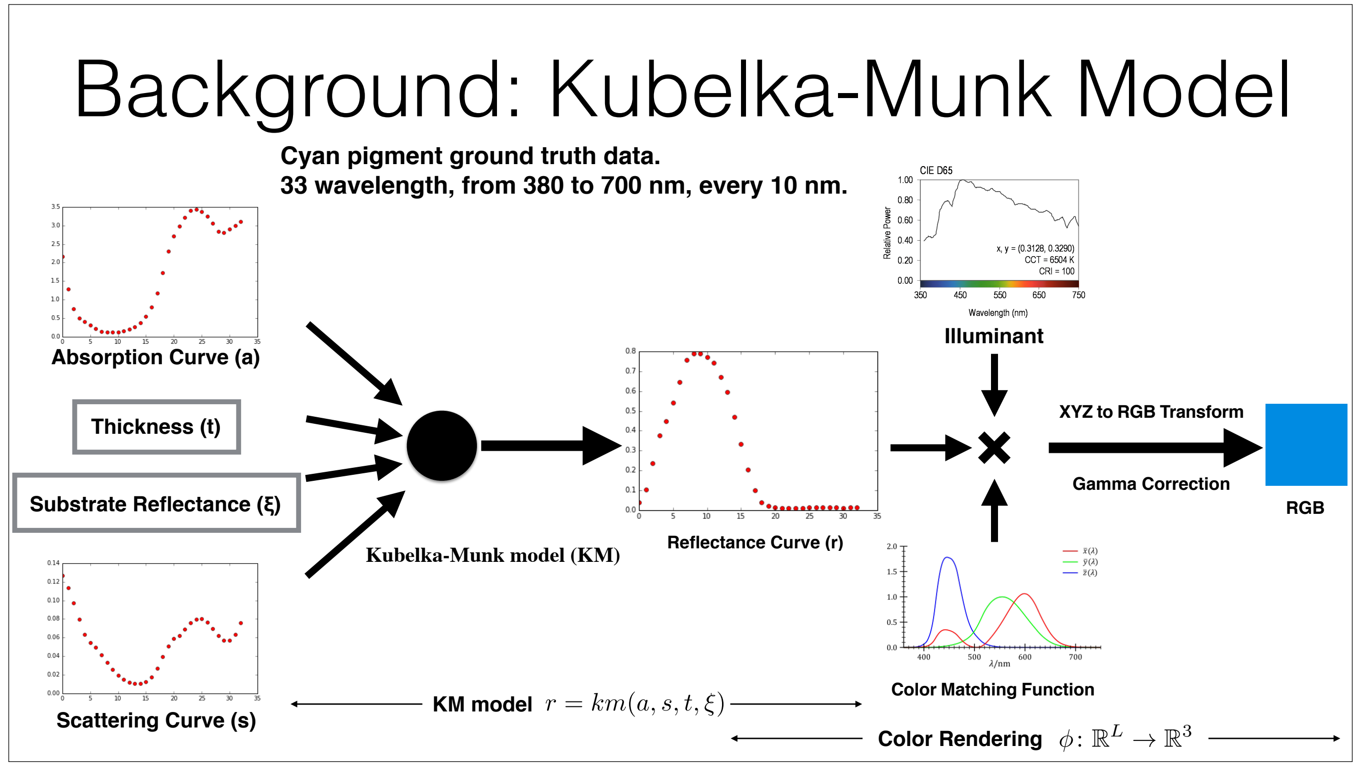
There is a physically inspired color compositing model called Kubelka-Munk model. This model takes pigment absorption, thickness, scattering and substrate reflectance as input to obtain pigment reflectance parameters, which can be feed into an standard imaging system to get final rendered RGB color of this pigment. The KM model and imaging system are both nonlinear transformations, represented by this km function and phi function. The whole color compositing system is nonlinear.



There is a physically inspired color compositing model called Kubelka-Munk model. This model takes pigment absorption, thickness, scattering and substrate reflectance as input to obtain pigment reflectance parameters, which can be feed into an standard imaging system to get final rendered RGB color of this pigment. The KM model and imaging system are both nonlinear transformations, represented by this km function and phi function. The whole color compositing system is nonlinear.



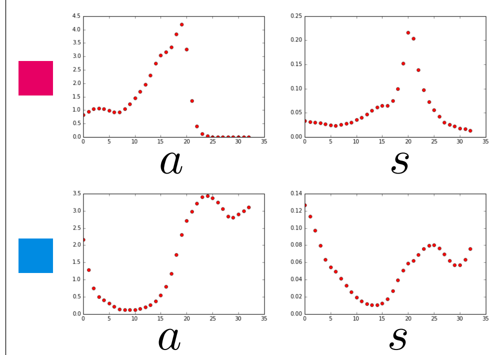
There is a physically inspired color compositing model called Kubelka-Munk model. This model takes pigment absorption, thickness, scattering and substrate reflectance as input to obtain pigment reflectance parameters, which can be feed into an standard imaging system to get final rendered RGB color of this pigment. The KM model and imaging system are both nonlinear transformations, represented by this km function and phi function. The whole color compositing system is nonlinear.



There is a physically inspired color compositing model called Kubelka-Munk model. This model takes pigment absorption, thickness, scattering and substrate reflectance as input to obtain pigment reflectance parameters, which can be feed into an standard imaging system to get final rendered RGB color of this pigment. The KM model and imaging system are both nonlinear transformations, represented by this km function and phi function. The whole color compositing system is nonlinear.

Background: Mixing multispectral pigments

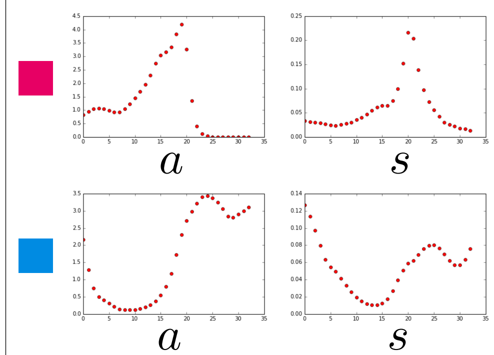
$$I_{RGB} = \phi(km(a, s, t = 1, \xi = 1))$$



When we mix two pigments, the mixing of two pigments' absorption and scatterings parameters is linear, however, the KM model is a nonlinear color compositing model. Here we use thickness is 1 and canvas reflectance is also 1 when rendering these RGB colors. canvas reflectance is 1 means that it is white.

Background: Mixing multispectral pigments

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



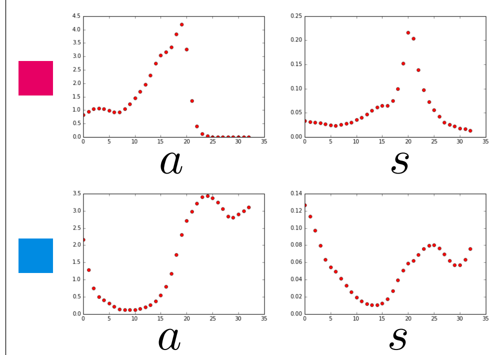
$$a_{mix} = \sum w_i a_i$$

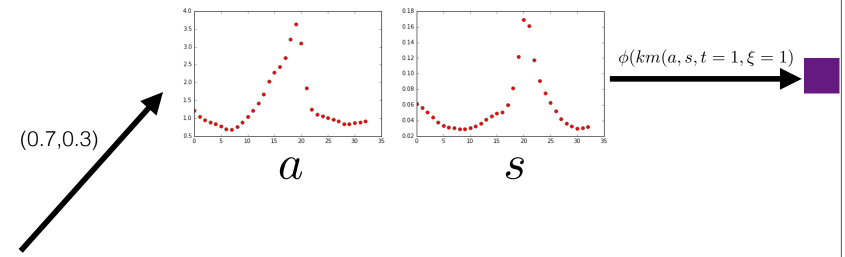
$$s_{mix} = \sum w_i s_i$$

When we mix two pigments, the mixing of two pigments’ absorption and scatterings parameters is linear, however, the KM model is a nonlinear color compositing model. Here we use thickness is 1 and canvas reflectance is also 1 when rendering these RGB colors. canvas reflectance is 1 means that it is white.

Background: Mixing multispectral pigments

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



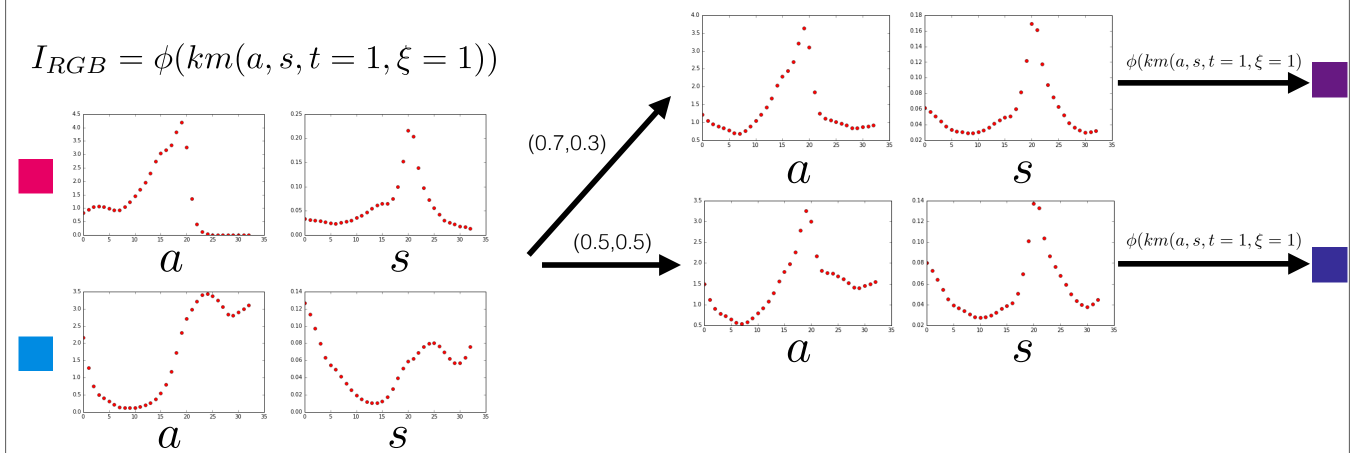


$$a_{mix} = \sum w_i a_i$$
$$s_{mix} = \sum w_i s_i$$

When we mix two pigments, the mixing of two pigments' absorption and scatterings parameters is linear, however, the KM model is a nonlinear color compositing model. Here we use thickness is 1 and canvas reflectance is also 1 when rendering these RGB colors. canvas reflectance is 1 means that it is white.

Background: Mixing multispectral pigments

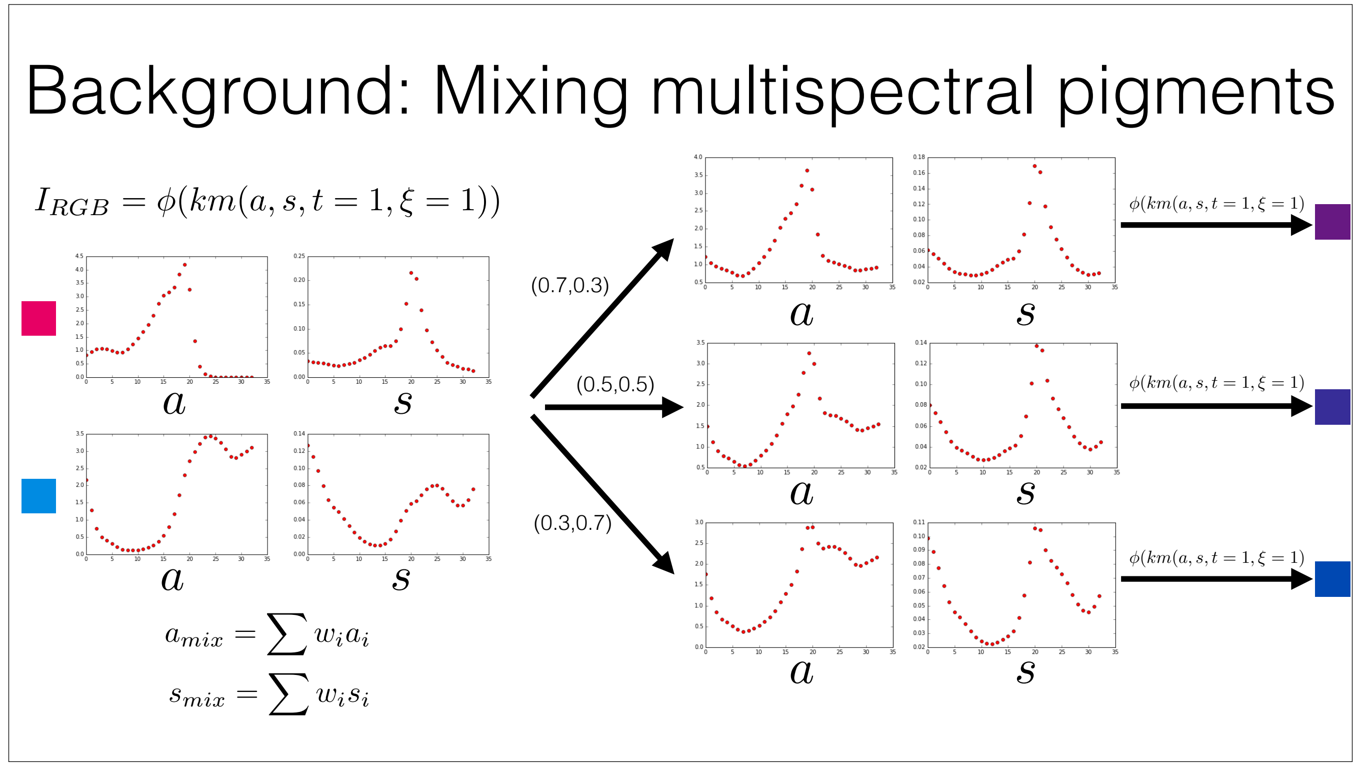
$$I_{RGB} = \phi(km(a, s, t = 1, \xi = 1))$$



$$a_{mix} = \sum w_i a_i$$

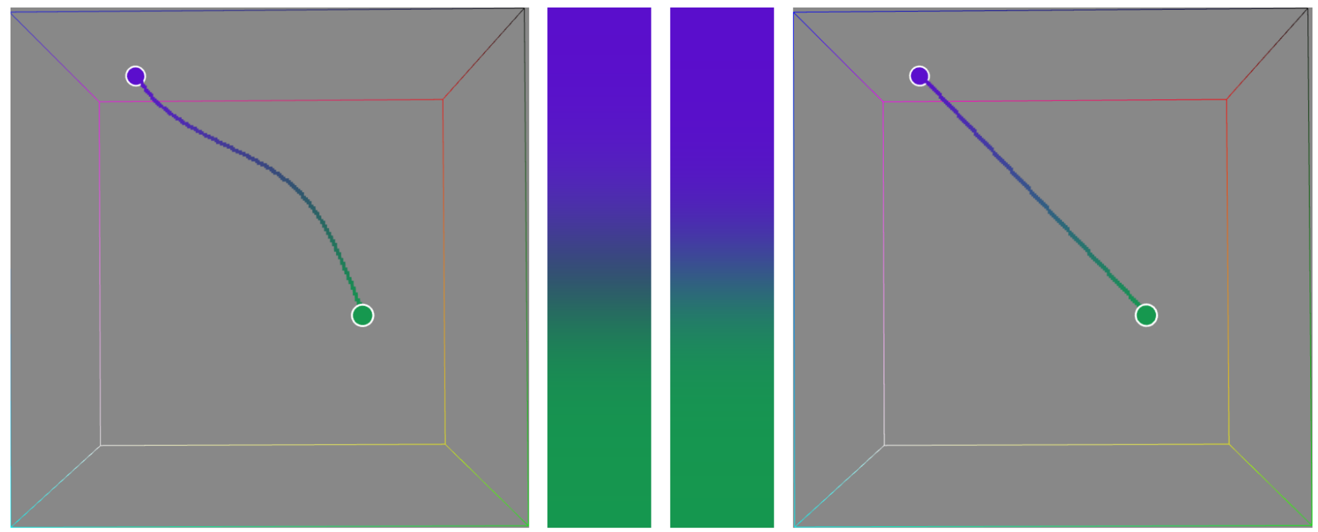
$$s_{mix} = \sum w_i s_i$$

When we mix two pigments, the mixing of two pigments' absorption and scatterings parameters is linear, however, the KM model is a nonlinear color compositing model. Here we use thickness is 1 and canvas reflectance is also 1 when rendering these RGB colors. canvas reflectance is 1 means that it is white.



