LoCoPalettes: Local Control for Palette-based Image Editing

Cheng-Kang Ted Chao  Jason Klein  Jianchao Tan  Jose Echevarria  Yotam Gingold
Palette-based Image Recoloring
Palette-based Image Recoloring

- Palette selection and image editing
  - [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]
Palette-based Image Recoloring

- Palette selection and image editing
- [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]
Palette-based Image Recoloring

- Palette selection and image editing
- [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

Photo by Tobi
Palette-based Image Recoloring

- Palette selection and image editing
- [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

Color space visualization: https://yig.github.io/image-rgb-in-3D/

Photo by Tobi
Palette-based Image Recoloring

- Palette selection and image editing

- [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

\[ I = W \cdot P \]

Photo by Tobi

Color space visualization: https://yig.github.io/image-rgb-in-3D/
Palette-based Image Recoloring

- Palette selection and image editing
- [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

\[ I = W \cdot P \]

Photo by Tobi

Color space visualization: https://yig.github.io/image-rgb-in-3D/
Palette-based Image Recoloring

• Palette selection and image editing

• [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

\[ I = W \cdot P \]

I = \( W \cdot P \)

Photo by Tobi

Color space visualization: https://yig.github.io/image-rgb-in-3D/
Palette-based Image Recoloring

- Palette selection and image editing

- [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

\[ I = W \cdot P \]

Photo by Tobi

Color space visualization: https://yig.github.io/image-rgb-in-3D/
Palette-based Image Recoloring

• Palette selection and image editing

• [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

\[ I = W \cdot P \]

Photo by Tobi

Color space visualization: https://yig.github.io/image-rgb-in-3D/
Palette-based Image Recoloring

- Palette selection and image editing

- [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

\[ I = W \cdot P \]
\[ I' = W \cdot P' \]

Color space visualization: https://yig.github.io/image-rgb-in-3D/
Palette-based Image Recoloring

- Palette selection and image editing
  - [Chang et al. 2015], [Tan et al. 2016], [Tan et al. 2018], [Chao et al. 2021]

\[I = W \cdot P\]

\[I' = W \cdot P'\]

Color space visualization: https://yig.github.io/image-rgb-in-3D/
What’s the problem?
What’s the problem?

• How to change a color of a specific pixel into another color?
What’s the problem?

- How to change a color of a specific pixel into another color?
- It’s tedious when the color mixture of the pixel is nonobvious
What’s the problem?

• How to change a color of a specific pixel into another color?

• It’s tedious when the color mixture of the pixel is nonobvious
What’s the problem?

- How to change a color of a specific pixel into another color?
- It’s tedious when the color mixture of the pixel is nonobvious
What’s the problem?

• How to change a color of a specific pixel into another color?

• It’s **tedious** when the color mixture of the pixel is nonobvious
What’s the problem?

• How to change a color of a specific pixel into another color?
• It’s **tedious** when the color mixture of the pixel is nonobvious
What’s the problem?

- How to change a color of a specific pixel into another color?
- It’s tedious when the color mixture of the pixel is nonobvious.
What’s the problem?

• How to change a color of a specific pixel into another color?

• It’s **tedious** when the color mixture of the pixel is nonobvious
What’s the problem?

• How to change a color of a specific pixel into another color?

• It’s **tedious** when the color mixture of the pixel is nonobvious.
What’s the problem?

• How to change a color of a specific pixel into another color?

• It’s tedious when the color mixture of the pixel is nonobvious.
What’s the problem?

- How to change a color of a specific pixel into another color?
- It’s tedious when the color mixture of the pixel is nonobvious.
ColorfulCurves [Chao et al. 2023]
The indirect editing problem can be solved by finding the **sparsest** palette change via an $L_{2,1}$ sparse optimization.
**ColorfulCurves [Chao et al. 2023]**

- The indirect editing problem can be solved by finding the **sparsest** palette change via an $L_{2,1}$ sparse optimization.
**ColorfulCurves [Chao et al. 2023]**

- The indirect editing problem can be solved by finding the *sparsest* palette change via an $L_{2,1}$ sparse optimization.
ColorfulCurves [Chao et al. 2023]

