SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

THE PREMIER CONFERENCE & EXHIBITION ON COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

IMAGE VECTORIZATION AND EDITING VIA LINEAR GRADIENT LAYER DECOMPOSITION

ZHENG-JUN DU — QINGHAI UNIVERSITY, TSINGHUA UNIVERSITY
LIANG-FU KANG — TSINGHUA UNIVERSITY
JIANCHAO TAN — KUAISHOU TECHNOLOGY
YOTAM GINGOLD — GEORGE MASON UNIVERSITY
KUN XU — TSINGHUA UNIVERSITY

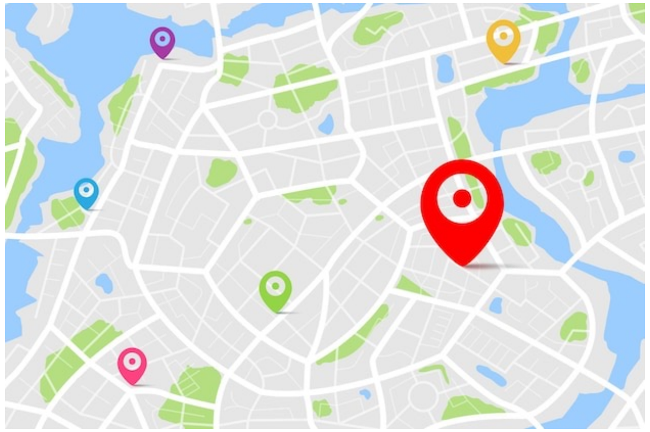
Vector graphics are editable and resolution independent

- Widely used in illustrations, fonts, logos, and map design.

Font

panda
funny lazy

Logo



Map

Multi-layer vectorization

- Multi-layer vectorization aims to decompose a raster image into multiple semi-transparent layers.

Input image

Multi-layer vectorization

- Multi-layer vectorization aims to decompose a raster image into multiple semi-transparent layers.

Input image

Decomposed semi-transparent layers

Multi-layer vectorization

- The decomposed layers reconstruct the input raster image by alpha blending.

Input image

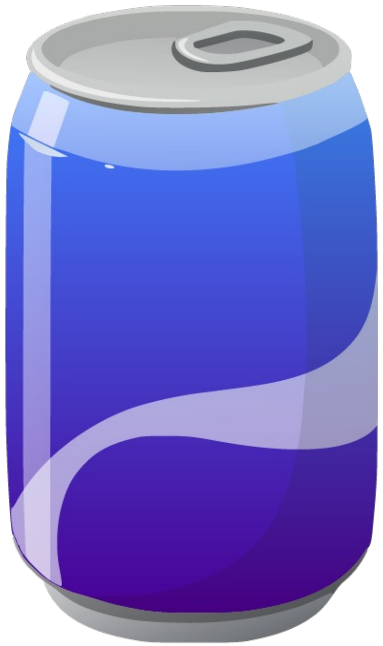
Reconstruction

Multi-layer vectorization

- Applications: recoloring, insert-remove-replace edits, etc.

Input image

Reconstruction



Recoloring

inserting objects

Related work: Multi-layer vectorization

- **Vectorizing Bitmaps into Semi-Transparent Gradient Layers**
  Richardt et al. EGSR 2014

- **Photo2ClipArt: Image Abstraction and Vectorization Using Layered Linear Gradients**
  Favreau et al. SIGGRAPH Asia 2017

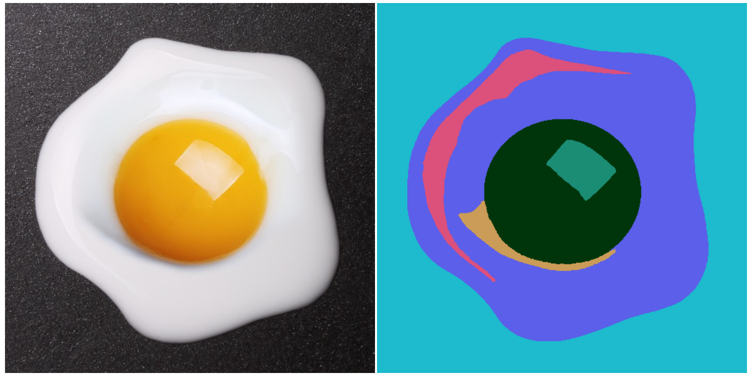
These methods require user interaction and may get stuck in local minima.

Our contributions

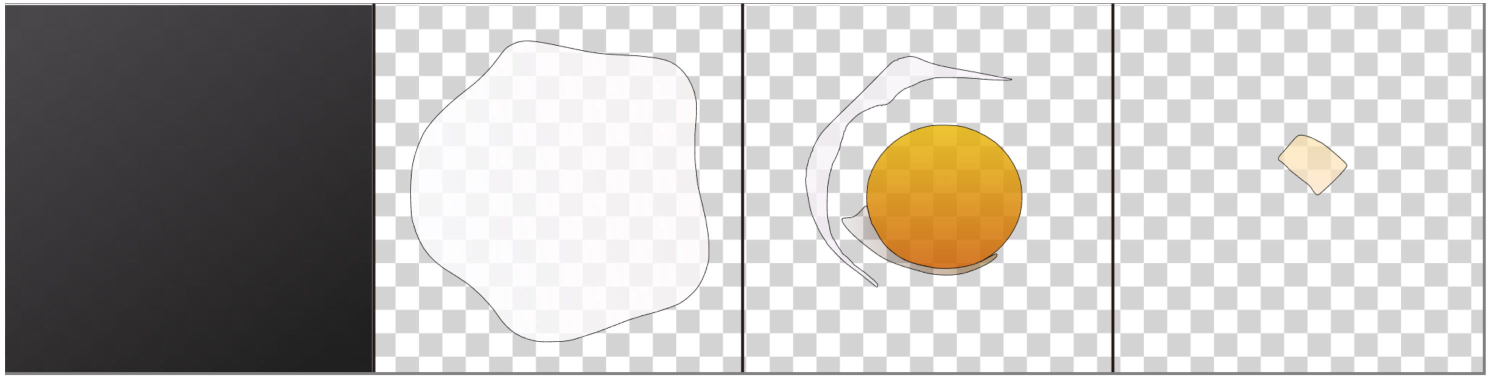
- A fully automatic approach to layer decomposition from a segmented input image.
- A drastically reduced search space via perceptually-motivated constraints.

Our approach

- Input: a segmented raster image.
- Output: a set of semi-transparent layers $L = \{L_1, L_2, \ldots, L_n\}$.