When we mix two pigments, the mixing of two pigments' absorption and scatterings parameters is linear, however, the KM model is a nonlinear color compositing model. Here we use thickness is 1 and canvas reflectance is also 1 when rendering these RGB colors. canvas reflectance is 1 means that it is white.

Background: Kubelka-Munk Mixing Model

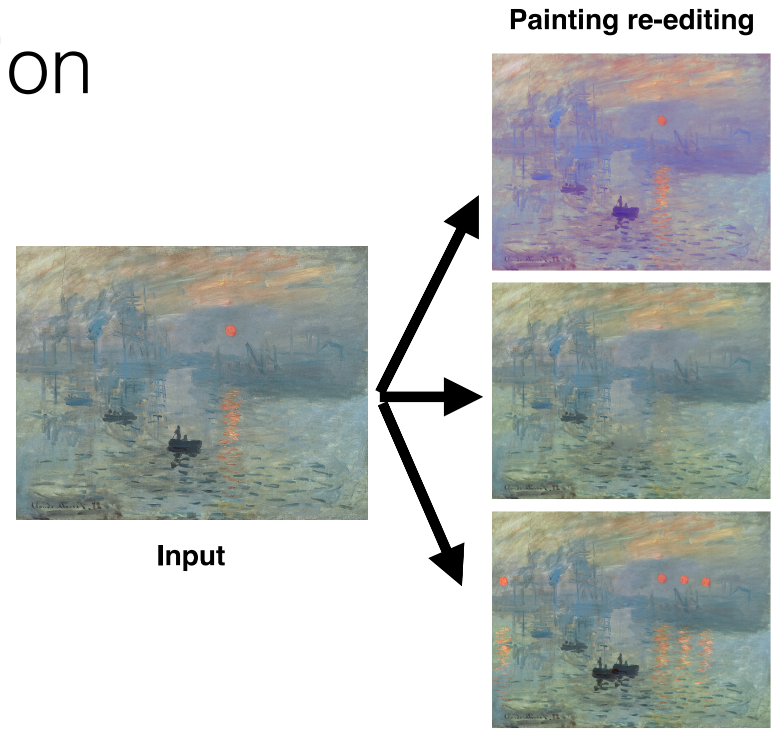


Multispectral KM Mixing

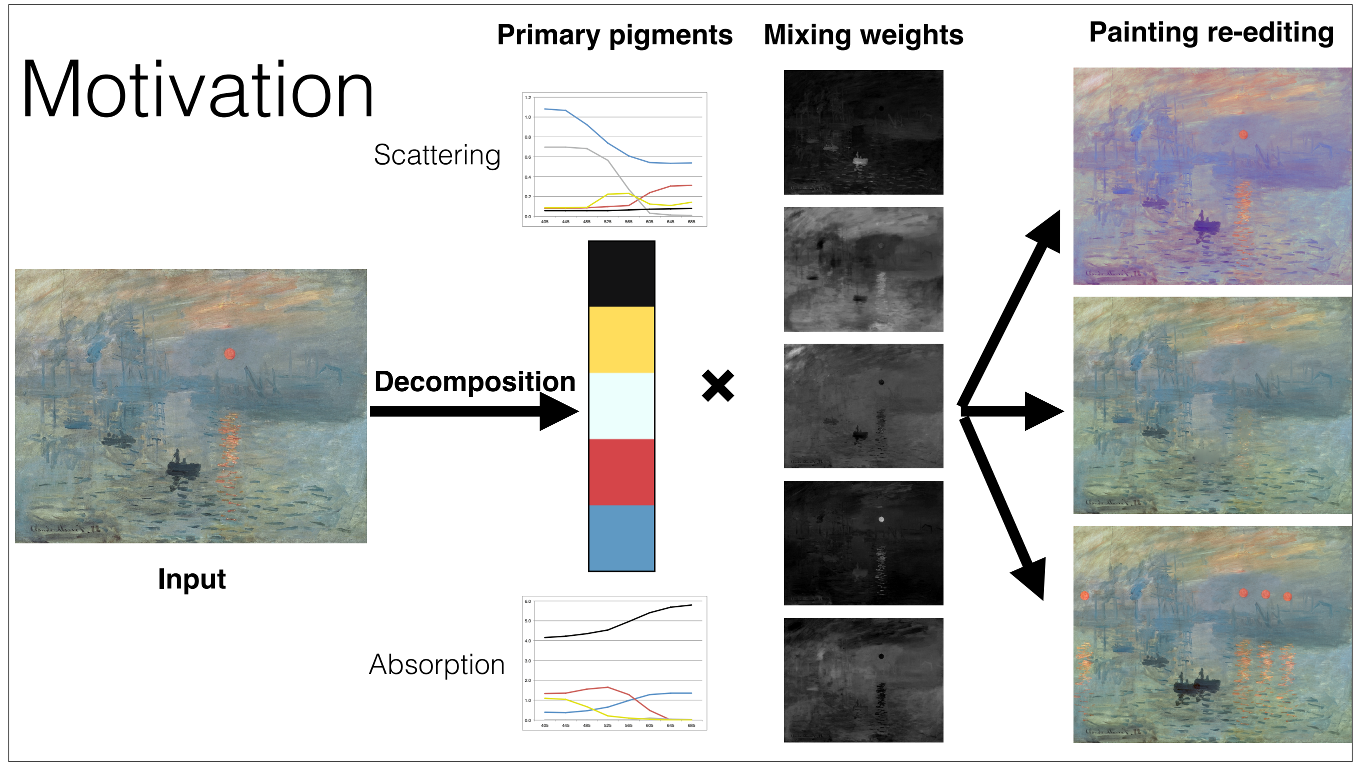
Linear RGB Mixing

When we visualize the continuous mixed RGB colors of two pigments, it is a curve in RGB space. This is different from linear mixing of two pigments' RGB colors using digital linear mixing model, which is just a straight line.

Motivation



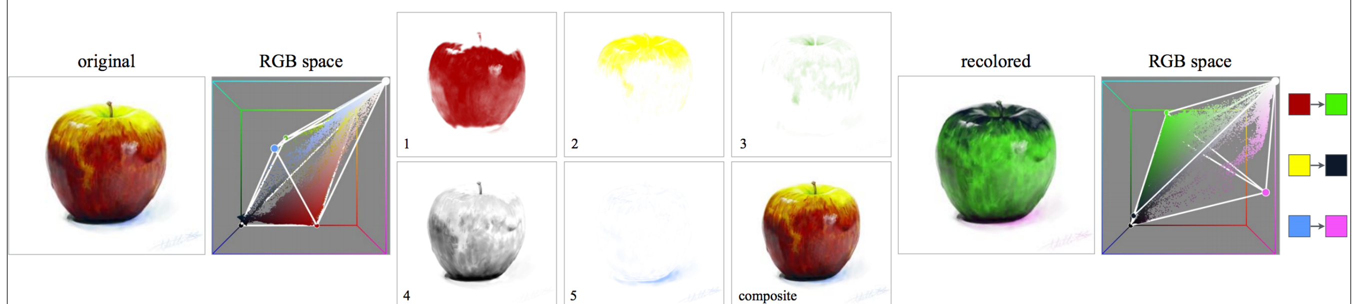
Given such painting image, we may want to enable editing operations such as recoloring, copy-paste-delete, edge enhancement and so on.



We reversely decompose the painting to several multispectral primary pigments and their corresponding per-pixel mixing weights, based on KM model. Then we can edit the painting in pigment space easily. The pigment here is not actually RGB colors, but several multispectral absorption and scattering curves.

Related Work

- Digital palette based editing.
    - Chang et al. 2015; Tan et al. 2016; Lin et al. 2017; Zhang et al. 2017, Aksoy et al. 2017.



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

There are several previous works focusing on palette based image decomposition and editing problems. Tan et al. 2016 is most related one with our current project. All these works are based on digital palette color compositing model, such as alpha blending, additive mixing and gaussian mixtures. Ours is based on a physically inspired color model.

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)

There are also several works focusing on KM model's applications in image editing, pigment identification and so on. Aharoni-Mack et al. 2017 is most similar to our work, it is contemporaneous with our work. They use a 3 wavelength watercolor pigment database from previous work to help extracting the watercolor pigments, assuming varying pigment thickness. while we evaluate our approach with (opaque) acrylic and oil paintings and compute an 8-wavelength constant-thickness decomposition.

Problem Statement

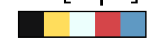
Input: Image pixels' RGB colors: **I**. 

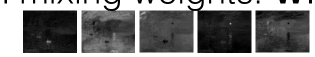
Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

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

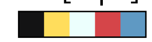
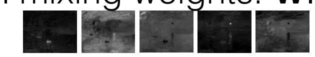




Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

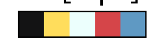
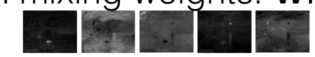
Output: Primary multispectral pigments: **H**=[**A**|**S**].  Their per-pixel mixing weights: **W**. 

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

Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

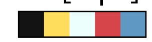
Output: Primary multispectral pigments: **H**=[**A**|**S**].  Their per-pixel mixing weights: **W**. 

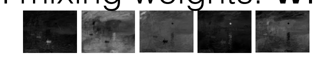
Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

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





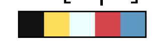
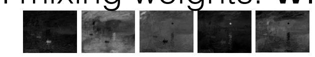
$$\mathbf{I} = \phi(km(\mathbf{WH}, t, \xi = 1))$$

Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

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

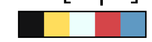
 

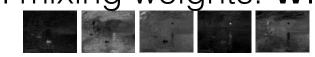
Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

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





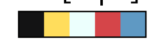
$$\mathbf{I} = \phi(km(\mathbf{WH}, t = 1, \xi = 1))$$

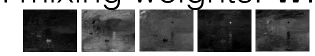
Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

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

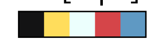
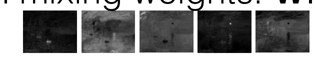




Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

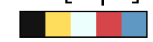
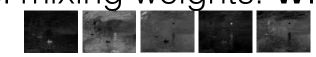
Output: Primary multispectral pigments: **H**=[**A**|**S**].  Their per-pixel mixing weights: **W**. 

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

Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

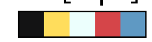
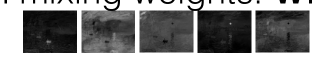
Input: Image pixels' RGB colors: **I**. 

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

Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

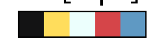
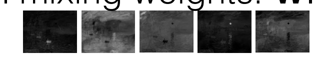
Output: Primary multispectral pigments: **H**=[**A**|**S**].  Their per-pixel mixing weights: **W**. 

$$||\mathbf{I} - \phi(km(\mathbf{WH}))||^2$$

Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Problem Statement

Input: Image pixels' RGB colors: **I**. 

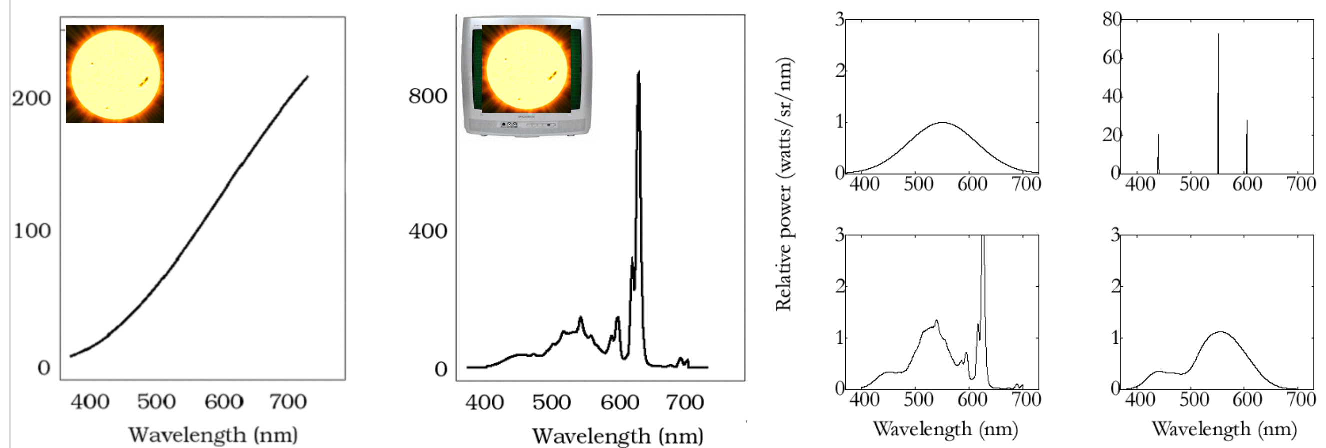
Output: Primary multispectral pigments: **H**=[**A**|**S**].  Their per-pixel mixing weights: **W**. 

$$||\mathbf{I} - \phi(km(\mathbf{WH}))||^2$$

It is under-constrained, and there are two additional challenges!

Our input is a digitized physical panting RGB image I.
Our output is some primary pigments H, which concatenates Absorption and Scattering together, and per-pixel mixing weights W. WH is total image's per pixel mixed pigment parameters. The relationship is modeled by such equation.
We have two assumptions to simplify the model complexity.
We assume the canvas reflectance eta is 1, which means canvas is pure white.
We assume the pigment thickness t over canvas is homogenous, with value equal to 1. The thickness in the model is a scale factor. so set to be 1 did not influence correctness of model, changing the constant thickness t to another value is equivalent to uniformly scaling all pigments' absorption and scattering.
After we fixed these two variables, then our model is only related with W and H.
Then we can solve this problem using a nonlinear least square optimization. but this is a under constrained optimization, since unknown variable number is much larger than known equation numbers. And there are two additional challenges hidden in this problem.

Challenge 1: Metamerism



CS184/284A, Lecture 15 Ren Ng, Spring 2016

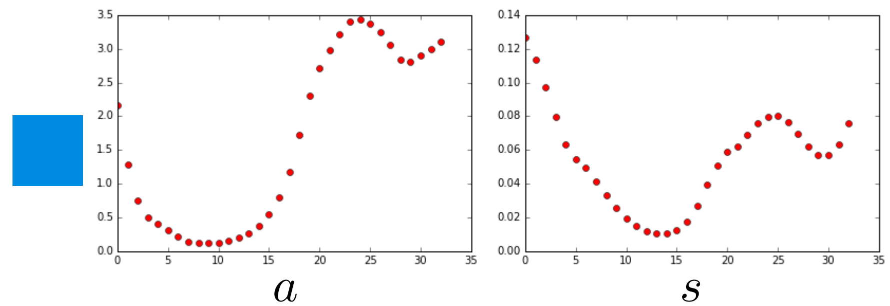
First challenge is Metamerism, which means that different high dimension spectrum can be rendered into same 3d color. It is useful for color reproduction on device screen. However, it is not good for reverse engineering works that want to reconstruct the spectrum from RGB color.