- The indirect editing problem can be solved by finding the \textit{sparsest} palette change via an $L_{2,1}$ sparse optimization.
ColorfulCurves [Chao et al. 2023]

- The indirect editing problem can be solved by finding the **sparsest** palette change via an $L_{2,1}$ sparse optimization

$$\min_{\Delta \mathcal{P}} \| \Delta \mathcal{P} \|_{2,1}$$
The indirect editing problem can be solved by finding the sparsest palette change via an $L_{2,1}$ sparse optimization:

$$\min_{\Delta \mathcal{P}} \left\| \Delta \mathcal{P} \right\|_{2,1} \text{ subject to}$$
**ColorfulCurves** [Chao et al. 2023]

- The indirect editing problem can be solved by finding the **sparsest** palette change via an $L_{2,1}$ sparse optimization.

\[ \min_{\Delta \mathcal{P}} \| \Delta \mathcal{P} \|_{2,1} \]

subject to

\[ W \cdot (\mathcal{P} + \Delta \mathcal{P}) = \mathcal{P} \]
The indirect editing problem can be solved by finding the **sparsest** palette change via an $L_{2,1}$ sparse optimization:

$$\min_{\Delta \mathbb{P}} \| \Delta \mathbb{P} \|_{2,1}$$

subject to

$$W \cdot (\mathbb{P} + \Delta \mathbb{P}) =$$
ColorfulCurves [Chao et al. 2023]

- The indirect editing problem can be solved by finding the sparsest palette change via an $L_{2,1}$ sparse optimization

$$\begin{align*}
\min_{\Delta \mathbf{P}} & \| \Delta \mathbf{P} \|_{2,1} \\
\text{subject to} & \quad W \cdot (\mathbf{P} + \Delta \mathbf{P}) = I \\
I' &= W \cdot (\mathbf{P} + \Delta \mathbf{P})
\end{align*}$$
The indirect editing problem can be solved by finding the \textit{sparsest} palette change via an $L_{2,1}$ sparse optimization:

$$\min_{\Delta} \| \Delta \Pi \|_{2,1} \quad \text{subject to}$$

$$W \cdot (\Pi + \Delta \Pi) = I' = W \cdot (\Pi + \Delta \Pi)$$

\textbf{ColorfulCurves [Chao et al. 2023]}
There are *still* problems
There are still problems

- The applied edits are not sparse enough
There are still problems

- The applied edits are not sparse enough
- Why is sparsity important?
There are *still* problems

- The applied edits are not sparse enough
- Why is sparsity important?
There are still problems

• The applied edits are not sparse enough

• Why is sparsity important?
There are *still* problems

- The applied edits are not sparse enough
- Why is sparsity important?
There are still problems

- The applied edits are not sparse enough
- Why is sparsity important?
There are *still* problems

- The applied edits are not sparse enough
- Why is sparsity important?

[Tan et al. 2018]
There are still problems

- The applied edits are not sparse enough
- Why is sparsity important?
- It’s impossible to recolor semantically

[Tan et al. 2018]
There are *still* problems

- The applied edits are not sparse enough
  - Why is sparsity important?
- It’s impossible to recolor *semantically*
  - Two different objects share the same color

[Tan et al. 2018]
There are *still* problems

- The applied edits are not sparse enough
- Why is sparsity important?
- It’s impossible to recolor *semantically*
- Two different objects share the same color

[Tan et al. 2018]
There are *still* problems

- The applied edits are not sparse enough
- Why is sparsity important?
- It’s impossible to recolor *semantically*
- Two different objects share the same color

[Tan et al. 2018]
There are still problems

- The applied edits are not sparse enough
  - Why is sparsity important?
- It’s impossible to recolor semantically
  - Two different objects share the same color

[Tan et al. 2018]
There are *still* problems

- The applied edits are not sparse enough
  - Why is sparsity important?
- It’s impossible to recolor *semantically*
  - Two different objects share the same color

[Tan et al. 2018], [Chao et al. 2023]
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow
Our Workflow

- Each edit needs to be applied in a sparse way
Our Workflow

- Each edit needs to be applied in a sparse way
- Each image-space constraint must be satisfied
Sparser Weights
Geometric Palette
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]
Geometric Palette

- Two-level decomposition [Tan et al. 2018]

\[ I = W_{RGBXY} \cdot (W_{RGB} \cdot P) \]
Sparsest Weights

EGSR 2023
THE 34TH EUROGRAPHICS SYMPOSIUM ON RENDERING
Sparsest Weights

- Observation: how to achieve maximum sparsity?
Sparsest Weights

- Observation: how to achieve maximum sparsity?
- Compute generalized barycentric coordinates with respect to RGB palette
Sparsest Weights

• Observation: how to achieve maximum sparsity?