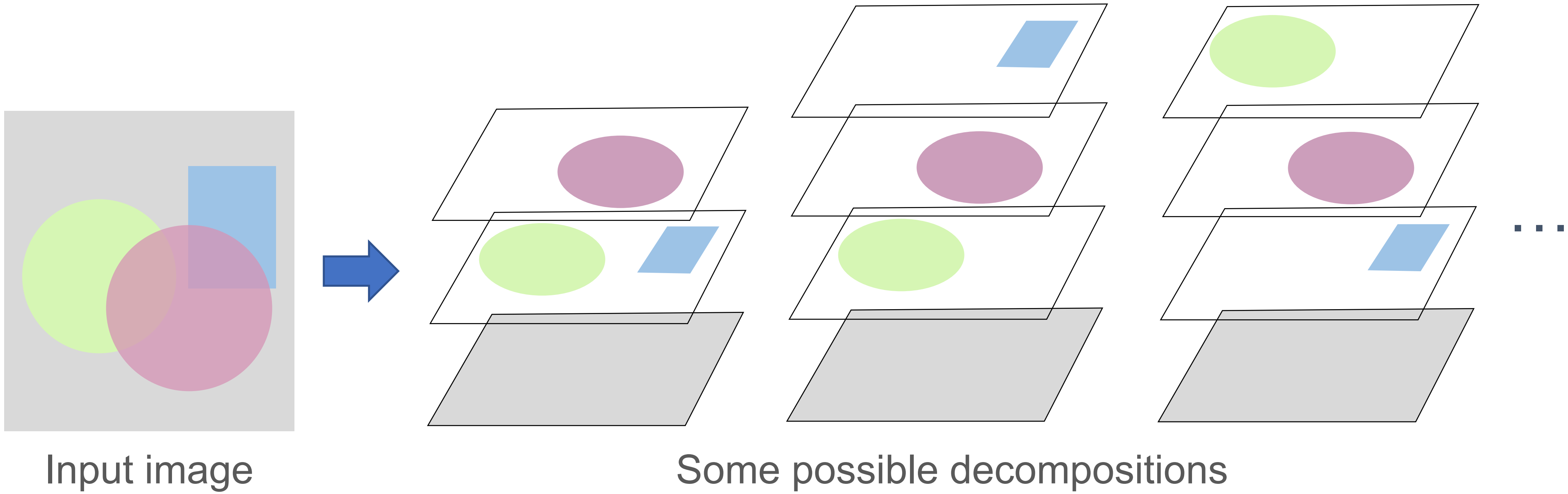
Input image & its segmentation



Output ordered semi-transparent layers (bottom to top)

Main challenge

- Multi-layer vectorization is severely under-constrained.

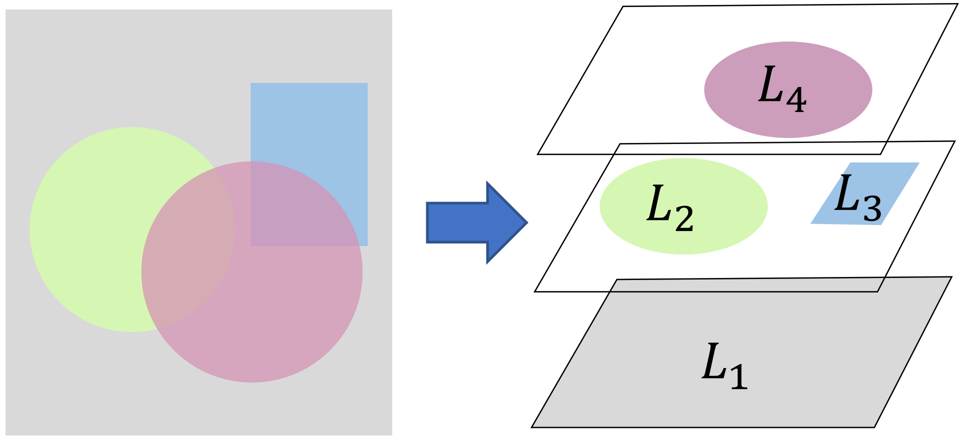


Given #layer = l, #pixel = n, #decomposition = 2^{nl}.

Given #layer = l, #region = m, #decomposition = 2^{ml}.

Main challenge

- For a decomposition, two types of parameters are difficult to solve
  - Layer configuration: the number of layers, their masks, and stack order.
  - Layer parameters: linear gradient parameters of each layer.



Input image

Layers

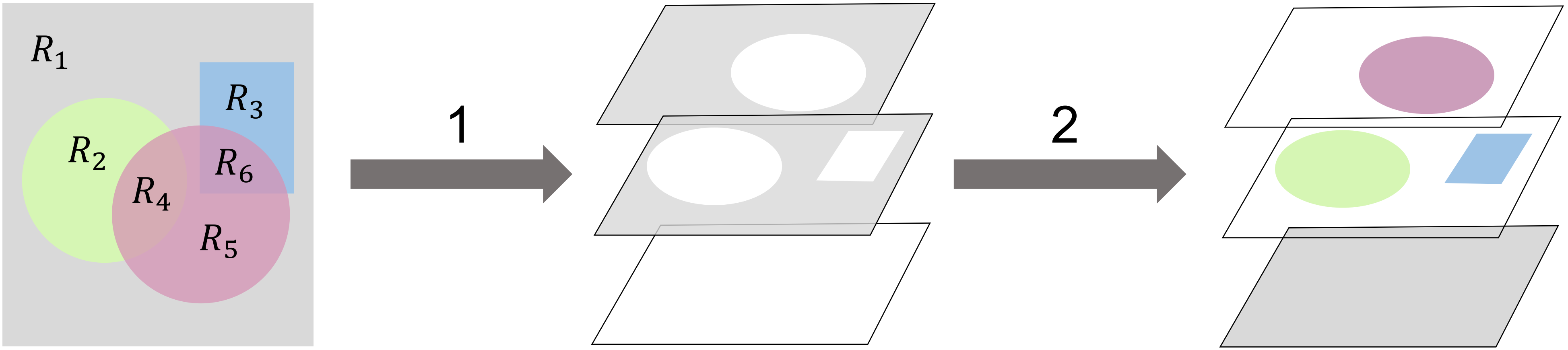
Layers: $[L_1, L_2, L_3, L_4]$
Layer masks
Stack order: $L_1 \rightarrow L_2 \rightarrow L_4,\ L_1 \rightarrow L_3 \rightarrow L_4$

$$C_{\mathrm{RGBA}}(\mathrm{p}) = \mathrm{c}_0 + (\mathrm{p} \cdot \mathrm{n}) \cdot \mathrm{c}_{\mathrm{g}}$$

Layer configuration & layer parameters

Our approach

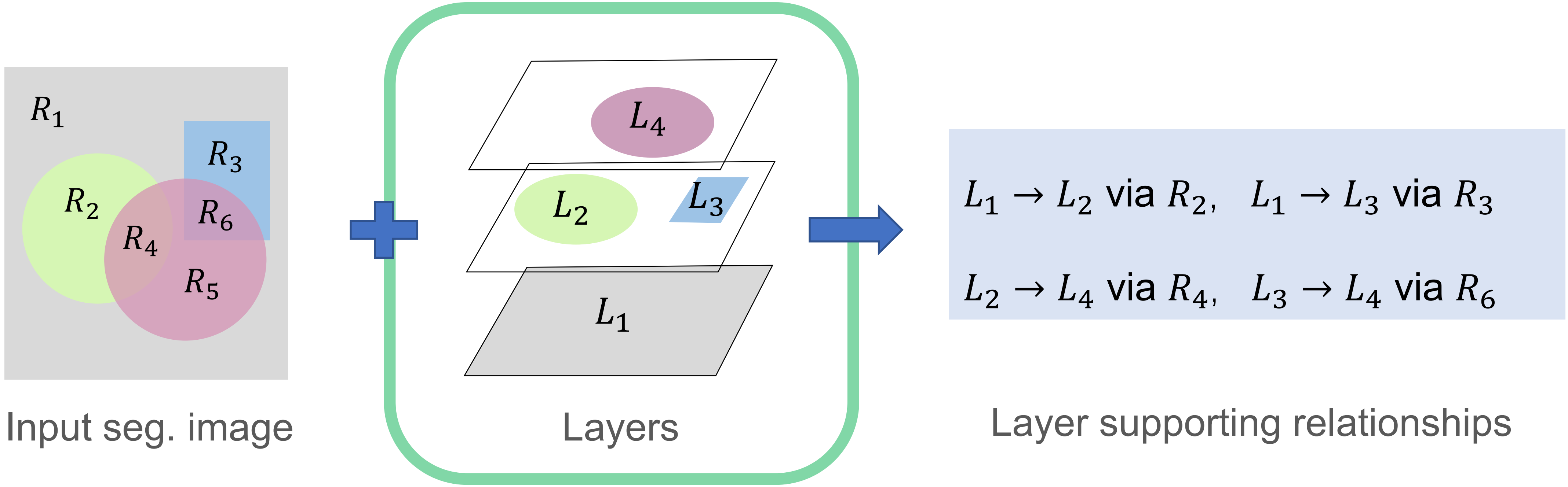
- We propose a two-step solution to multi-layer vectorization
  1. Determine the layer configuration.
  2. Estimate the layer parameters.