Solution: Smoothness Regularization

However, when observe these ground truth pigments’ absorption and scattering curves, we can find their spectra is smooth across wavelength. Also, we find the division of two parameters should also be smooth, this division is an important variable in KM model, that is related with pigments’ mass tone color.
We can enforce these three curve smoothness during the optimization to regularize the spectrum shape. This regularization is useful for solving Metamerism problem.

Solution: Smoothness Regularization

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

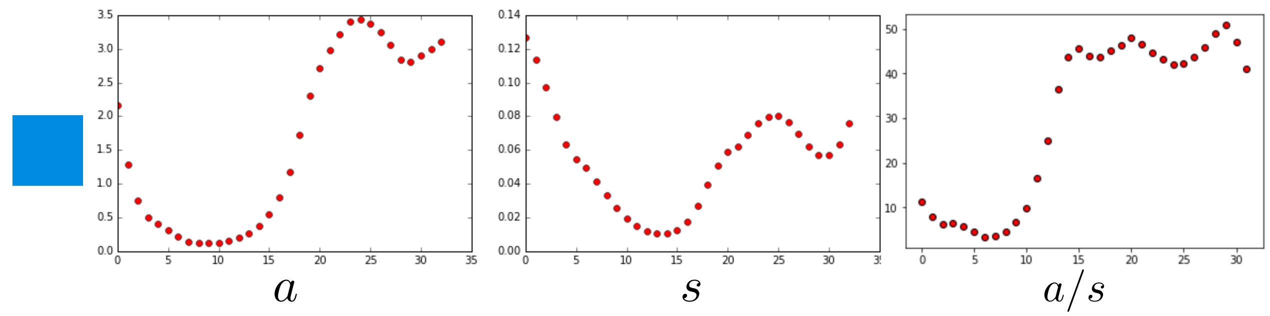


However, when observe these ground truth pigments' absorption and scattering curves, we can find their spectra is smooth across wavelength. Also, we find the division of two parameters should also be smooth, this division is an important variable in KM model, that is related with pigments' mass tone color.

We can enforce these three curve smoothness during the optimization to regularize the spectrum shape. This regularization is useful for solving Metamerism problem.

Solution: Smoothness Regularization

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



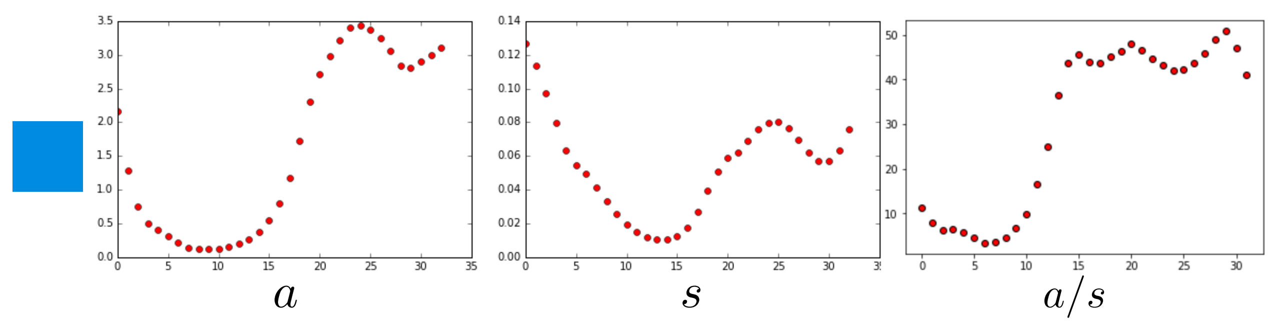
Absorption and Scattering's division curve should also be smooth.

However, when observe these ground truth pigments' absorption and scattering curves, we can find their spectra is smooth across wavelength. Also, we find the division of two parameters should also be smooth, this division is an important variable in KM model, that is related with pigments' mass tone color.

We can enforce these three curve smoothness during the optimization to regularize the spectrum shape. This regularization is useful for solving Metamerism problem.

Solution: Smoothness Regularization

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



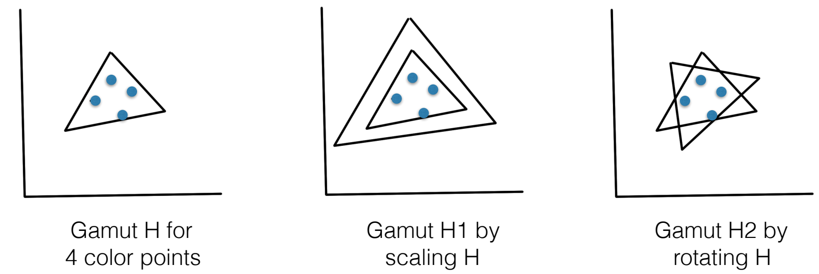
Absorption and Scattering's division curve should also be smooth.

Useful for Metamerism problem!

However, when observe these ground truth pigments' absorption and scattering curves, we can find their spectra is smooth across wavelength. Also, we find the division of two parameters should also be smooth, this division is an important variable in KM model, that is related with pigments' mass tone color.

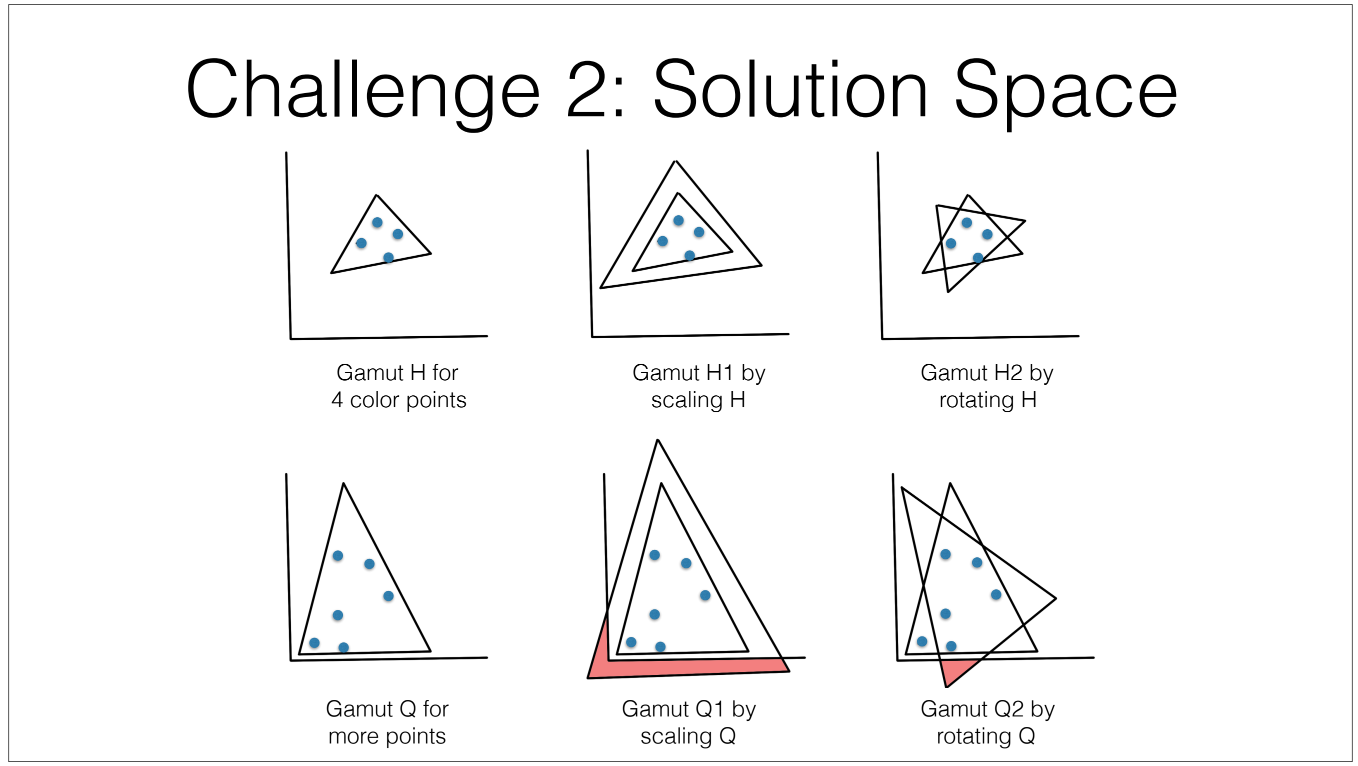
We can enforce these three curve smoothness during the optimization to regularize the spectrum shape. This regularization is useful for solving Metamerism problem.

Challenge 2: Solution Space



The other challenge is large solution space. For such 4 color points far away from color space boundary, suppose they are mixed by three primary pigments. The triangle gamut H is ground truth. However, any gamut scaled or rotated from H can be the new valid gamut for these four points. like H1 and H2.

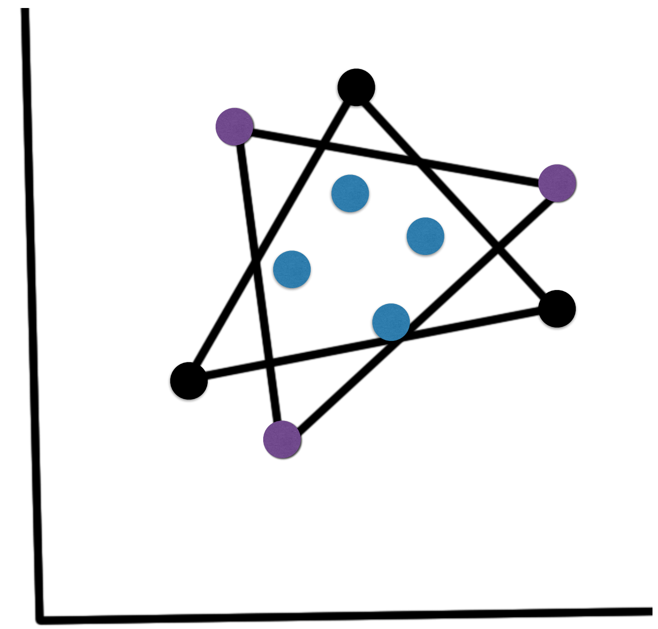
If color points are close to color space boundary, such gamut Q is already close to boundary. The scaling and rotating of ground truth gamut Q may be out of boundary, which means such gamut Q1 and Q2 are not acceptable, which is more restricted than top row situation. Such gamut problem is usually solved by Nonnegative Matrix Factorization optimization, the initial value is important for this optimization.



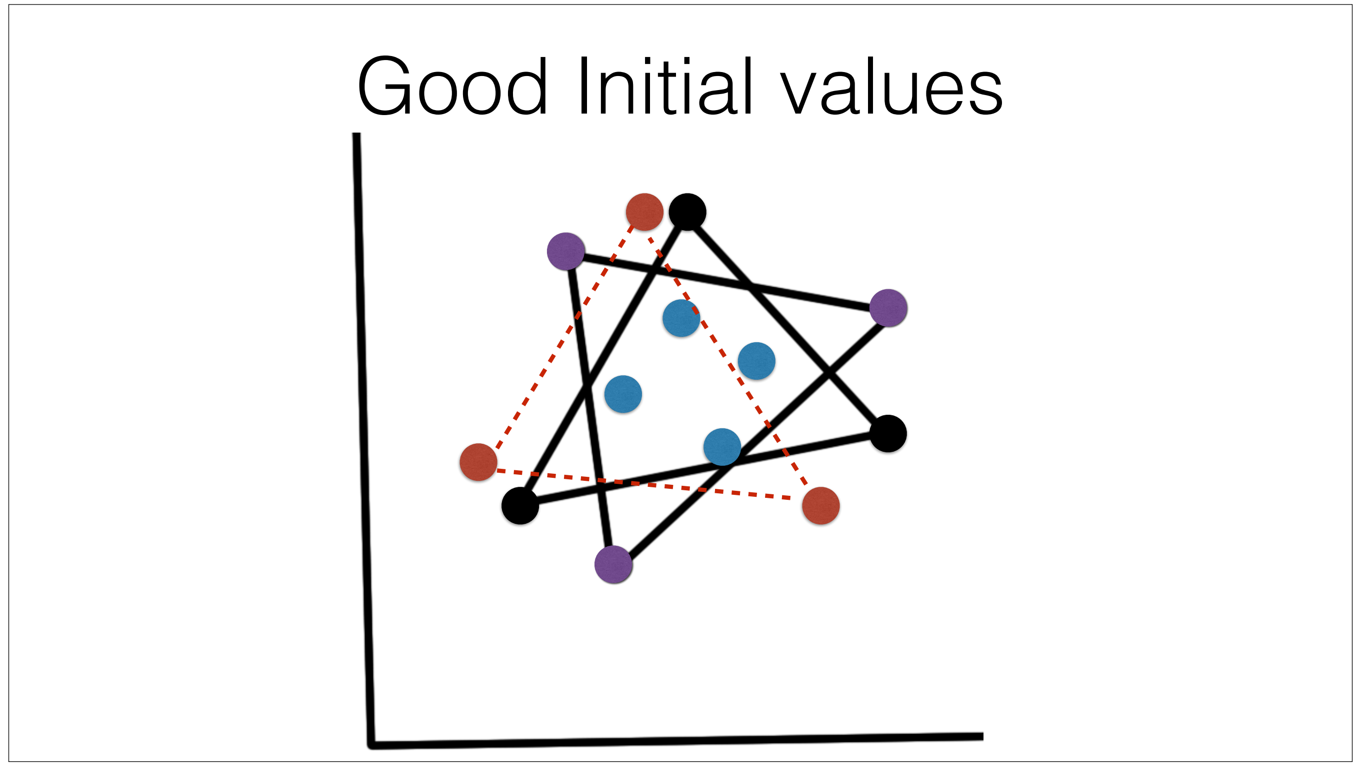
The other challenge is large solution space. For such 4 color points far away from color space boundary, suppose they are mixed by three primary pigments. The triangle gamut H is ground truth. However, any gamut scaled or rotated from H can be the new valid gamut for these four points. like H1 and H2.

If color points are close to color space boundary, such gamut Q is already close to boundary. The scaling and rotating of ground truth gamut Q may be out of boundary, which means such gamut Q1 and Q2 are not acceptable, which is more restricted than top row situation. Such gamut problem is usually solved by Nonnegative Matrix Factorization optimization, the initial value is important for this optimization.

Good Initial values

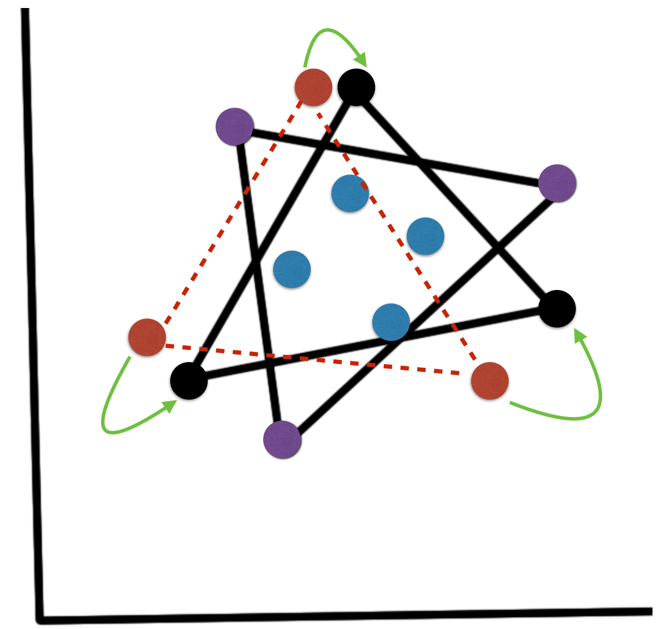


Here is a simple example to show good initial values are important. The black dots are ground truth pigments. The purple dots are other pigments that can reconstruct blue dots inside. The red dots means good initial values of optimization. The red gamut will gradually move to the gamut with black pigments, not the gamut with purple pigments.