• Compute generalized barycentric coordinates with respect to RGB palette
Sparsest Weights

- Observation: how to achieve maximum sparsity?
- Compute generalized barycentric coordinates with respect to RGB palette
Sparsest Weights

- Observation: how to achieve maximum sparsity?
- Compute generalized barycentric coordinates with respect to RGB palette
• Observation: how to achieve maximum sparsity?

• Compute generalized barycentric coordinates with respect to RGB palette
Sparsest Weights

• Observation: how to achieve maximum sparsity?
  • Compute generalized barycentric coordinates with respect to RGB palette
Sparest Weights

• Observation: how to achieve maximum sparsity?

• Compute generalized barycentric coordinates with respect to RGB palette
Sparsest Weights

- Observation: how to achieve maximum sparsity?
- Compute generalized barycentric coordinates with respect to RGB palette
Sparsest Weights

• Observation: how to achieve maximum sparsity?

• Compute generalized barycentric coordinates with respect to RGB palette

[Only RGB palette]

[Tan et al. 2018]
Sparsest Weights

• Observation: how to achieve maximum sparsity?

• Compute generalized barycentric coordinates with respect to RGB palette

[Only RGB palette]
[Tan et al. 2018]
Sparser Weights
Sparser Weights

• Add some internal vertices to the RGBXY convex hull
Sparsers Weights

- Add *some* internal vertices to the RGBXY convex hull
- Note: *Sparest* weights = treating all pixel colors as internal vertices
Sparser Weights

- Add *some* internal vertices to the RGBXY convex hull
- Note: Sparsest weights = treating all pixel colors as internal vertices
- Randomly sample? Any luck? Sample size?
**Sparser Weights**

- Add *some* internal vertices to the RGBXY convex hull

- Note: Sparsest weights = treating all pixel colors as internal vertices

- Randomly sample? Any luck? Sample size?

- Intuition: internal vertices need to be reasonably distant from each other
Spars
er Weights

• Add *some* internal vertices to the RGBXY convex hull
  • Note: Sparsest weights = treating all pixel colors as internal vertices
  • Randomly sample? Any luck? Sample size?
• Intuition: internal vertices need to be reasonably distant from each other
  • Concatenate image data with feature vectors [Aksoy et al. 2018] → $I_{RGBFEAXY}$
Sparser Weights

- Add *some* internal vertices to the RGBXY convex hull
- Note: **Spars**est weights = treating all pixel colors as internal vertices
- Randomly sample? Any luck? Sample size?
- Intuition: internal vertices need to be reasonably distant from each other
  - Concatenate image data with feature vectors [Aksoy et al. 2018] → $I_{RGBFEAXY}$
  - Internal vertices are $V_A = \text{ConvexHull}( \text{PCA}( I_{RGBFEAXY}, \text{dim}=5 ) ) |_{RGBXY}$
[Tan et al. 2018]

Ours
Ours [Tan et al. 2018]
Tan et al. 2018

Ours
[Tan et al. 2018]

Ours
Sparse Editing
Sparse Editing

- Follow [Chao et al. 2023]: Solve for minimum palette change
Sparse Editing

- Follow [Chao et al. 2023]: Solve for minimum palette change

- image-space constraints: \[ \| LAB(w_x \cdot (P + \Delta P)) - LAB(c_x) \|_2 \leq JND \]
Sparse Editing

- Follow [Chao et al. 2023]: Solve for minimum palette change

- Image-space constraints: $\|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND$
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change

• image-space constraints: \[ \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \]
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change

• image-space constraints: \[ \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \]
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change

• image-space constraints: \[ \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \]

• palette constraints: \( (P + \Delta P)[j] = c_p \)
Sparse Editing

- Follow [Chao et al. 2023]: Solve for minimum palette change

  - image-space constraints: $\|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND$