Input seg. image　　Layer configuration　　Output parameterized layers

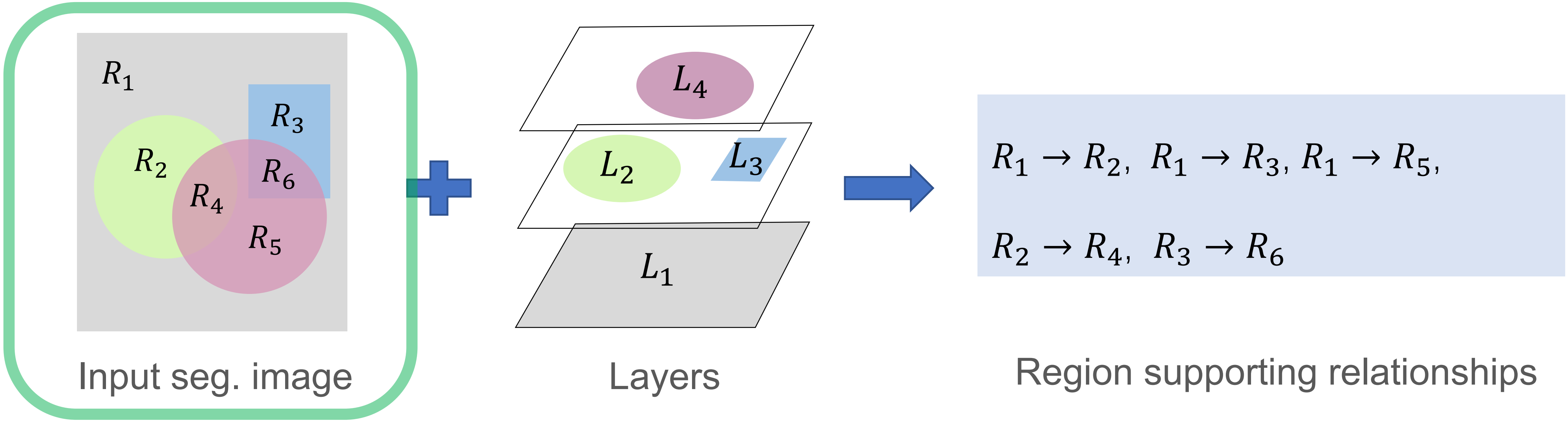
Basic concepts

- Layer support: A lower layer L_i supports ($\rightarrow$) a higher layer L_j if
  1) L_i and L_j overlap.
  2) No regions in their overlap are covered by a layer in-between.



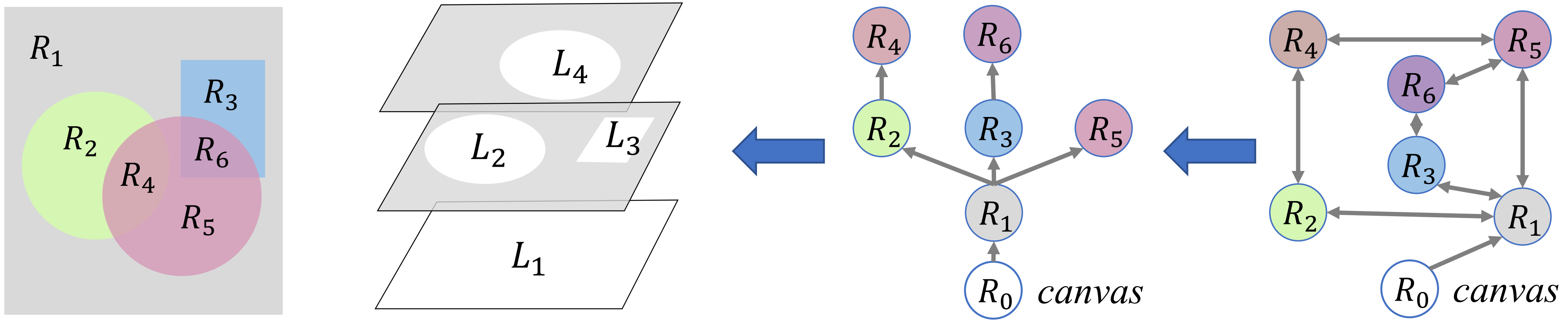
Basic concepts

- Region support: Region R_i supports ($\rightarrow$) region R_j if
  1) R_i and R_j are adjacent.
  2) R_i's top layer L_i supports R_j's top layer L_j.
  3) R_j is in the overlap of L_i and L_j.



Determining the layer configuration

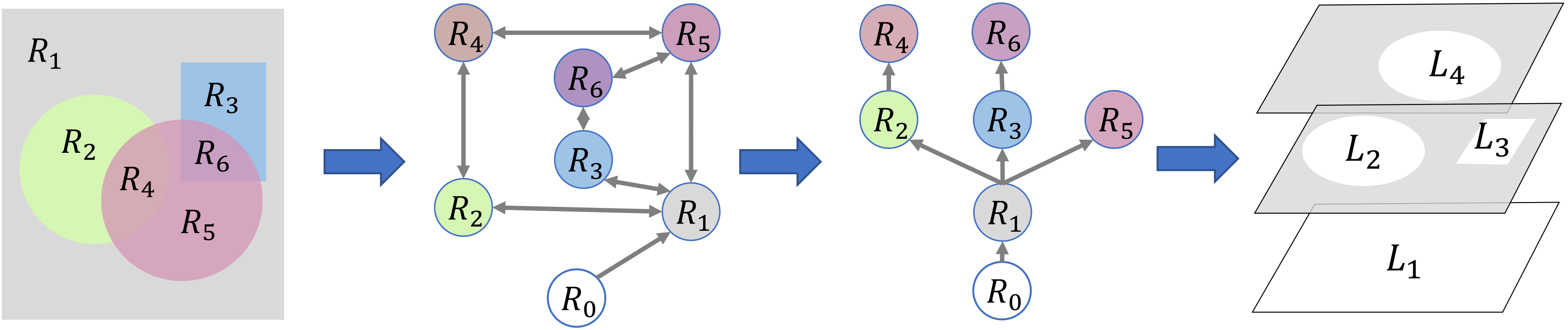
- Key observation
  - A layer configuration can be expressed as a region supporting tree, which can be enumerated from the region adjacency graph.