Here is a simple example to show good initial values are important. The black dots are ground truth pigments. The purple dots are other pigments that can reconstruct blue dots inside. The red dots means good initial values of optimization. The red gamut will gradually move to the gamut with black pigments, not the gamut with purple pigments.

Good Initial values



Here is a simple example to show good initial values are important. The black dots are ground truth pigments. The purple dots are other pigments that can reconstruct blue dots inside. The red dots means good initial values of optimization. The red gamut will gradually move to the gamut with black pigments, not the gamut with purple pigments.

Divide into two subproblems

Directly solving this problem is hard.

Directly solving this optimization with those regularization terms is hard, in terms of performance and convergence of such a non-convex problem.
It is better to divide it into two subproblems.
First, extract primary pigments and second, fixed primary pigments to extract per-pixel mixing weights.

Divide into two subproblems

Directly solving this problem is hard.

We divide it into two subproblems:

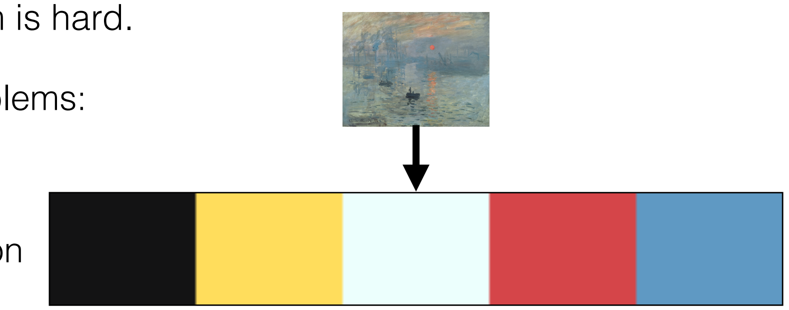
Directly solving this optimization with those regularization terms is hard, in terms of performance and convergence of such a non-convex problem.
It is better to divide it into two subproblems.
First, extract primary pigments and second, fixed primary pigments to extract per-pixel mixing weights.

Divide into two subproblems

Directly solving this problem is hard.

We divide it into two subproblems:

1. Primary pigments extraction

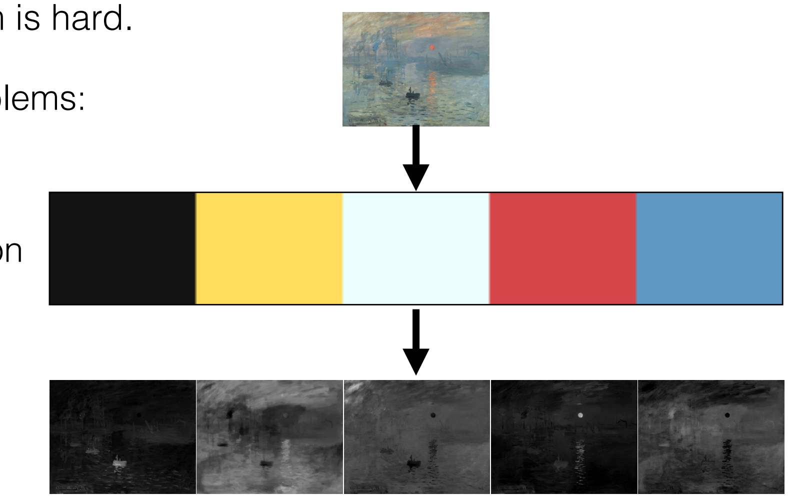


Directly solving this optimization with those regularization terms is hard, in terms of performance and convergence of such a non-convex problem.
It is better to divide it into two subproblems.
First, extract primary pigments and second, fixed primary pigments to extract per-pixel mixing weights.

Divide into two subproblems

Directly solving this problem is hard.

We divide it into two subproblems:



1. Primary pigments extraction

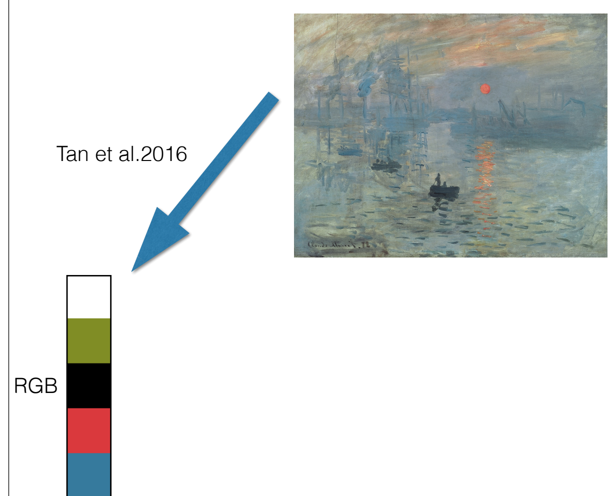
2. Mixing weights extraction

Directly solving this optimization with those regularization terms is hard, in terms of performance and convergence of such a non-convex problem.
It is better to divide it into two subproblems.
First, extract primary pigments and second, fixed primary pigments to extract per-pixel mixing weights.

Pigments Extraction

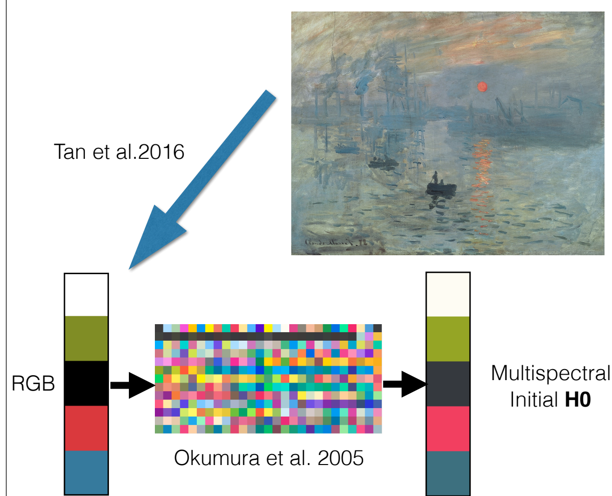
First we introduce pigment extraction.
We use Tan et al. 2016 simplified convex hull method to extract a small size RGB palette
We search a database to find closest RGB entry in database for each RGB palette color, and return corresponding pigment parameters as our initial pigment H0. We use a multispectral pigment database from Okumura et al. 2005, which contains only 26 acrylic pigments absorption and scattering parameters, we interpolate every pigment pairs by half-half mixture to expand the database.
We do not recover pigments from whole image pixels colors, we only use image's original convex hull vertices as representative colors of the image, since these vertices forms gamut shape.
Then we solve an Alternating Nonlinear Least Square optimization to extract pigment parameters, with help of pigment smoothness regularization term. this is actually the only stage of the algorithm that uses the A, S, A/S smoothness terms. Finally, we will get final multispectral pigments H star.

Pigments Extraction

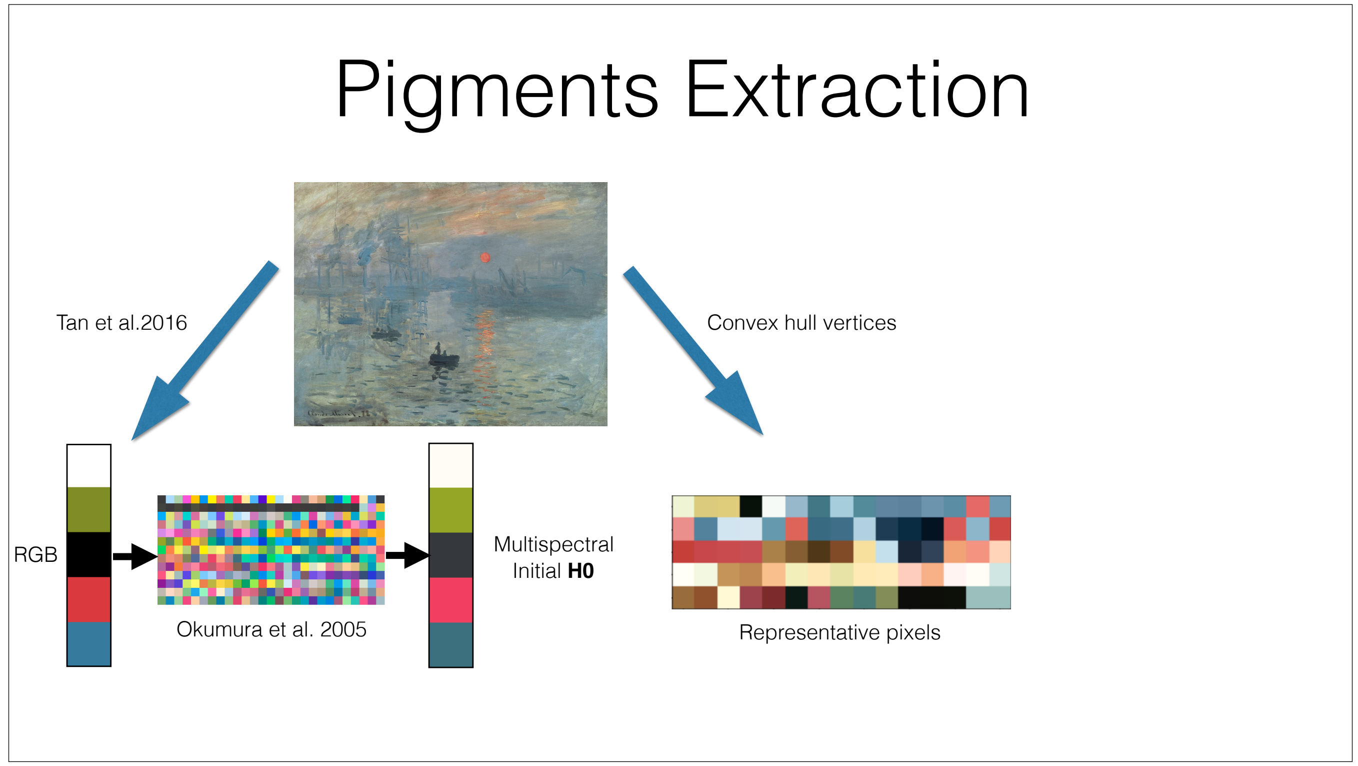


First we introduce pigment extraction.
We use Tan et al. 2016 simplified convex hull method to extract a small size RGB palette
We search a database to find closest RGB entry in database for each RGB palette color, and return corresponding pigment parameters as our initial pigment H0. We use a multispectral pigment database from Okumura et al. 2005, which contains only 26 acrylic pigments absorption and scattering parameters, we interpolate every pigment pairs by half-half mixture to expand the database.
We do not recover pigments from whole image pixels colors, we only use image's original convex hull vertices as representative colors of the image, since these vertices forms gamut shape.
Then we solve an Alternating Nonlinear Least Square optimization to extract pigment parameters, with help of pigment smoothness regularization term. this is actually the only stage of the algorithm that uses the A, S, A/S smoothness terms. Finally, we will get final multispectral pigments H star.