  - palette constraints: $(P + \Delta P)[j] = c_P$
Sparse Editing

- Follow [Chao et al. 2023]: Solve for minimum palette change

- Image-space constraints: \[ \| \text{LAB}(w_x \cdot (P + \Delta P)) - \text{LAB}(c_x) \|_2 \leq JND \]

- Palette constraints: \[ (P + \Delta P)[j] = c_P \]
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change

• image-space constraints: $\|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND$

• palette constraints: $(P + \Delta P)[j] = c_P$

• All together:
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change

• image-space constraints: \[ \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \]

• palette constraints: \( (P + \Delta P)[j] = c_P \)

• All together:

\[ \min_{\Delta P} \|\Delta P\|_{2,1} \]
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change
  
  • image-space constraints: \( \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \)
  
  • palette constraints: \( (P + \Delta P)[j] = c_P \)
  
  • All together:
    
    \[ \min_{\Delta P} \|\Delta P\|_{2,1} \]
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change

• image-space constraints: \[ \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \]

• palette constraints: \[ (P + \Delta P)[j] = c_P \]

• All together:

\[ \min_{\Delta P} \|\Delta P\|_{2,1} \]

• Subject to \[ 0 \leq P + \Delta P \leq 1 \] and
Sparse Editing

- Follow [Chao et al. 2023]: Solve for minimum palette change
  - image-space constraints: \[ \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \]
  - palette constraints: \[(P + \Delta P)[j] = c_p\]
- All together:
  - \[\min_{\Delta P} \|\Delta P\|_{2,1}\]
  - Subject to \[0 \leq P + \Delta P \leq 1\] and \[1\]
Sparse Editing

• Follow [Chao et al. 2023]: Solve for minimum palette change

  • image-space constraints: \( \|LAB(w_x \cdot (P + \Delta P)) - LAB(c_x)\|_2 \leq JND \) (1)

  • palette constraints: \((P + \Delta P)[j] = c_P\) (2)

• All together:

  • \( \min_{\Delta P} ||\Delta P||_{2,1} \)

  • Subject to \( 0 \leq P + \Delta P \leq 1 \) and (1) (2)
16× palette manipulations
LoCoPalettes

[Tan et al. 2018]

16× palette manipulations
LoCoPalettes

[Tan et al. 2018]

16× palette manipulations
LoCoPalettes

[Tan et al. 2018]
LoCoPalettes

[Tan et al. 2018]

16\times palette manipulations

29\times palette manipulations
Local Control
Palette and Weight Hierarchy
Definition
Palette and Weight Hierarchy

Definition

- Hierarchical data structure to support local edits
Palette and Weight Hierarchy

Definition

• Hierarchical data structure to support local edits
  • Each node: local palette, local weights, local soft mask
Palette and Weight Hierarchy

Definition

• Hierarchical data structure to support local edits
• Each node: local palette, local weights, local soft mask
• DETR [Carion et al. 2020]'s panoptic segmentation
Palette and Weight Hierarchy

Definition

• Hierarchical data structure to support local edits
• Each node: local palette, local weights, local soft mask
• DETR [Carion et al. 2020]'s panoptic segmentation
  • Root $\rightarrow$ Classes $\rightarrow$ Instances
Palette and Weight Hierarchy

Definition

- Hierarchical data structure to support local edits
- Each node: local palette, local weights, local soft mask
- DETR [Carion et al. 2020]'s panoptic segmentation
  - Root → Classes → Instances
Palette and Weight Hierarchy

Definition

• Hierarchical data structure to support local edits
  • Each node: local palette, local weights, local soft mask
• DETR [Carion et al. 2020]'s panoptic segmentation
  • Root → Classes → Instances
Palette and Weight Hierarchy

Definition

- Hierarchical data structure to support local edits
  - Each node: local palette, local weights, local soft mask
  - DETR [Carion et al. 2020]'s panoptic segmentation
    - Root → Classes → Instances
Palette and Weight Hierarchy

Definition

- Hierarchical data structure to support local edits
- Each node: local palette, local weights, local soft mask
- DETR [Carion et al. 2020]'s panoptic segmentation
  - Root $\rightarrow$ Classes $\rightarrow$ Instances
Palette and Weight Hierarchy