Input seg. image | Layer configuration | Region supporting tree | Region adj. graph

Determining the layer configuration

- Pipeline

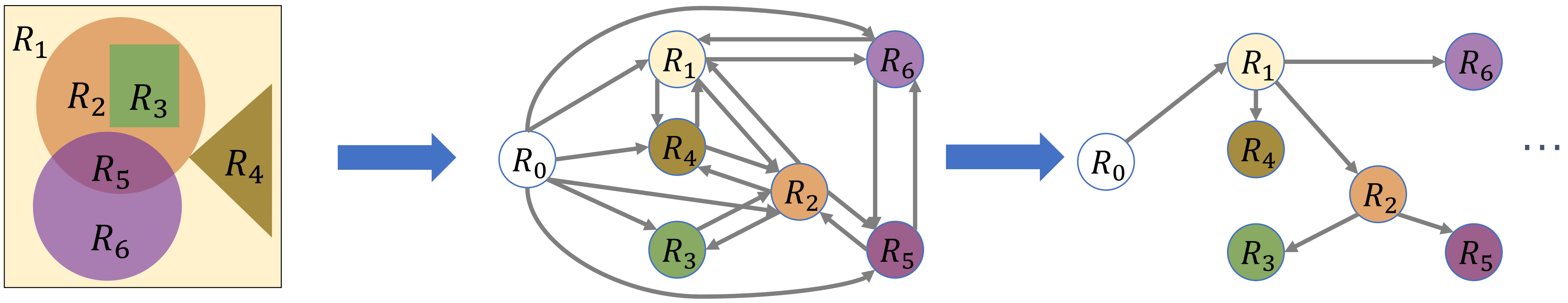


Input seg. image | Build the region adjacency graph | Enumerate region supporting trees | Merge layers with X-junction hints

The region adjacency graph implies the space of region supporting trees.

Building the region adjacency graph

- Directly enumerating over all region supporting trees from the initial region adjacency graph is rather expensive.

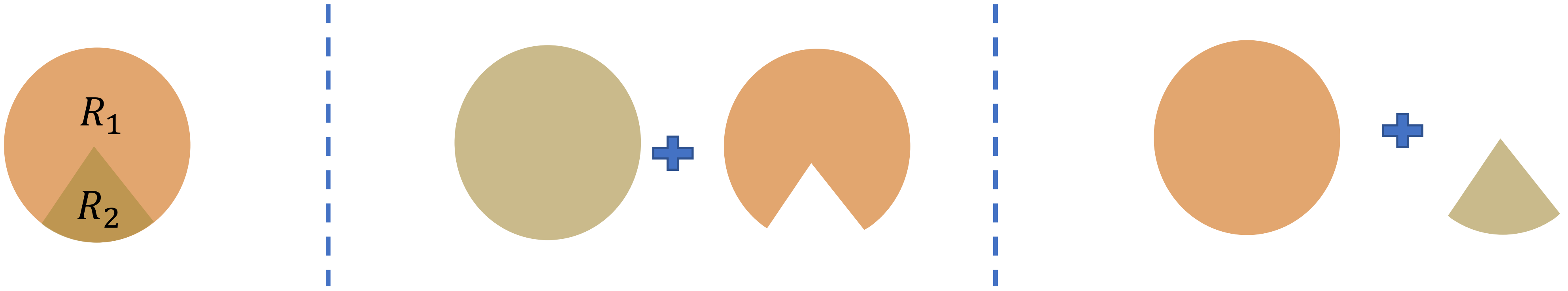


Input seg. image | Initial region adj. graph | Many region supporting trees

How to reduce the search space of region supporting trees?

Simplifying the region adjacency graph

- Size rule: very small regions can't support large ones.



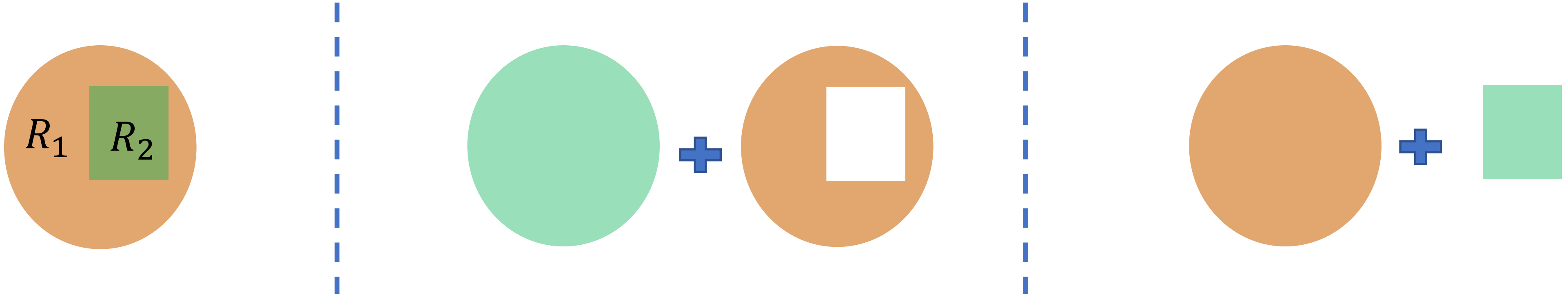
Input

Decomposition without rule ($R_2 \rightarrow R_1$)

Decomposition with rule ($R_1 \rightarrow R_2$)

Simplifying the region adjacency graph

- Surrounding rule: surrounded regions (islands) can't support their surroundings.



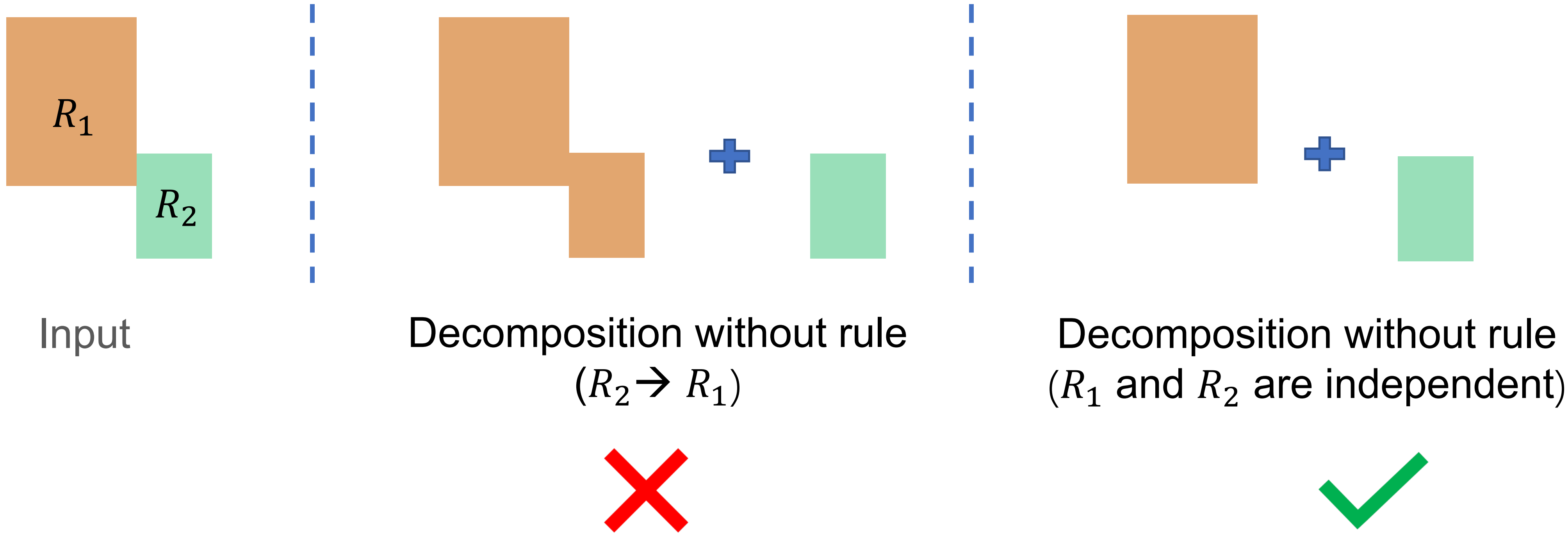
Input

Decomposition without rule
$(R_2 \rightarrow R_1)$

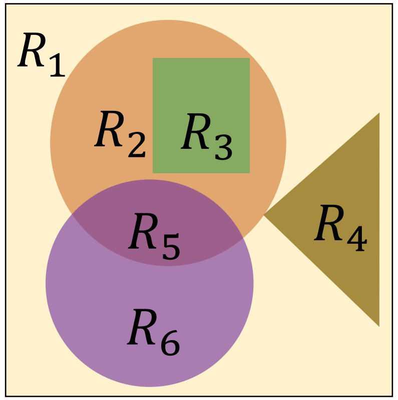
Decomposition with rule
$(R_1 \rightarrow R_2)$