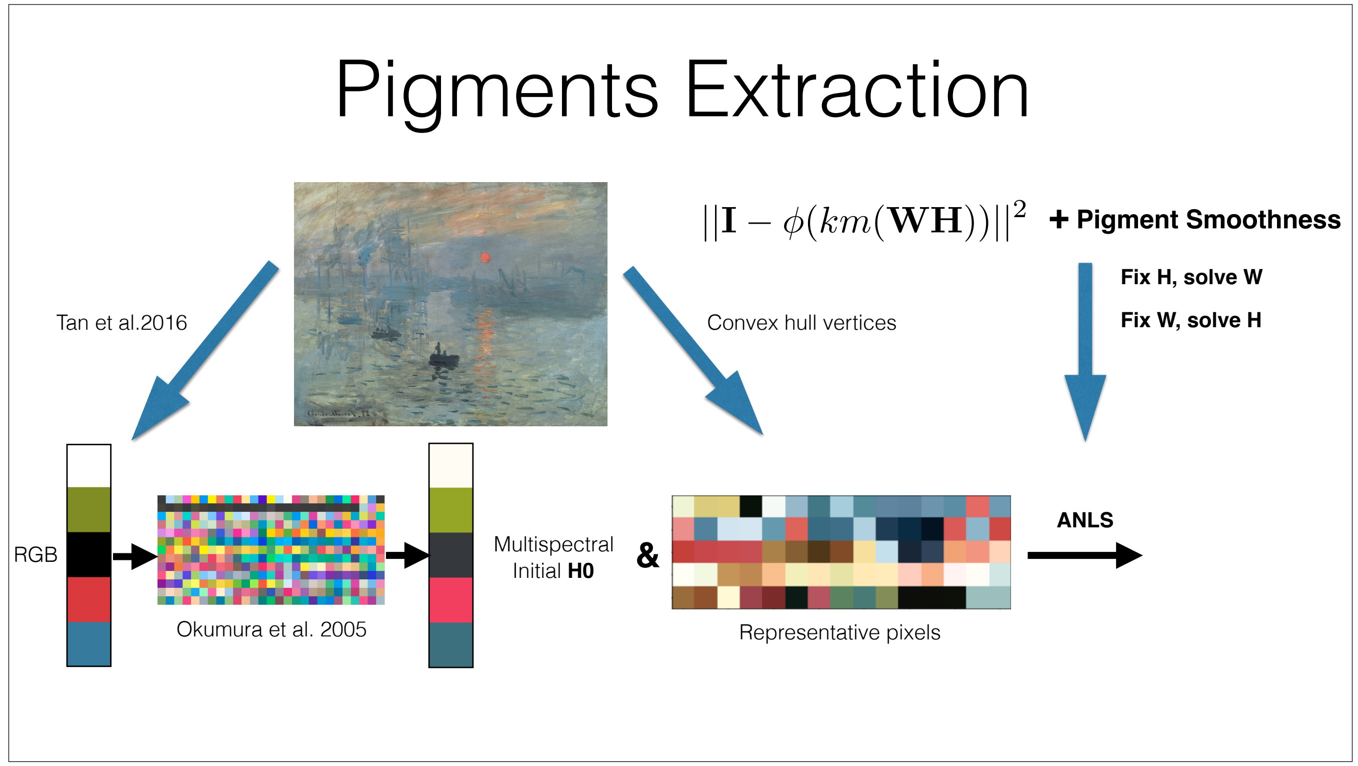
Pigments Extraction



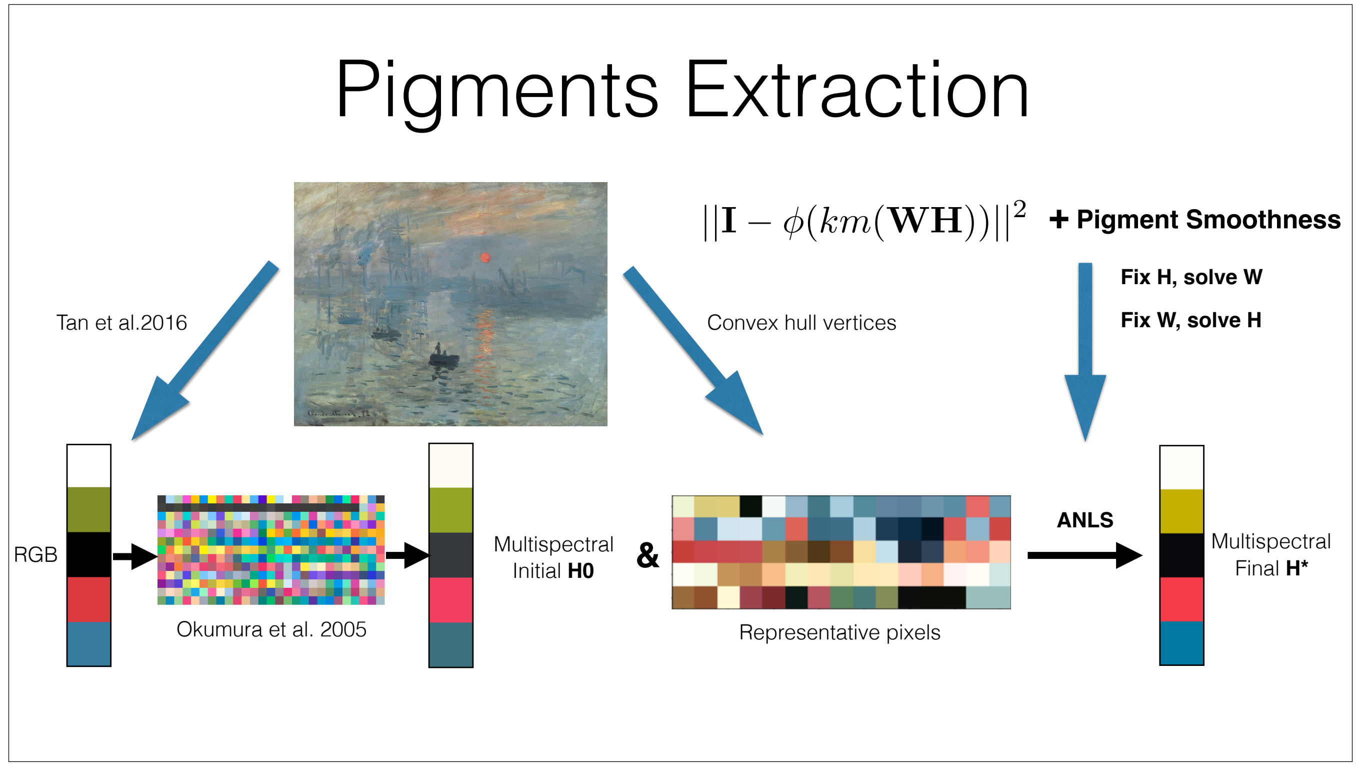
First we introduce pigment extraction.
We use Tan et al. 2016 simplified convex hull method to extract a small size RGB palette
We search a database to find closest RGB entry in database for each RGB palette color, and return corresponding pigment parameters as our initial pigment H0. We use a multispectral pigment database from Okumura et al. 2005, which contains only 26 acrylic pigments absorption and scattering parameters, we interpolate every pigment pairs by half-half mixture to expand the database.
We do not recover pigments from whole image pixels colors, we only use image's original convex hull vertices as representative colors of the image, since these vertices forms gamut shape.
Then we solve an Alternating Nonlinear Least Square optimization to extract pigment parameters, with help of pigment smoothness regularization term. this is actually the only stage of the algorithm that uses the A, S, A/S smoothness terms. Finally, we will get final multispectral pigments H star.



First we introduce pigment extraction.
We use Tan et al. 2016 simplified convex hull method to extract a small size RGB palette
We search a database to find closest RGB entry in database for each RGB palette color, and return corresponding pigment parameters as our initial pigment H0. We use a multispectral pigment database from Okumura et al. 2005, which contains only 26 acrylic pigments absorption and scattering parameters, we interpolate every pigment pairs by half-half mixture to expand the database.
We do not recover pigments from whole image pixels colors, we only use image's original convex hull vertices as representative colors of the image, since these vertices forms gamut shape.
Then we solve an Alternating Nonlinear Least Square optimization to extract pigment parameters, with help of pigment smoothness regularization term. this is actually the only stage of the algorithm that uses the A, S, A/S smoothness terms. Finally, we will get final multispectral pigments H star.



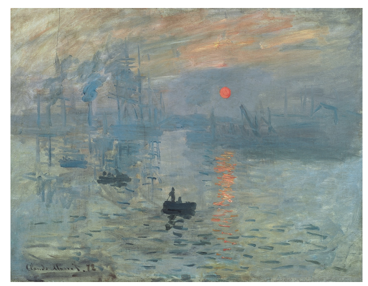
First we introduce pigment extraction.
We use Tan et al. 2016 simplified convex hull method to extract a small size RGB palette
We search a database to find closest RGB entry in database for each RGB palette color, and return corresponding pigment parameters as our initial pigment H0. We use a multispectral pigment database from Okumura et al. 2005, which contains only 26 acrylic pigments absorption and scattering parameters, we interpolate every pigment pairs by half-half mixture to expand the database.
We do not recover pigments from whole image pixels colors, we only use image's original convex hull vertices as representative colors of the image, since these vertices forms gamut shape.
Then we solve an Alternating Nonlinear Least Square optimization to extract pigment parameters, with help of pigment smoothness regularization term. this is actually the only stage of the algorithm that uses the A, S, A/S smoothness terms. Finally, we will get final multispectral pigments H star.



First we introduce pigment extraction.
We use Tan et al. 2016 simplified convex hull method to extract a small size RGB palette
We search a database to find closest RGB entry in database for each RGB palette color, and return corresponding pigment parameters as our initial pigment H0. We use a multispectral pigment database from Okumura et al. 2005, which contains only 26 acrylic pigments absorption and scattering parameters, we interpolate every pigment pairs by half-half mixture to expand the database.
We do not recover pigments from whole image pixels colors, we only use image's original convex hull vertices as representative colors of the image, since these vertices forms gamut shape.
Then we solve an Alternating Nonlinear Least Square optimization to extract pigment parameters, with help of pigment smoothness regularization term. this is actually the only stage of the algorithm that uses the A, S, A/S smoothness terms. Finally, we will get final multispectral pigments H star.

Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

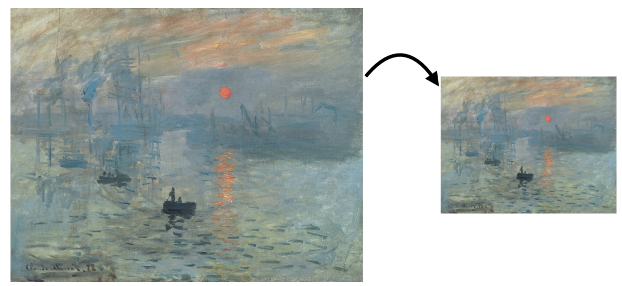


After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.



After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.
We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

Mixing Weights Extraction

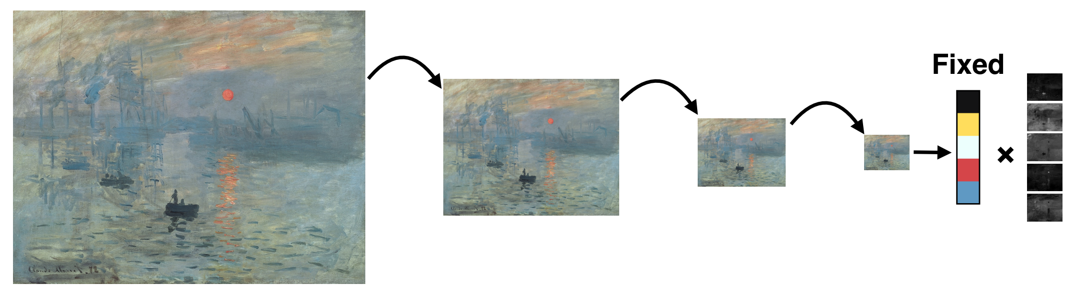
Given primary pigments, find per-pixel mixing weights.

After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.



After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

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

After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

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

After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

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

After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

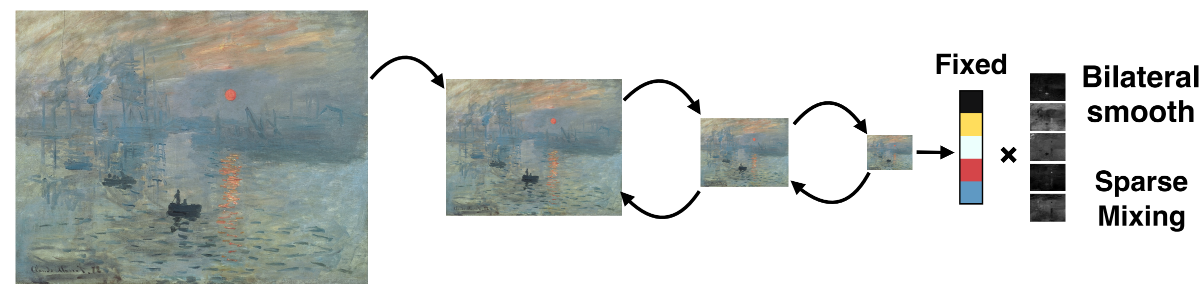
We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

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



After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

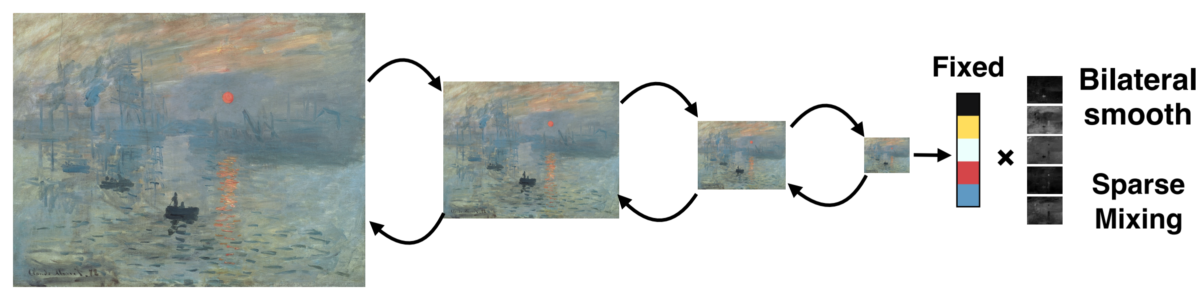
We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

Mixing Weights Extraction

Given primary pigments, find per-pixel mixing weights.

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

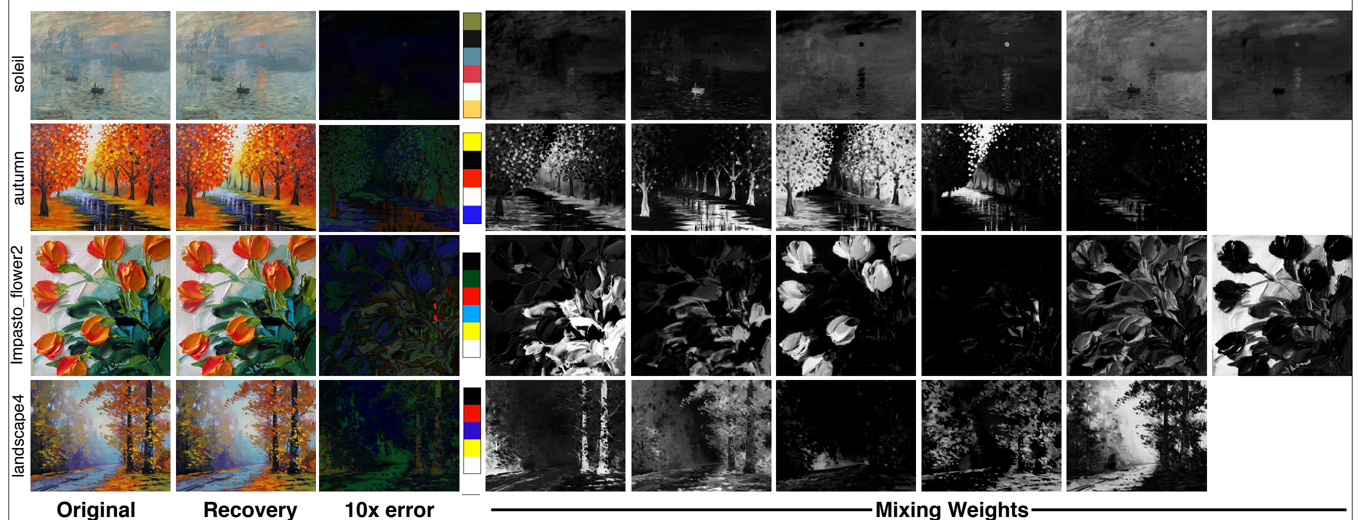
Sparsity: Each pixel's color is a mixing of smallest subset of primary pigments



After we extract the primary pigment parameters, we will fixed it and solve per pixel mixing weights for whole size image in a coarse-to-find manner. We down-sample the image into different resolution. At the smallest resolution level (in our case, the criteria is that one edge of image is below 80 pixels), we will solve the optimization with mixing weights spatial smoothness and mixing weights sparsity regularization.

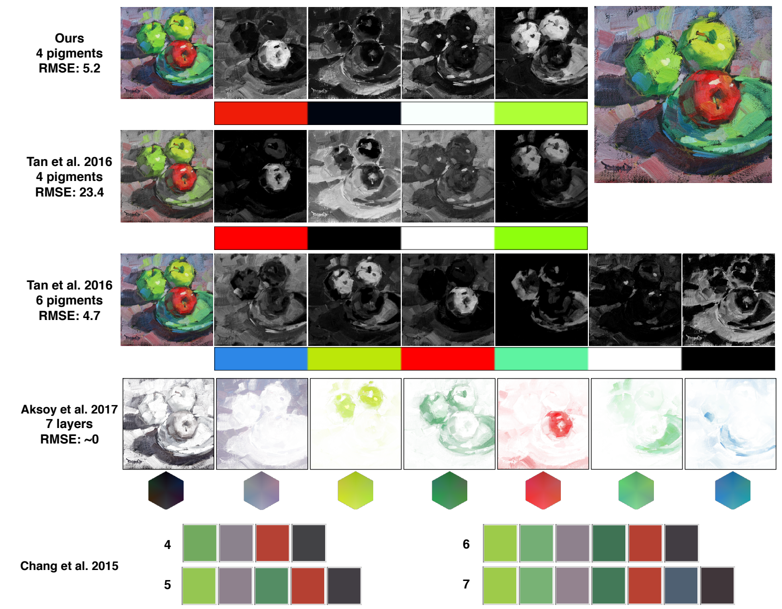
We choose bilateral grid smoothness because we want to keep edge structure of the paintings. Refer the details in our paper. After we got the mixing weights at lowest level, we will up-sampling it and feed as higher level's initial value to solve optimization. Finally we will get original resolution mixing weights.

Our results



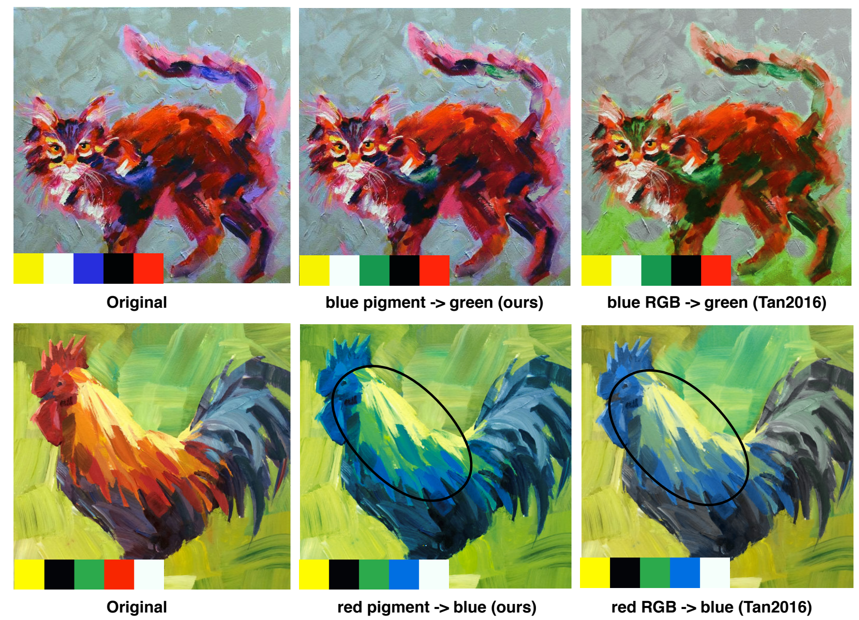
Here are our results for multiple images. From left to right: they are the original image, the reconstruction image, the error images that is already multiplied with scale 10, the extracted pigments, and their mixing weight maps. Because our pigments are multispectral, we show them as RGB colors rendered on a white canvas with unit thickness.