Definition

• Hierarchical data structure to support local edits
  • Each node: local palette, local weights, local soft mask
  • DETR [Carion et al. 2020]'s panoptic segmentation
    • Root → Classes → Instances
Palette and Weight Hierarchy

Definition

- Hierarchical data structure to support local edits
- Each node: local palette, local weights, local soft mask
- DETR [Carion et al. 2020]'s panoptic segmentation
  - Root → Classes → Instances
Editing Example

$P_i, W_i \quad P_0 \quad P_1 \quad P_2 \quad P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
Editing Example

Image $I$

$P_i, W_i$

$P_0$

$P_1$

$P_2$

$P_3$
 Editing Example

$P_i, W_i$
Sparse Editing with Hierarchy

Palette splitting rules
Sparse Editing with Hierarchy

Palette splitting rules

- Store booleans to track node activations
Sparse Editing \textit{with} Hierarchy

Palette splitting rules

- Store booleans to track node activations

\[\begin{align*}
    a_0 &= \text{True} \\
    a_1 &= \text{False} \\
    a_2 &= \text{False} \\
    a_3 &= \text{False} \\
    a_4 &= \text{False} \\
    a_5 &= \text{False}
\end{align*}\]
Sparse Editing with Hierarchy

Palette splitting rules

• Store booleans to track node activations

• A new image-space constraint starts at the most local (deepest) active node containing it

\[
\begin{align*}
a_0 &= \text{True} \\
a_1 &= \text{False} \\
a_2 &= \text{False} \\
a_3 &= \text{False} \\
a_4 &= \text{False} \\
a_5 &= \text{False}
\end{align*}
\]
Sparse Editing with Hierarchy
Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it

\[ a_0 = \text{True} \]
\[ a_1 = \text{False} \]
\[ a_2 = \text{False} \]
\[ a_3 = \text{False} \]
\[ a_4 = \text{False} \]
\[ a_5 = \text{False} \]
Sparse Editing with Hierarchy
Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it
Sparse Editing with Hierarchy
Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it

\[ a_0 = \text{True} \]
\[ a_1 = \text{False} \]
\[ a_2 = \text{False} \]
\[ a_3 = \text{False} \]
\[ a_4 = \text{False} \]
\[ a_5 = \text{False} \]
Sparse Editing with Hierarchy

Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it

Run Sparse Editing Optimization!
Sparse Editing with Hierarchy

Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it
- Optimization fails → activate the next deeper node containing it

![Diagram showing the process of Sparse Editing with Hierarchy with booleans and node activations]

- $a_0 = True$
- $a_1 = False$
- $a_2 = False$
- $a_3 = False$
- $a_4 = False$
- $a_5 = False$

Run Sparse Editing Optimization!
Sparse Editing with Hierarchy

Palette splitting rules

- Store booleans to track node activations

- A new image-space constraint starts at the most local (deepest) active node containing it

- Optimization fails → activate the next deeper node containing it
Sparse Editing with Hierarchy

Palette splitting rules

- Store booleans to track node activations

- A new image-space constraint starts at the most local (deepest) active node containing it

- Optimization fails $\rightarrow$ activate the next deeper node containing it

$\begin{align*}
a_0 &= \text{True} \\
a_1 &= \text{False} \\
a_2 &= \text{False} \\
a_3 &= \text{False} \\
a_4 &= \text{False} \\
a_5 &= \text{False}
\end{align*}$

Run Sparse Editing Optimization!
Sparse Editing with Hierarchy
Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it
- Optimization fails → activate the next deeper node containing it

$\begin{align*}
a_0 &= \text{True} \\
a_1 &= \text{False} \\
a_2 &= \text{False} \\
a_3 &= \text{False} \\
a_4 &= \text{False} \\
a_5 &= \text{False}
\end{align*}$

Run Sparse Editing Optimization!
Sparse Editing with Hierarchy

Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it
- Optimization fails → activate the next deeper node containing it

Run Sparse Editing Optimization!