Simplifying the region adjacency graph

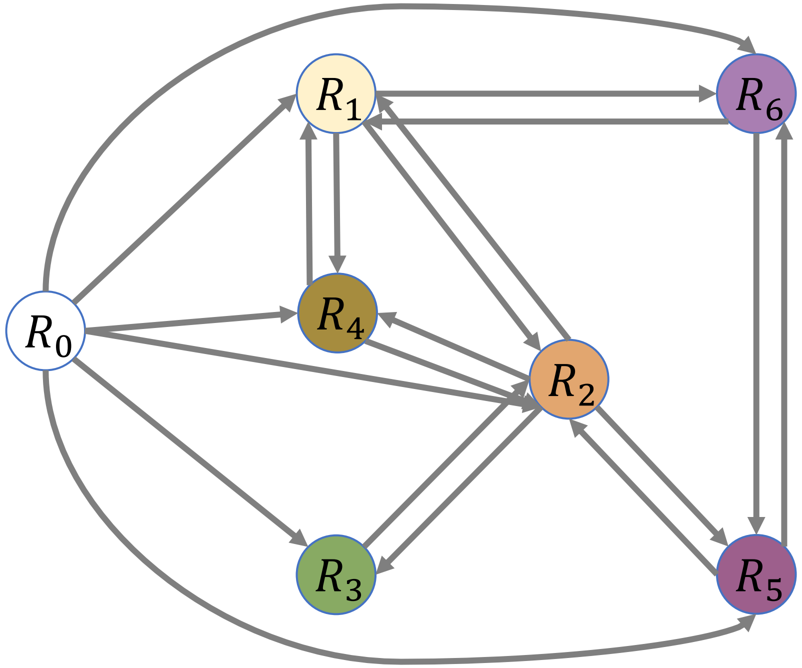
- Adjacent strength rule: Two regions sharing a very short boundary can't support each other.