Compare to results from other models



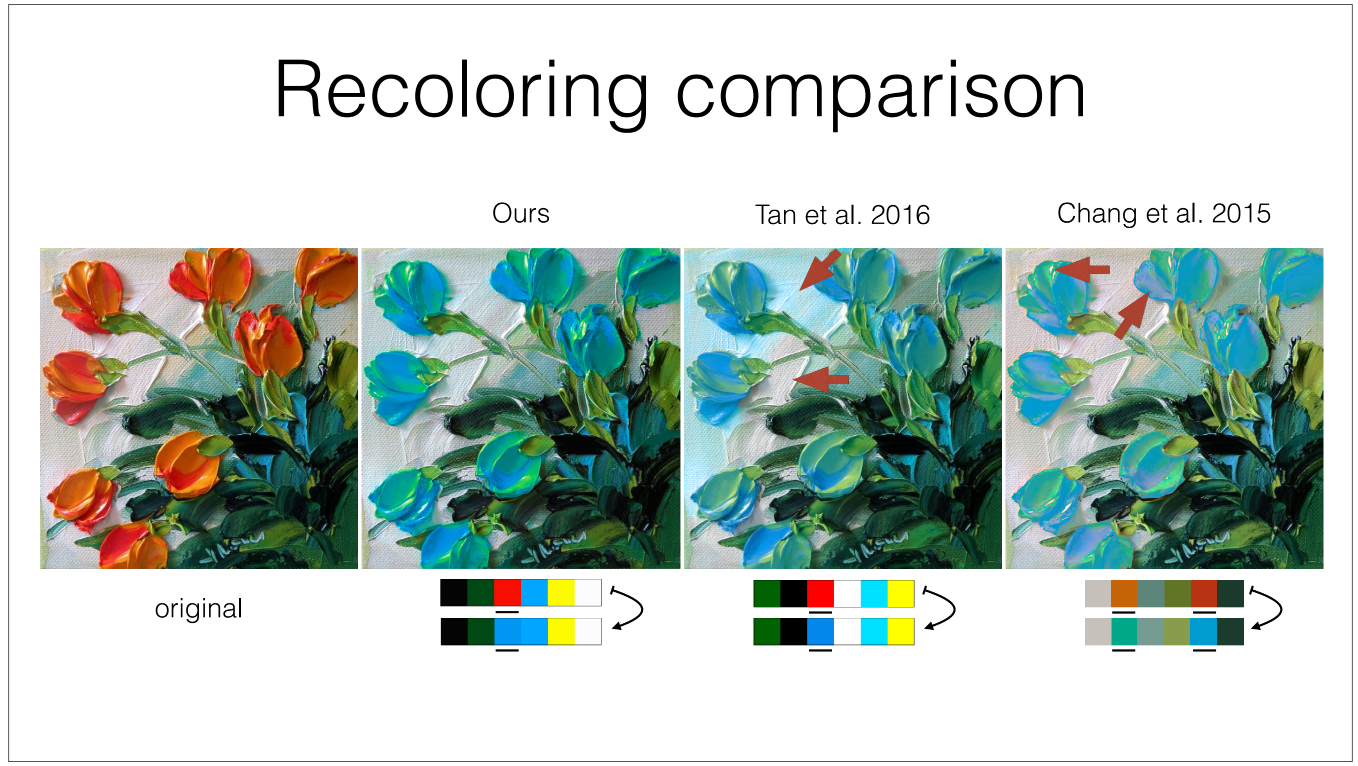
Here is comparison with the layer decompositions of Tan2016 and Aksoy2017, and with the palettes extracted by Chang2015. This apple painting was painted with exactly four physical pigments in real world. Our results match ground truth pigments and Reconstruction RMSE is small. When constrained to four colors only, Tan2016's approach has very high reconstruction error. To match our reconstruction error, Tan2016's approach needs to use more colors. Aksoy2017 approach extracts layers guaranteed to have zero reconstruction error, but the extracted layers are not composed of a single color, but color distributions. Chang2015 use modified k-means method to extract a palette whose size is also chosen by the user. For this example, Chang2015's palettes never contain the known ultramarine blue pigment, even for large size palettes.

Recoloring comparison



We compare KM model's recoloring results with digital color mixing based recoloring results.

We use our palette's RGB colors for layers in Tan2016 for direct comparison. In the cat painting, our KM mixing weight map for the blue pigment is sparse and therefore the recoloring effect is localized on the body of the cat. The weight map from Tan2016 is not sparse in the background area, resulting in undesired recoloring artifacts. For the rooster painting, using our KM model, more vibrant green is obtained from mixing yellow and new blue color in the circled region.



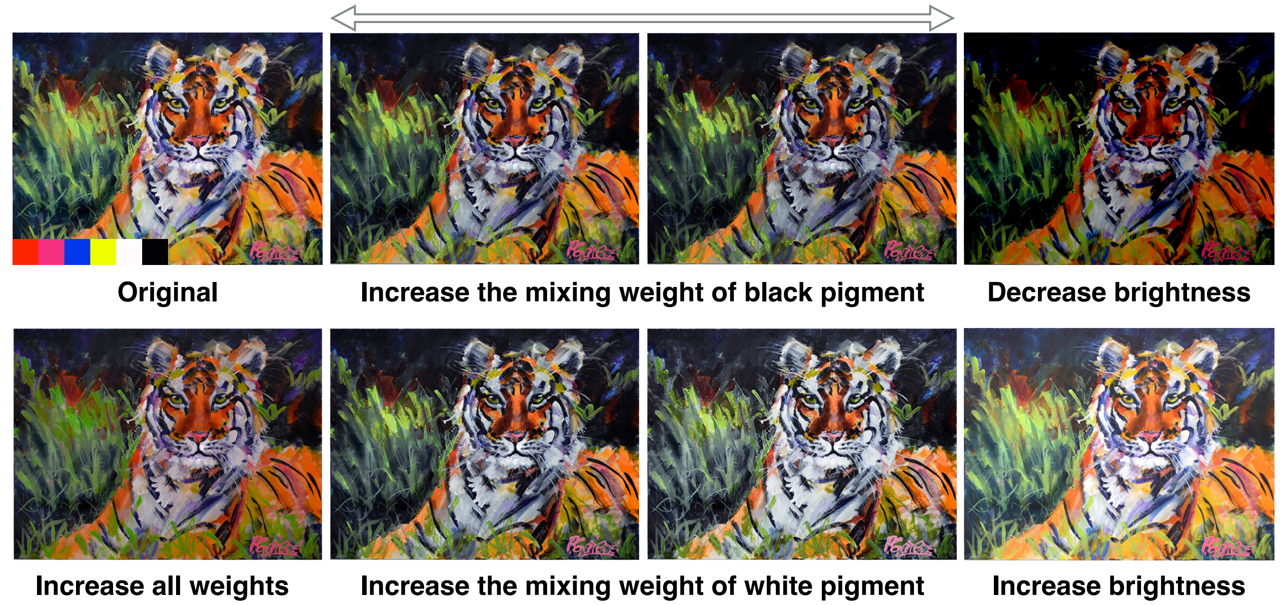
Here we compared our Kubelka-munk model based recoloring results with porter-duff model in tan2016 and Gaussian mixture model in chang2015. To be a fair comparison, each method extracts its own palette from the input image, so we attempt to mimic our result as closely as possible. Tan2016 suffers from lack of sparsity, while Chang2015 has local colors artifacts as pointed by red arrows in images.

Applications

Recoloring by modifying pigment weights

Different from only changing pigment colors to enable painting recoloring. We can also adjust the mixing weight value of a pigment to create recolored painting that would be difficult to reproduce using the features of a digital image software. Top is scaling mixing weights map of yellow pigment, bottom is scaling mixing weights map of red pigment.

Modify weights of black/white pigment



Adjusting the relative weights of black and white pigments is akin to adjusting the brightness and contrast of an image .
the result of increasing the black pigment weight is more like emphasizing shadows and detail, instead of just darkening globally, while the result of increasing the white pigment 's weight is desaturation of the colors.
Left bottom is increasing all pigments' mixing weights, which obtain paintings' mass tone color, which is unique from the traditional photoshop editing.
Right column is using photoshop to manipulating RGB image brightness globally.

Modify pigment scattering parameters

Original **Increase scattering** **Decrease scattering**

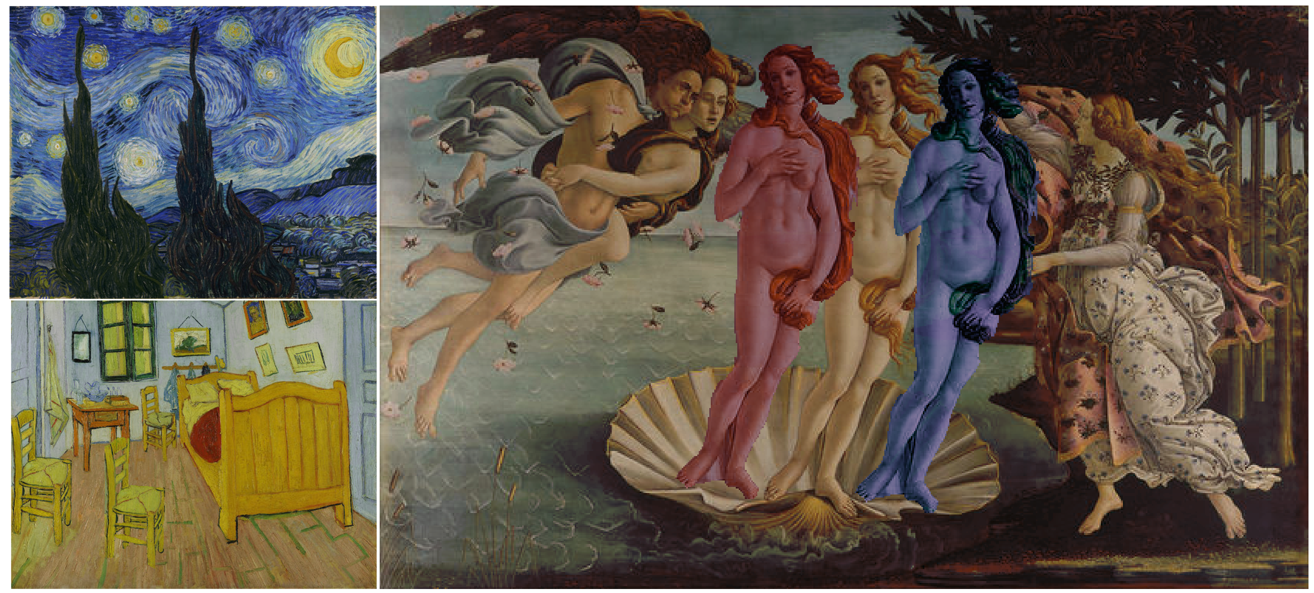
Our multispectral KM model results also can enable more paint like editing. For this example, green pigment's scattering is changed.
Increasing pigment scattering means that more light will be reflected back, so in some sense this is similar to brightening the green and making it more opaque, while decreasing scattering creates a darker green that absorbs more light than it scatters, so perhaps more like a stained glass.

Mask Selection

Rectangle Input   Grabcut on KM layer   Grabcut on RGB

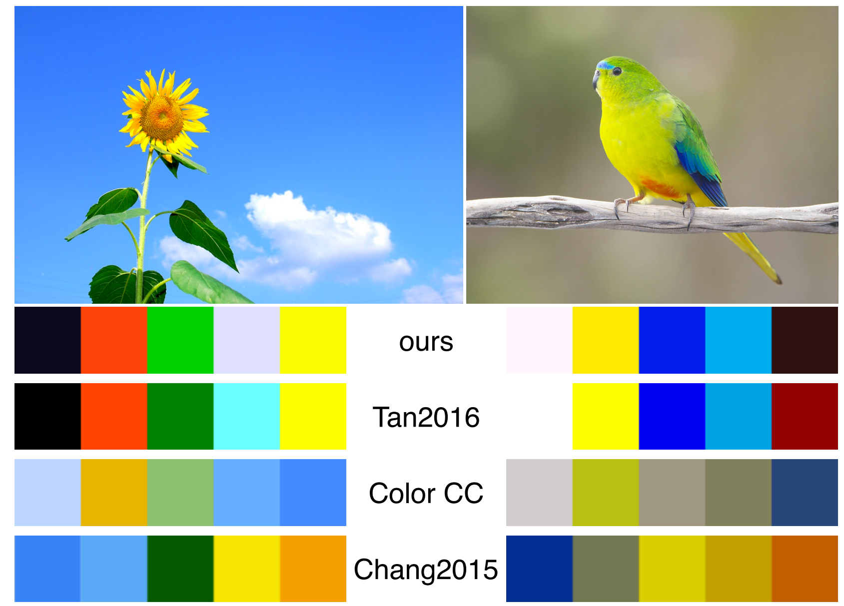
Mask Selection in paintings can be improved by optimizing on mixing weights map of a specific pigment, instead of optimizing on RGB colors.
On top example, GrabCut is performed on mixing weights of red pigment, on bottom example, Grabcut is performed on mixing weights map of black pigment. Both results are better than Grabcut results on RGB image. For fair comparison, no scribbles are provided for GrabCut algorithm.