- $a_0 = \text{True}$
- $a_1 = \text{False}$
- $a_2 = \text{False}$
- $a_3 = \text{False}$
- $a_4 = \text{False}$
- $a_5 = \text{False}$
Sparse Editing *with* Hierarchy

Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it
- Optimization fails $\rightarrow$ activate the next deeper node containing it
Sparse Editing *with Hierarchy*

Palette splitting rules

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it
- Optimization fails $\rightarrow$ activate the next deeper node containing it

$$a_0 = \text{True} \quad a_1 = \text{False} \quad a_2 = \text{False} \quad a_3 = \text{False} \quad a_4 = \text{False} \quad a_5 = \text{False}$$
Sparse Editing *with Hierarchy*

**Palette splitting rules**

- Store booleans to track node activations
- A new image-space constraint starts at the most local (deepest) active node containing it
- Optimization fails $\rightarrow$ activate the next deeper node containing it

$\begin{align*}
a_0 &= \text{True} \\
a_1 &= \text{False} \\
a_2 &= \text{False} \\
\end{align*}$

$\begin{align*}
a_3 &= \text{False} \\
a_4 &= \text{False} \\
a_5 &= \text{False} \\
\end{align*}$

Run Sparse Editing Optimization!
Palette and Weight Hierarchy

Reconstruction

\[
\begin{align*}
    a_0 &= \text{True} \\
    a_1 &= \text{True} \\
    a_2 &= \text{False} \\
    a_3 &= \text{False} \\
    a_4 &= \text{False} \\
    a_5 &= \text{False}
\end{align*}
\]
Palette and Weight Hierarchy

Reconstruction

- How to reconstruct the edited image under the hierarchy?
Palette and Weight Hierarchy

Reconstruction

• How to reconstruct the edited image under the hierarchy?

• Alpha compositing over activated nodes
Palette and Weight Hierarchy

Reconstruction

- How to reconstruct the edited image under the hierarchy?
- Alpha compositing over activated nodes
- Leaf palettes are more local!
Palette and Weight Hierarchy

Reconstruction

• How to reconstruct the edited image under the hierarchy?
• Alpha compositing over activated nodes
• Leaf palettes are more local!
Palette and Weight Hierarchy

Palette Propagation

\[ a_0 = \text{True} \]
\[ a_1 = \text{True} \]
\[ a_2 = \text{False} \]
\[ a_3 = \text{False} \]
\[ a_4 = \text{False} \]
\[ a_5 = \text{False} \]
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if **not activated**
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if **not activated**

\[ \min_{P_c} \| W_c \cdot P_c - W_p \cdot P_p \|_2^2 \]

- \( a_0 = \text{True} \)
- \( a_1 = \text{True} \)
- \( a_2 = \text{False} \)
- \( a_3 = \text{False} \)
- \( a_4 = \text{False} \)
- \( a_5 = \text{False} \)
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if **not activated**

- \[ \min_{P_c} \| W_c \cdot P_c - W_p \cdot P_p \|^2 \]

- Subject to \( 0 \leq P_c \leq 1 \)
Palette and Weight Hierarchy

Palette Propagation

• Propagate changes towards leaf palettes if **not activated**

\[
\min_{P_c} \| W_c \cdot P_c - W_p \cdot P_p \|^2_2
\]

• Subject to \( 0 \leq P_c \leq 1 \)

\[a_0 = \text{True}\]
\[a_1 = \text{True}\]
\[a_2 = \text{False}\]
\[a_3 = \text{False}\]
\[a_4 = \text{False}\]
\[a_5 = \text{False}\]
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if **not activated**

$$\min_{P_c} \left\| W_c \cdot P_c - W_p \cdot P_p \right\|_2^2$$

- Subject to $0 \leq P_c \leq 1$

Small $p \times p$ quadratic programming problem!
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if **not activated**

$$\min_{\mathbf{P}_c} \| \mathbf{W}_c \cdot \mathbf{P}_c - \mathbf{W}_p \cdot \mathbf{P}_p \|^2_2$$

- Subject to $0 \leq \mathbf{P}_c \leq 1$

Small $\#p \times \#p$ quadratic programming problem!
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if **not activated**

\[
\min_{P_c} \|W_c \cdot P_c - W_p \cdot P_p\|_2^2
\]

- Subject to \(0 \leq P_c \leq 1\)