An example

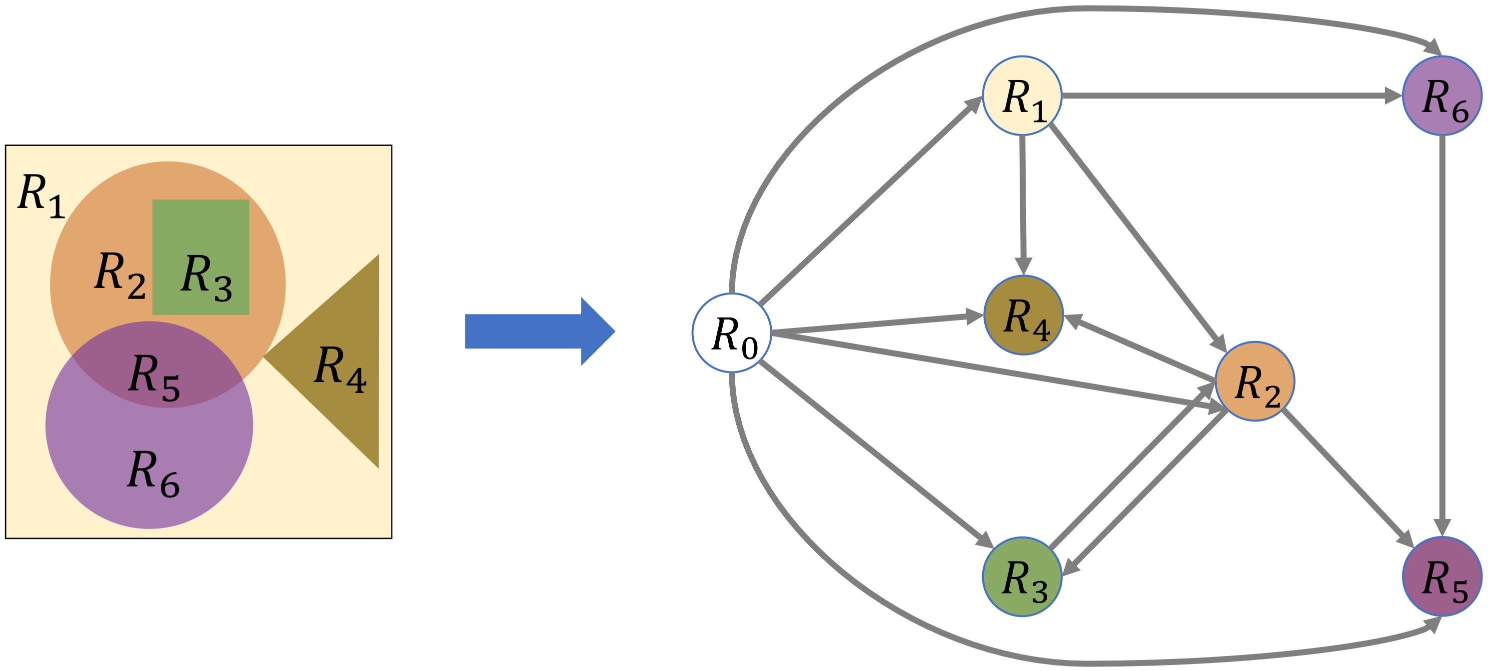


Input seg. image



Initial region adjacency graph

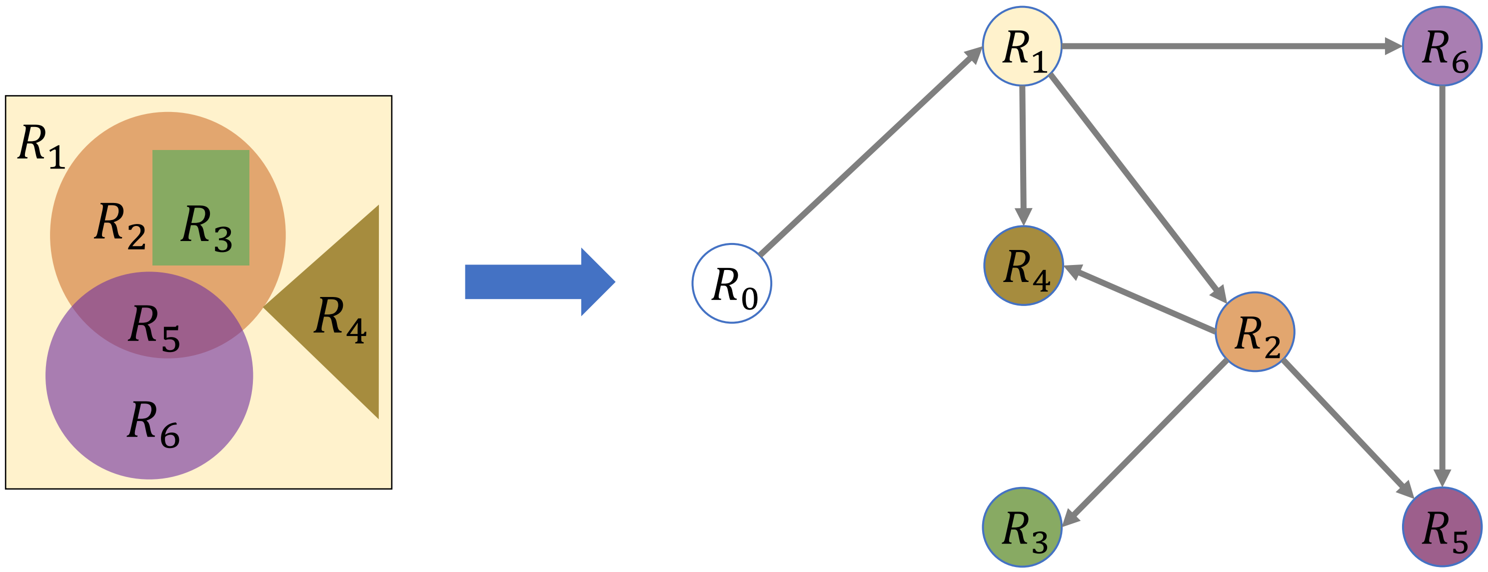
An example



Input seg. image

Simplified with the size rule

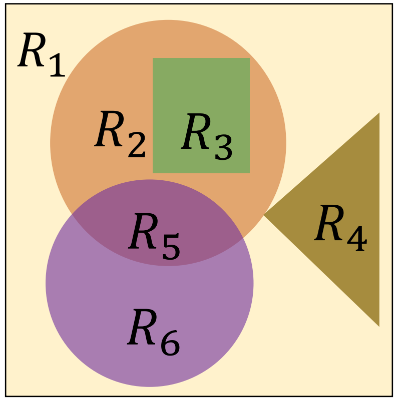
An example



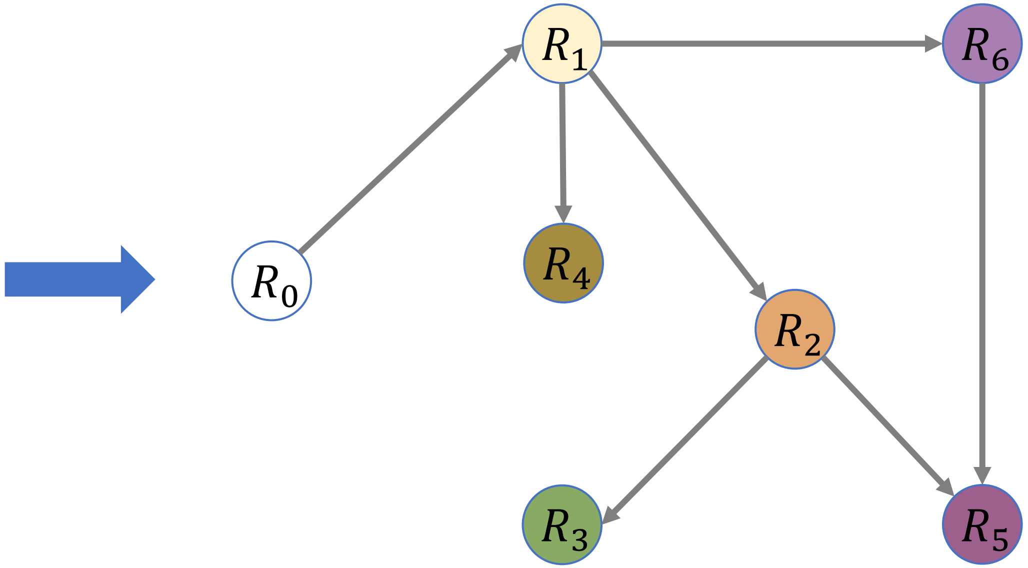
Input seg. image

Simplified with the size rule
surrounding rule

An example



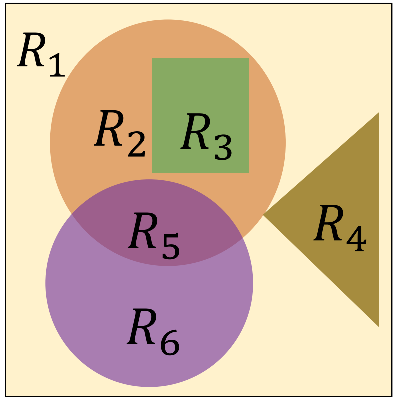
Input seg. image



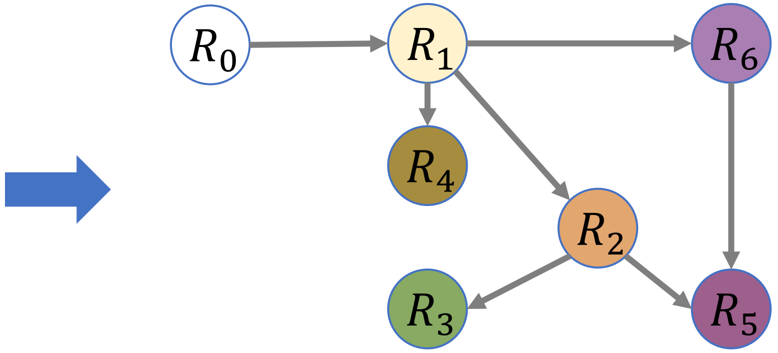
Simplified with the size rule
surrounding rule
adjacent strength rule

Enumerating region support trees

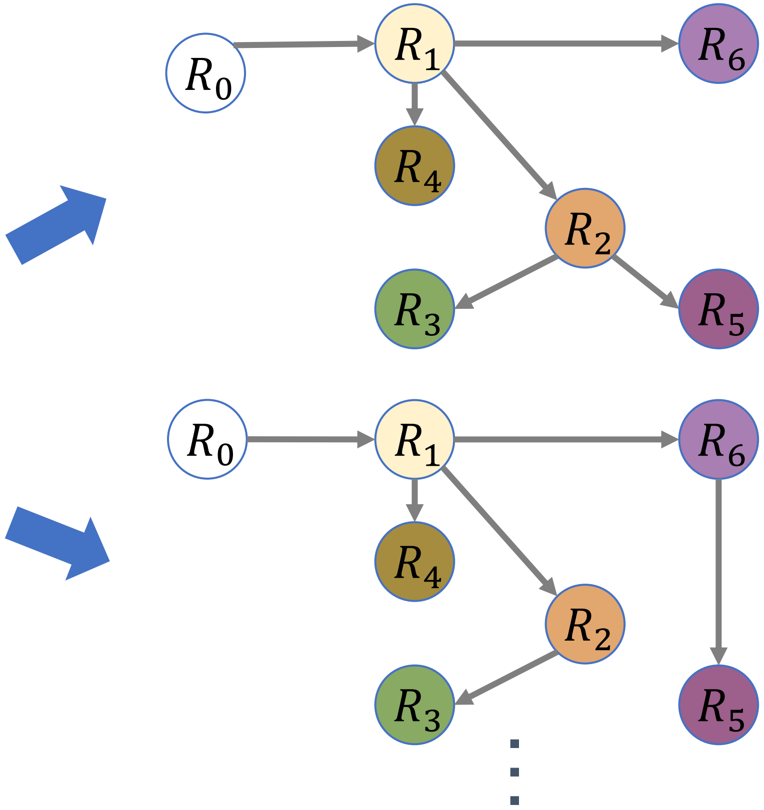
- Enumerating the spanning trees (region support trees) from the simplified region adjacency graph.