Copy-Paste in pigment space

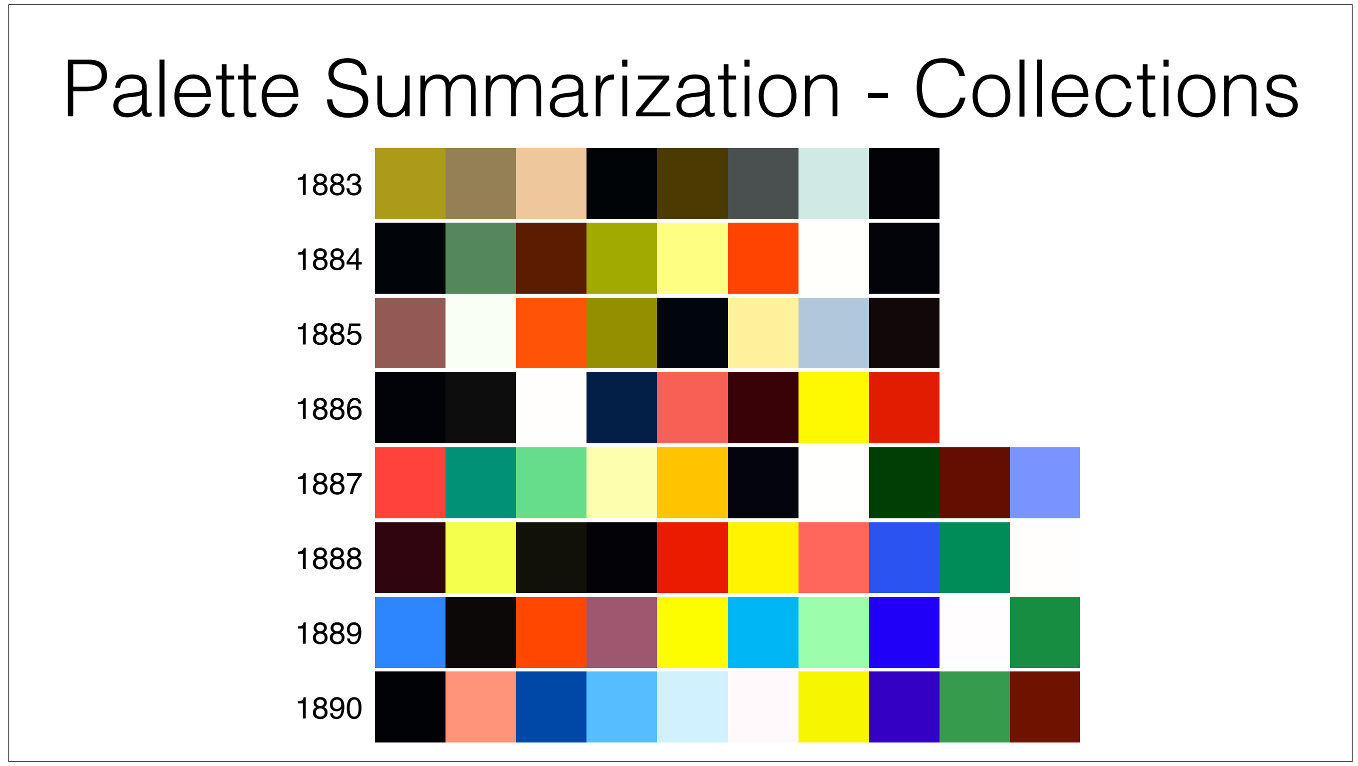


Here are some results of copy paste in pigment space. Each of these classical paintings has been modified by selecting some set of pigments from a region of pixels, and adding them as a new layer on top elsewhere in the image. While the pasted regions are not identical to the copied regions, similar to standard RGB editing, however, they appear as if they were painted as part of the image, if you do not know the original painting.

Palette Summarization - Photos



The first stage of our algorithm can also be seen as yet another method for extracting a small palette from an arbitrary image, not necessarily of paintings.
Here is Palette summarization applied to photos, as compared to Tan2016, Kuler, and Chang2015 . it is clear to see that Chang2015 and Kuler attempt to find “salient” or meaningful colors in some sense. Tan2016 and our work focus on colors that reconstruct the images. We achieve similar palette to Tan2016, but as we showed earlier our reconstructions have much lower error for the same number of colors.



Here are palette summarizations of Van Gogh's paintings arranged by year to show evolution of style. Two conclusions are clear from this analysis. First, the range of colors that Van Gogh painted with, expanded over the 1880's, as we expanded from eight pigments to ten pigments to achieve good reconstruction errors. Second, the pigment color vibrancy increased dramatically as well.

Edge detection and enhancement

on weights map | on RGB | original | Enhancement

Our weight maps can improve edge-based image analysis. We apply an existing edge detection method to each weight map separately and merge the per-pigment response using the per-pixel max operation. Edge images can be used to adapt standard image processing routines to be paint-aware. For example, we do edge enhancement by increase thickness of pigments near boundaries according to the edge response, which can visually emphasize painted objects in a different way than RGB edge enhancement.

Conclusion

read each

Conclusion

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

read each

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.

read each

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.

read each

Limitation and future work

read each

Limitation and future work

- Using prior acyclic pigment database as initial value may cause overfitting problem.

read each

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.

read each

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.

read each

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.

read each

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.

read each

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.

.

read each

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/
- **Our exposure in I-Programmer website:** https://www.i-programmer.info/news/144-graphics-and-games/10990-pigments-beyond-rgb.html
- **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.

Here is our contact information and project website.(If you have any suggestions or questions, please contact us.) Thanks for these artists providing great painting materials for us. Thank you!

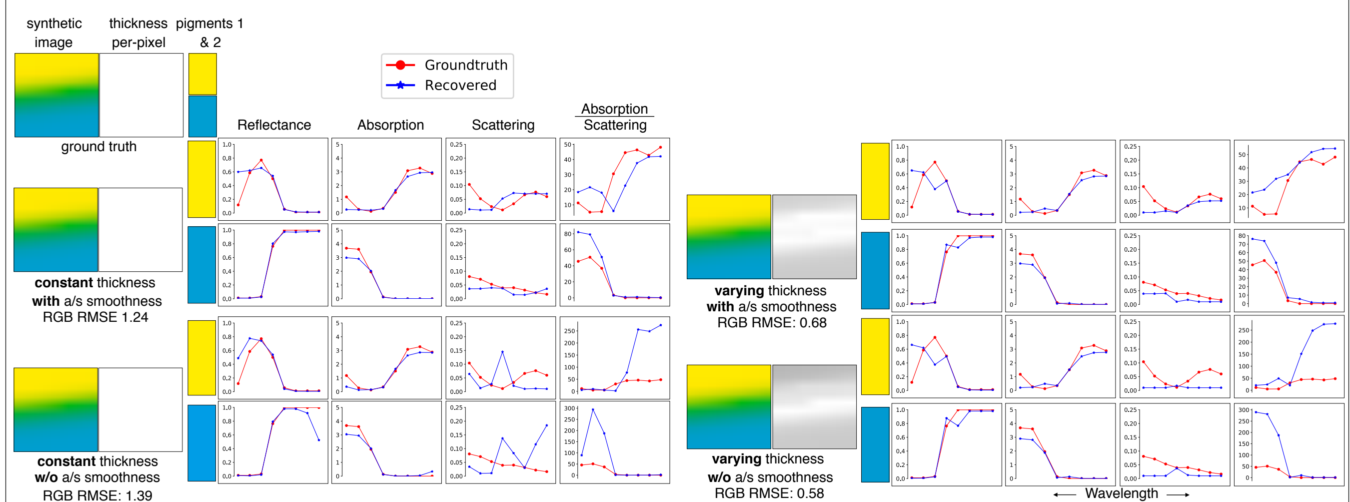
Extra Slides

Performance Information

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

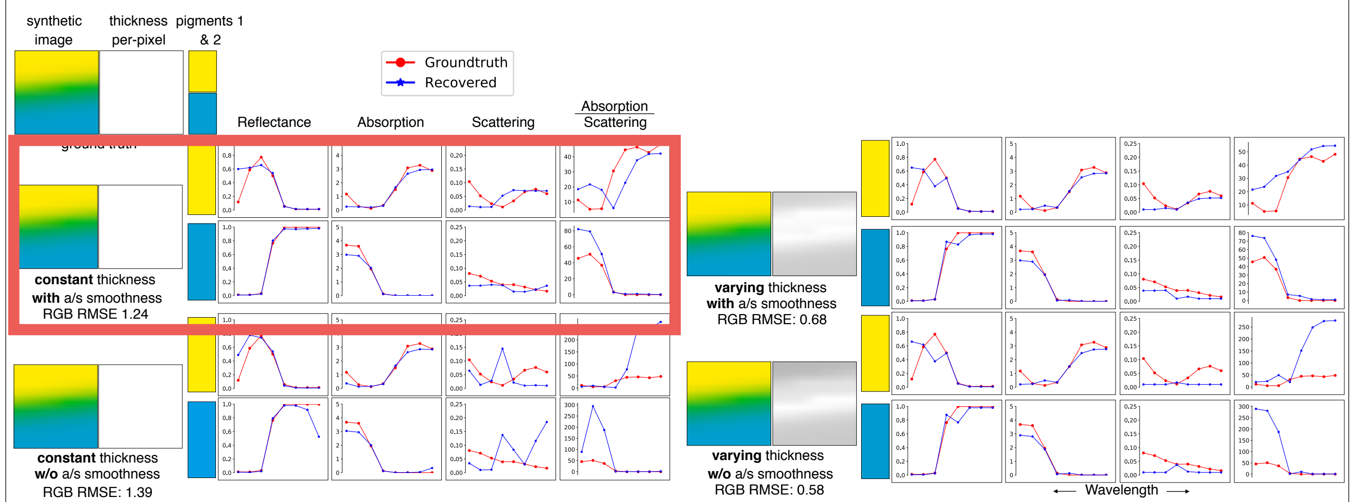
Runtime information is presented here. Our pipeline extracts M primary pigments in a few seconds and mixing weights maps in less than 10 minutes for a normal size image, with low RGB image reconstruction error. This table shows that we are generally faster than Tan2016, although our model is more complicated than Tan2016. Once the primary pigments and mixing weights are estimated, all of our editing applications occur in realtime.

Pigment smoothness and thickness



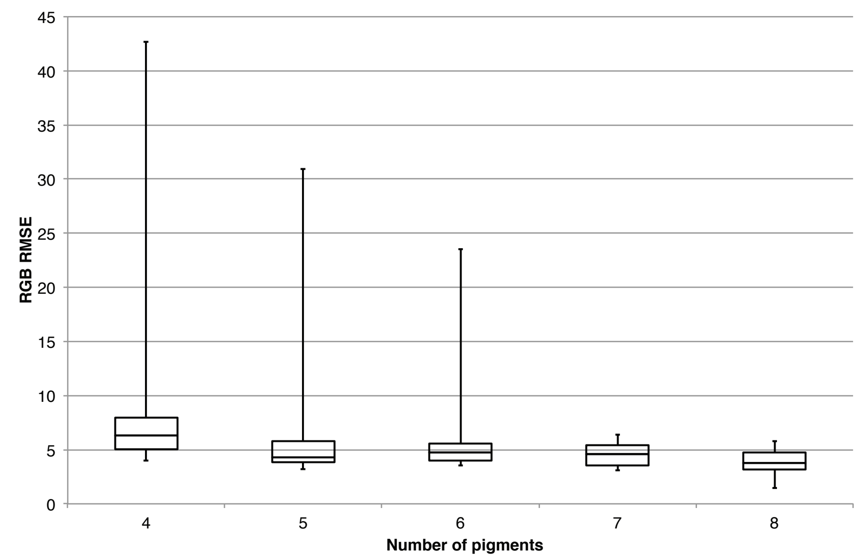
Allowing the thickness to vary introduces an additional degree-of-freedom per pixel. this figure shows an experiment in which we solve for two pigments' multispectral absorption and scattering parameters and per-pixel mixing weights; we optionally allow thickness to vary per-pixel. When thickness varies, the problem is under-constrained. To make the problem tractable, we add a smoothness regularization term. However, this leads to incorrect thickness estimation and less accurate multispectral reflectance (and slower optimization performance). While varying thickness may be particularly useful for watercolor or translucent paint, we did not pursue it in our thick-paint scenario beyond these initial experiments. The mass tone smoothness term results in scattering parameters that more closely match ground truth.

Pigment smoothness and thickness



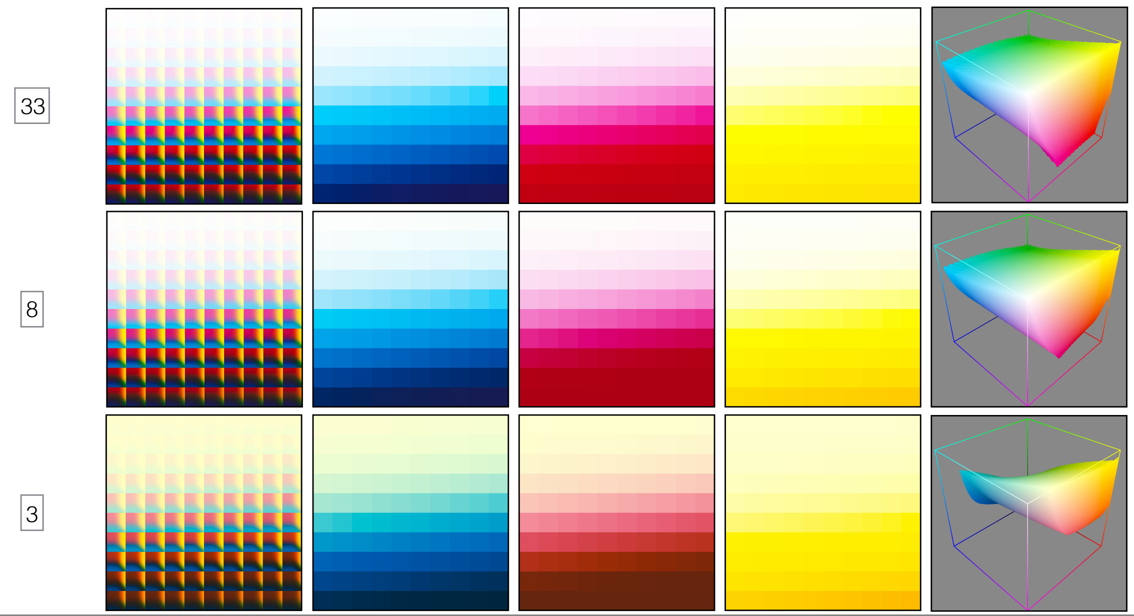
Allowing the thickness to vary introduces an additional degree-of-freedom per pixel. this figure shows an experiment in which we solve for two pigments' multispectral absorption and scattering parameters and per-pixel mixing weights; we optionally allow thickness to vary per-pixel. When thickness varies, the problem is under-constrained. To make the problem tractable, we add a smoothness regularization term. However, this leads to incorrect thickness estimation and less accurate multispectral reflectance (and slower optimization performance). While varying thickness may be particularly useful for watercolor or translucent paint, we did not pursue it in our thick-paint scenario beyond these initial experiments. The mass tone smoothness term results in scattering parameters that more closely match ground truth.

Pigment number influence



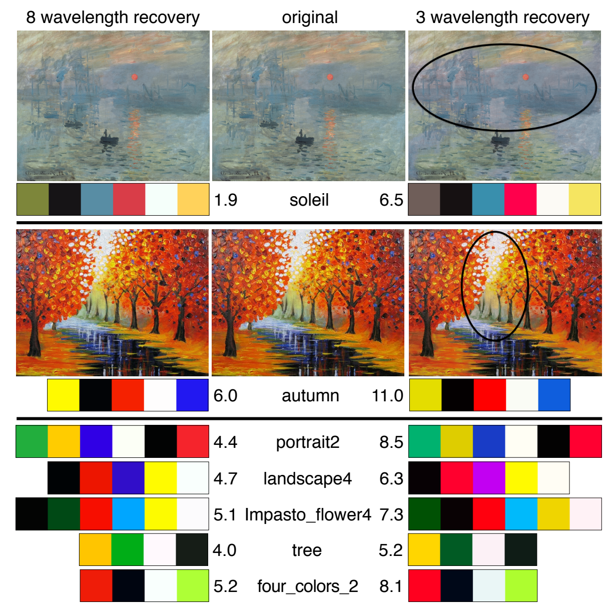
We plot the distribution of RGB RMSE of 12 example images' reconstructions on different palette size. Generally, RMSE will decrease when palette size increase, and RMSE distribution deviation will decrease when palette size increase.

Wavelength influence



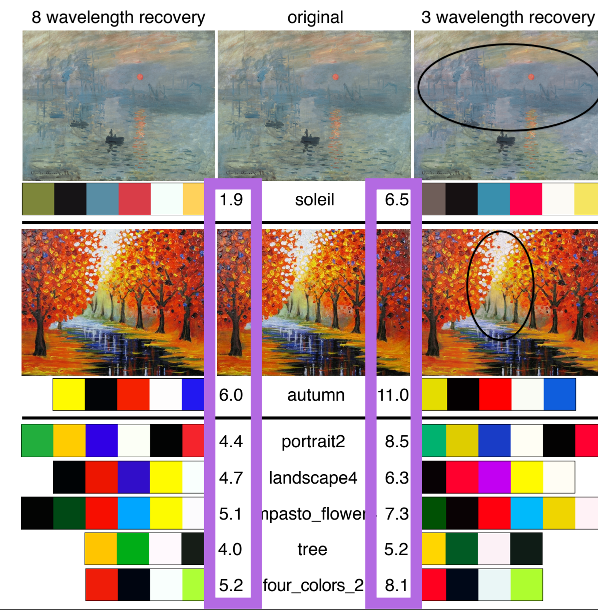
These color space point clouds show us the reason why we use multispectral pigments. when wavelength decreases, these three colors's gamut shrinks. We choose 8 wavelength in our all experiments, based on the trade off between variables numbers in optimization and image's reconstruct accuracy.