Small \#p x \#p quadratic programming problem!
Palette and Weight Hierarchy

Palette Propagation

• Propagate changes towards leaf palettes if **not activated**

\[
\min_{P_c} \left\| W_c \cdot P_c - W_p \cdot P_p \right\|_2^2
\]

• Subject to \( 0 \leq P_c \leq 1 \)

Small \( p \times p \) quadratic programming problem!
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if not activated

\[
\min_{P_c} \| W_c \cdot P_c - W_p \cdot P_p \|^2_2
\]

- Subject to \( 0 \leq P_c \leq 1 \)

Small #p x #p quadratic programming problem!
Palette and Weight Hierarchy

Palette Propagation

- Propagate changes towards leaf palettes if **not activated**

\[
\min \left\| W_c \cdot P_c - W_p \cdot P_p \right\|_2^2
\]

- Subject to \(0 \leq P_c \leq 1\)

Small \(p \times p\) quadratic programming problem!
LoCoPalettes [Tan et al. 2018]
LoCoPalettes

Input

[LoCoPalettes](Tan et al. 2018) 13x palette manipulations
Input
Input

[Tan et al. 2018]
Input  |  LoCoPalettes  |  [Tan et al. 2018]
LoCoPalettes

[Tan et al. 2018]
LoCoPalettes

[Tan et al. 2018]
LoCoPalettes

[Tan et al. 2018]
Input

LoCoPalettes

[Tan et al. 2018]

17x palette manipulations
Conclusion
Conclusion

- LoCoPalettes provides local control for palette-based editing
Conclusion

- LoCoPalettes provides local control for palette-based editing
- Extends [Chao et al. 2023] to a palette hierarchy
Conclusion

• LoCoPalettes provides local control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
  • Relies on high-quality hierarchical segmentation
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  
  • Extends [Chao et al. 2023] to a palette hierarchy
  
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
  
  • Relies on high-quality hierarchical segmentation
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
  • Relies on high-quality hierarchical segmentation
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  - Extends [Chao et al. 2023] to a palette hierarchy
  - Improves weight sparsity while maintaining spatial coherence

• **Limitations**
  - Relies on high-quality hierarchical segmentation
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
  • Relies on high-quality hierarchical segmentation
Conclusion

- LoCoPalettes provides local control for palette-based editing
  - Extends [Chao et al. 2023] to a palette hierarchy
  - Improves weight sparsity while maintaining spatial coherence

- Limitations
  - Relies on high-quality hierarchical segmentation

- Future Work
Conclusion

• LoCoPalettes provides local control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
  • Relies on high-quality hierarchical segmentation

• Future Work
  • Palette-based video editing, e.g. [Du et al. 2021]
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
  • Relies on high-quality hierarchical segmentation

• Future Work
  • Palette-based video editing, e.g. [Du et al. 2021]
  • Speed up local palette computation from a global palette
Conclusion

• LoCoPalettes provides **local** control for palette-based editing
  • Extends [Chao et al. 2023] to a palette hierarchy
  • Improves weight sparsity while maintaining spatial coherence

• Limitations
  • Relies on high-quality hierarchical segmentation

• Future Work
  • Palette-based video editing, e.g. [Du et al. 2021]
  • Speed up local palette computation from a global palette
  • Text-guided color manipulations
Thank you

• Project page: https://cragl.cs.gmu.edu/locopalettes/

• Code and data: https://github.com/tedchao/LoCoPalettes

• Financial support
  • Adobe
Evaluation

KNN Matting [Chen et al. 2013]
# Sparsity Evaluation

<table>
<thead>
<tr>
<th>Sparsity Estimate:</th>
<th>Tan et al. [2016]</th>
<th>Aksoy et al. [2017]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tan et al. [2018]</td>
<td>Ours</td>
</tr>
<tr>
<td>Mountain</td>
<td>0.2630</td>
<td>0.2586</td>
</tr>
<tr>
<td>Birds</td>
<td>0.2670</td>
<td>0.2614</td>
</tr>
<tr>
<td>Colorful</td>
<td>0.2549</td>
<td>0.2511</td>
</tr>
<tr>
<td>Boy</td>
<td>0.2676</td>
<td>0.2638</td>
</tr>
</tbody>
</table>