Input seg. image



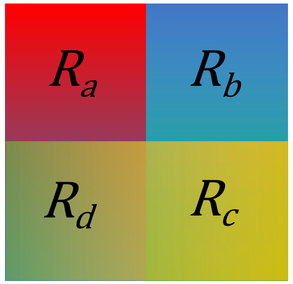
Simplified region adj. graph



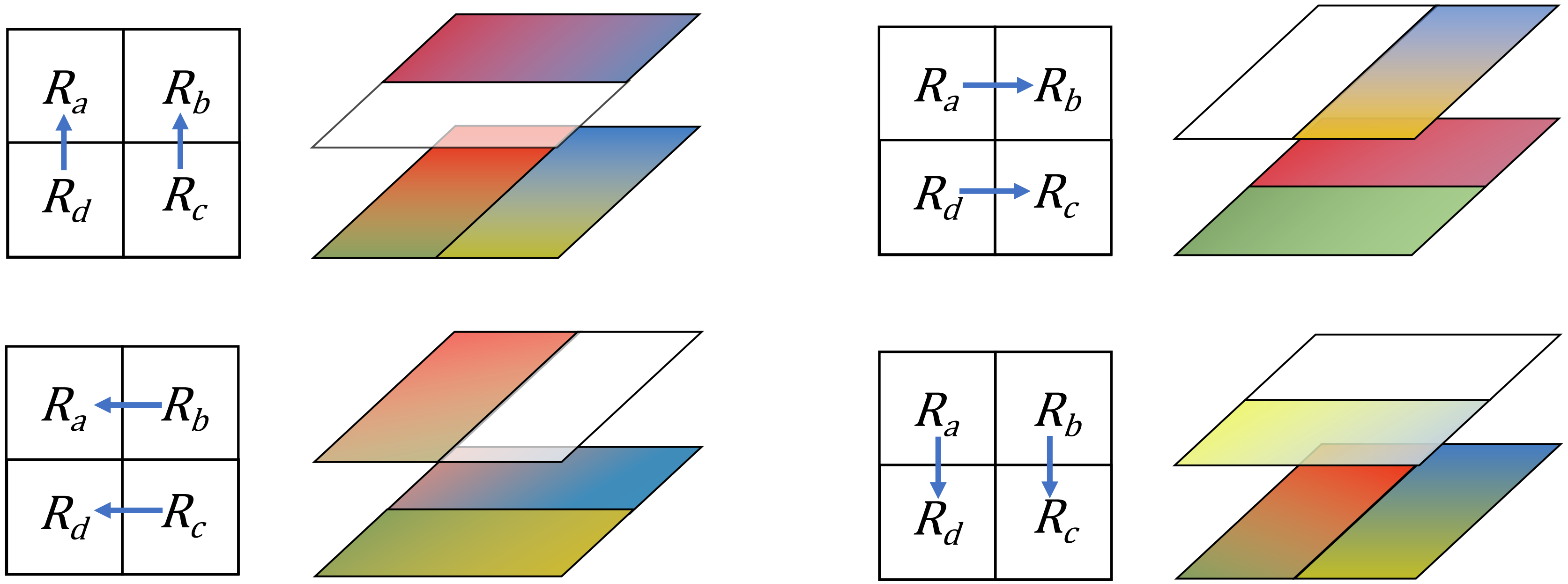
Region supporting trees

Merging layers with X-junction hints

- An X-junction occurs when a semi-transparent layer runs across 2 other layers [Meteli 1974, 1985].