Wavelength influence

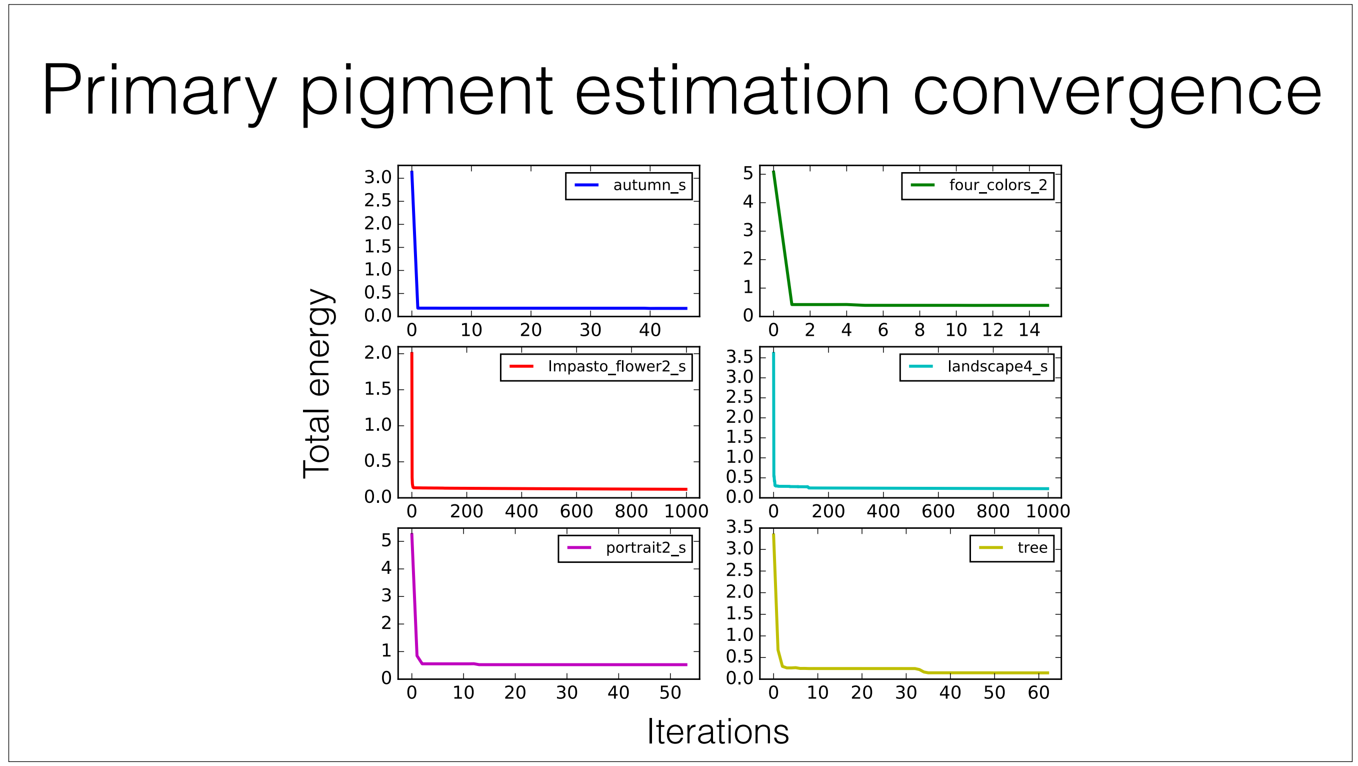


Comparison of 3 and 8 wavelength recovery, with RGB RMSE. We find 3 wavelength reconstruction error is higher for all examples. Soleil and autumn example show color distortion, due to the restricted gamut of the 3 wavelength pigment model.

Wavelength influence

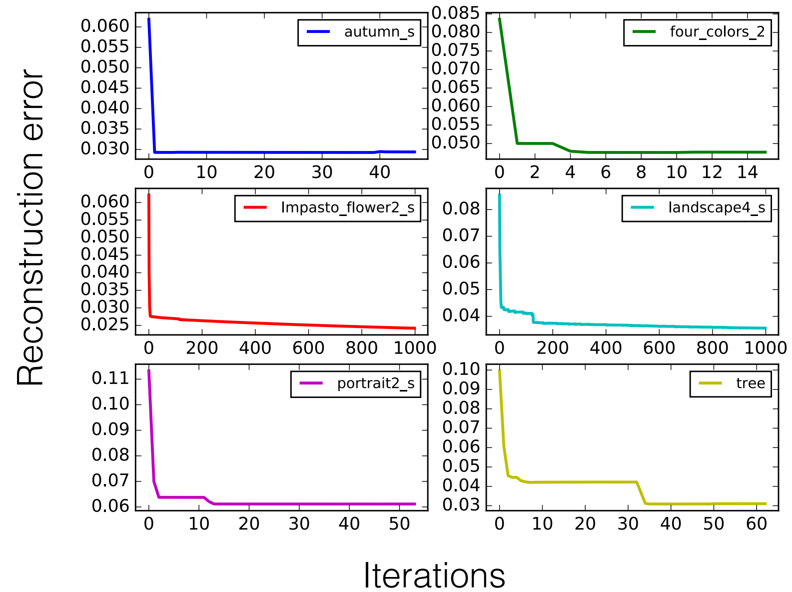


Comparison of 3 and 8 wavelength recovery, with RGB RMSE. We find 3 wavelength reconstruction error is higher for all examples. Soleil and autumn example show color distortion, due to the restricted gamut of the 3 wavelength pigment model.



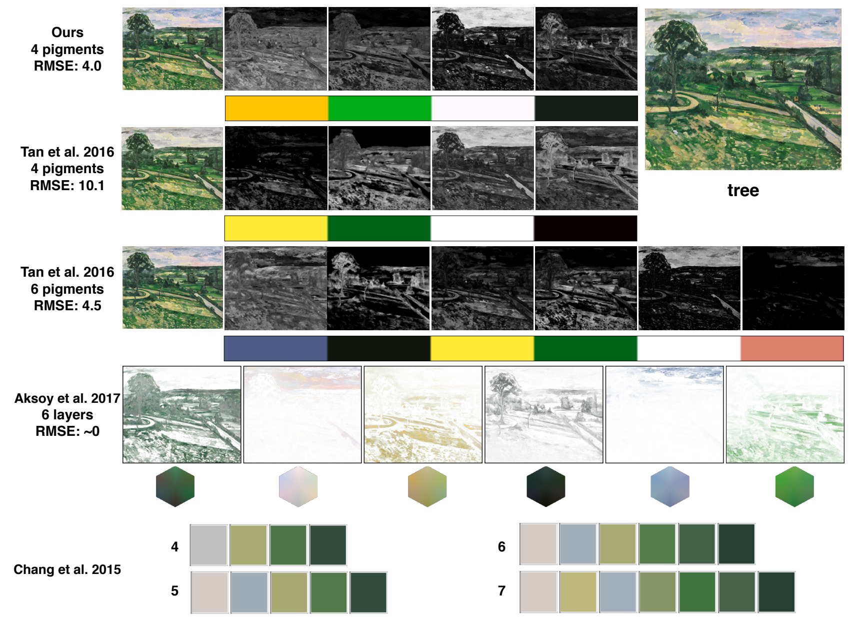
The total energy for our primary pigment estimation optimization decreases monotonically and rapidly after a few iterations for all examples. Some examples reach the maximum number of iterations rather than our strict convergence criteria.

Primary pigment estimation convergence

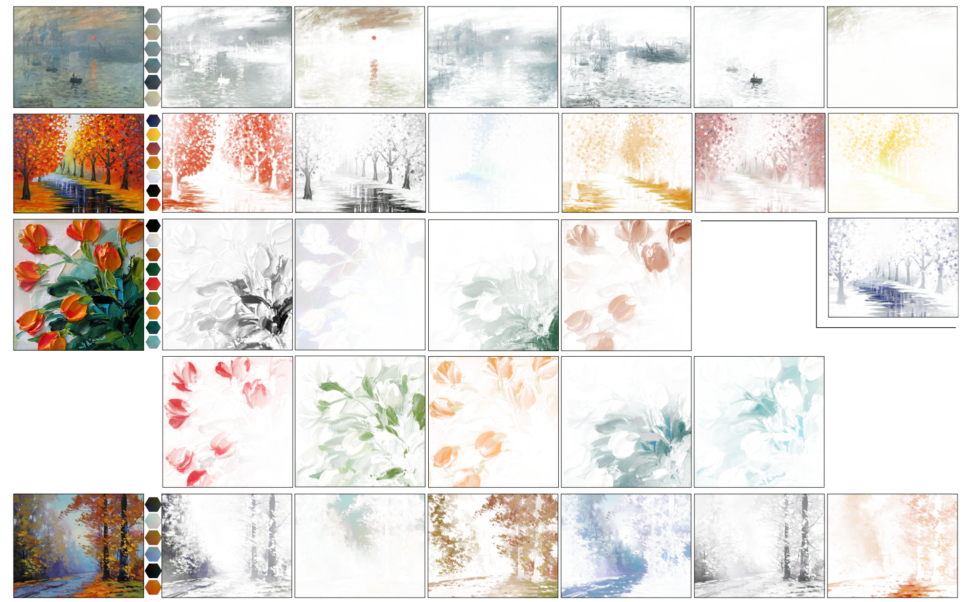


Reconstruction RMSE also decrease monotonically and rapidly.

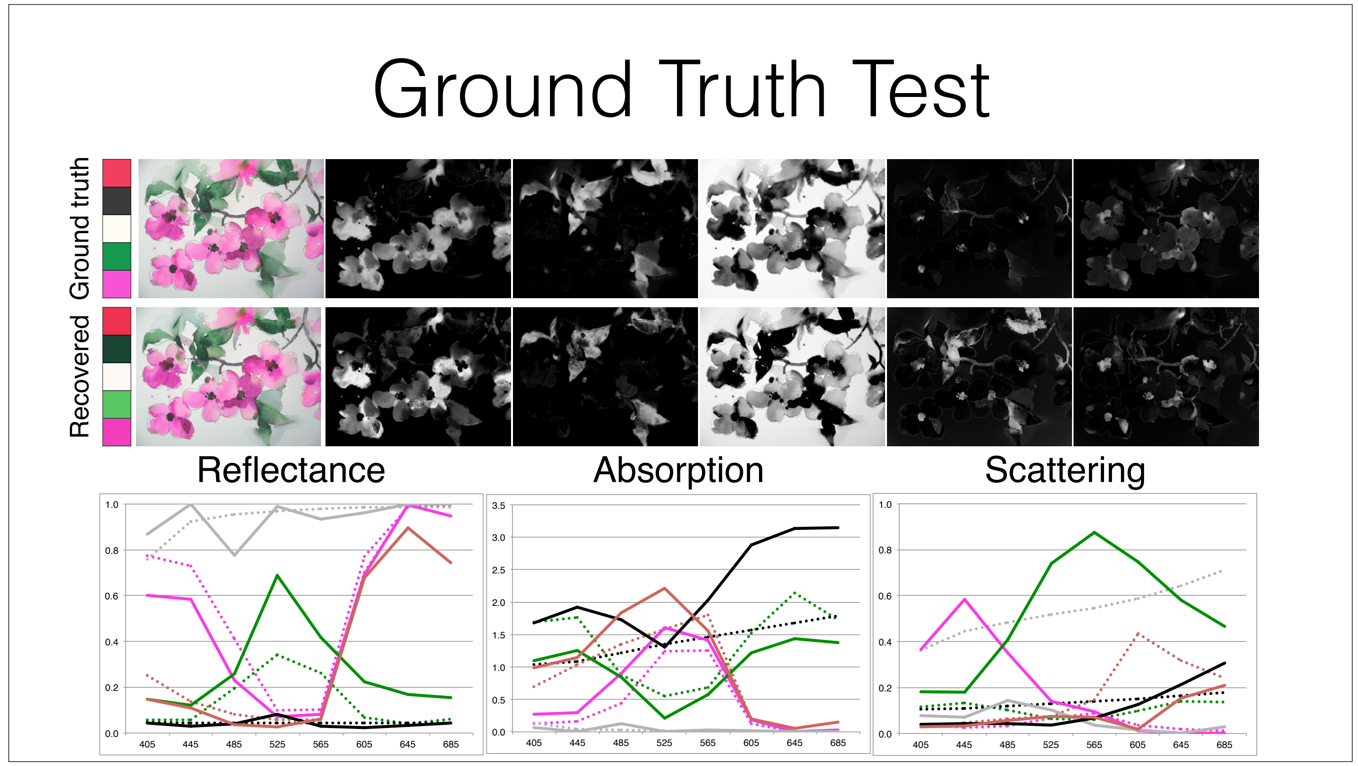
Compare to results from other models



Aksoy et al. 2017 results



The approach of Aksoy2017 applied to the same examples. The columns show the input image, their extracted palettes, and their layers. Reconstructions are not shown, because Aksoy's approach has no reconstruction error. This is because their palettes contain color distributions, not single colors. As a result, their layers are sometimes quite colorful and difficult to edit. The approach automatically chooses a palette size balancing choosing larger (sometimes redundant) palettes with less colorful layers.

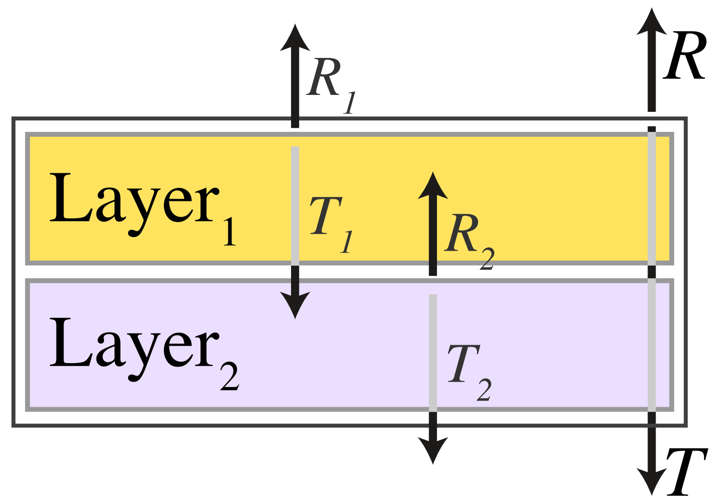


Recovering ground truth. Our reconstruction has low RGB error and the palette and mixing weight maps are similar upon inspection. The graphs of spectral curves show reflectances are recovered well, but absorption and scattering are less so. Ground truth curves are dashed, recovered are solid, and colors correspond to palette colors. This experiment confirms that there are many solutions to our reconstruction problem, but that we are able to reproduce plausible values.

Ground truth test information

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

Kubelka-Munk Layer Model



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

R is Reflectance
T is Transmittance

When people draw a painting, they usually use brush to put new pigments layers onto other existing pigments on the canvas. The procedure is similar to layer stacking in digital software. The KM model also include a nonlinear layer compositing model, which composite two pigment layer's reflectance and transmittance to get final Reflectance. Reflectance can be feed into standard imaging system to get final composited RGB colors.