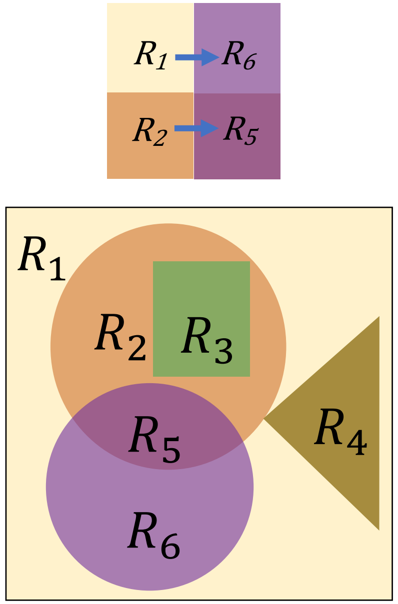


An X-junction

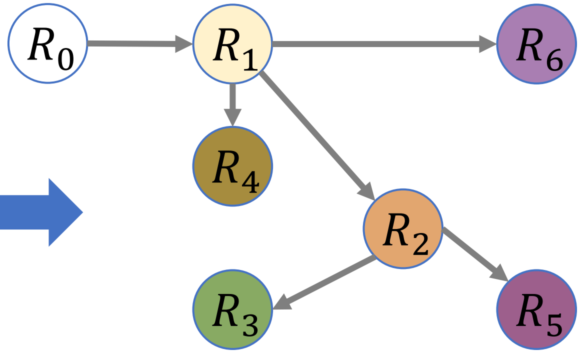


Regions' supporting relationships and layer configurations

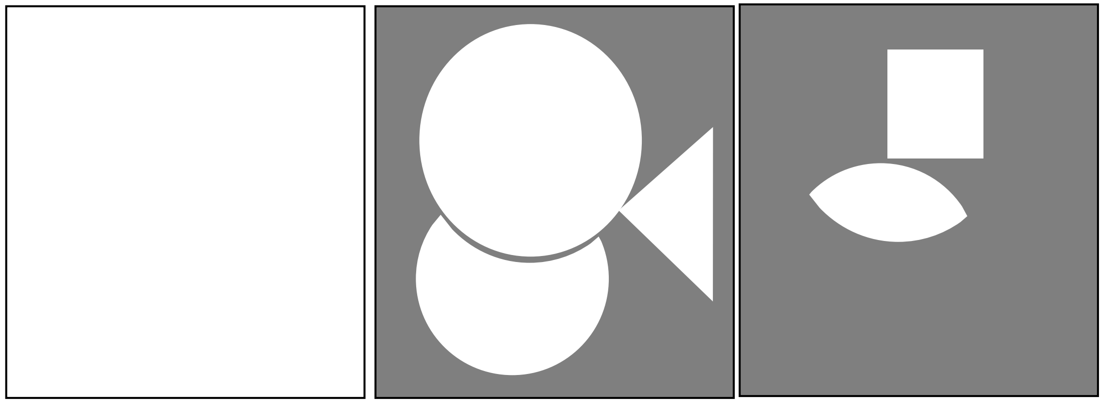
Merging layers with X-junction hints



Input seg. image
& an X-junction

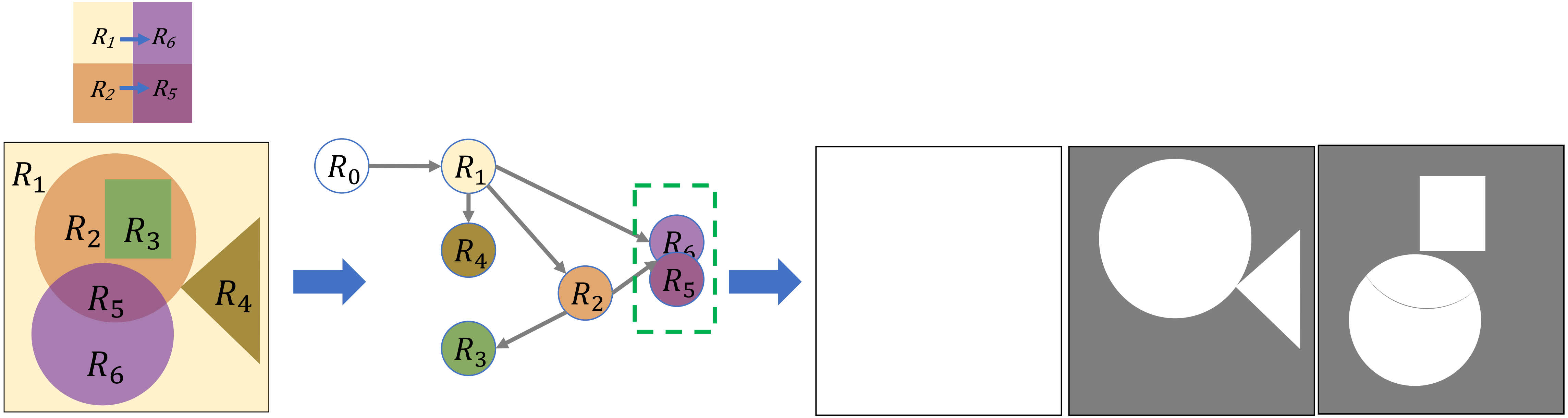


A region supporting tree



Layer configuration before layer merging

Merging layers with X-junction hints



Input seg. image & an X-junction

Merging the top layers of R_5 & R_6

Layer configuration after layer merging

Estimating layer parameters

- Energy

$$E = w_r E_{\text{recon}} + w_g E_{\text{gamut}} + w_c E_{\text{compact}}$$

Reconstruction loss: $E_{\text{recon}} = \frac{1}{N} \sum_{\text{p}} \| I^n_{\text{RGB}}(\text{p}) - I_{\text{RGB}}(\text{p}) \|^2$

Input

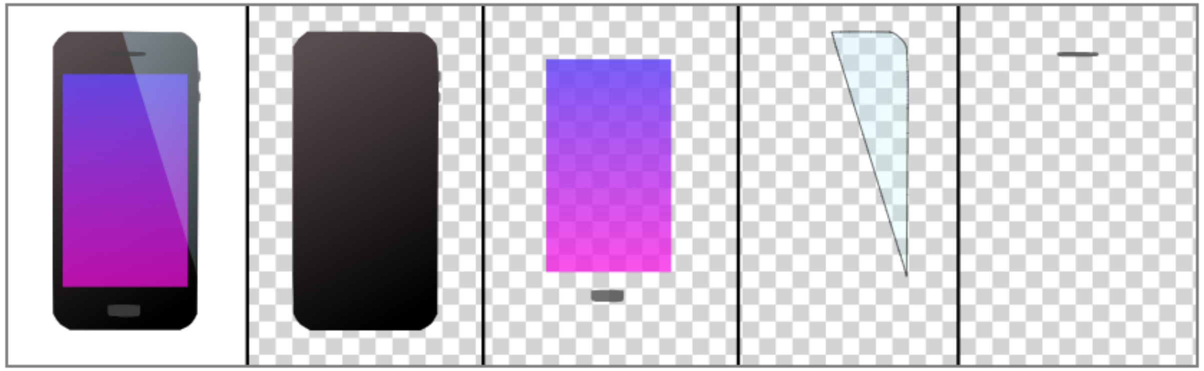
Recon. image

Estimating layer parameters

- Energy

$$E = w_r E_{\mathrm{recon}} + w_g E_{\mathrm{gamut}} + w_c E_{\mathrm{compact}}$$

Gamut loss: $E_{\mathrm{gamut}} = \frac{1}{M}\sum_{i=1}^{n}\sum_{\mathrm{p}\in L_i}\left\|C_{\mathrm{RGB}}^{i}(\mathrm{p}) - \overline{C_{\mathrm{RGB}}(\mathrm{p})}\right\|^2 + \left\|C_{\mathrm{A}}^{i}(\mathrm{p}) - \overline{C_{\mathrm{A}}(\mathrm{p})}\right\|^2$

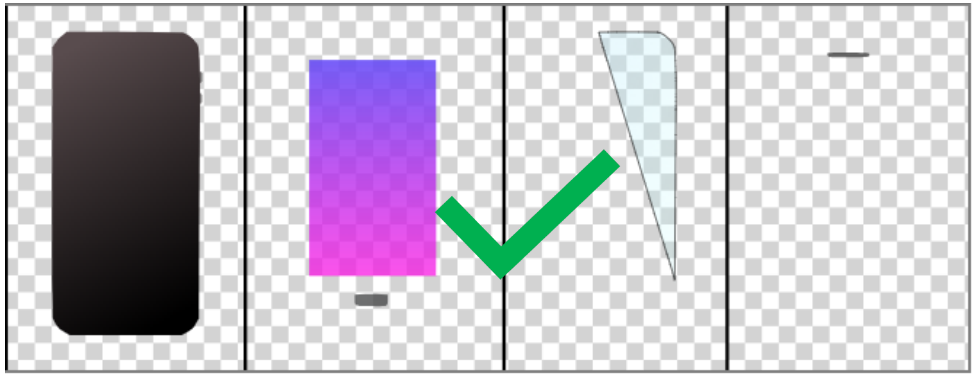


Estimating layer parameters

- Energy

$$E = w_r E_{\text{recon}} + w_g E_{\text{gamut}} + w_c E_{\text{compact}}$$

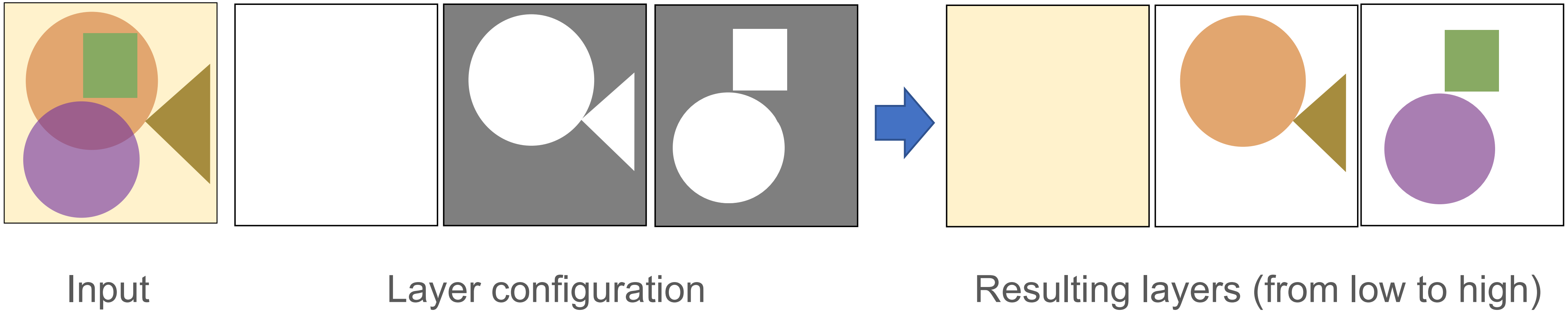
Compactness loss: $E_{\text{stasck}} = \sum_{1 \leq i < j \leq n,\, L_i \cap L_j \neq \emptyset} 1(|L_i| - |L_j|)^2$



Estimating layer parameters

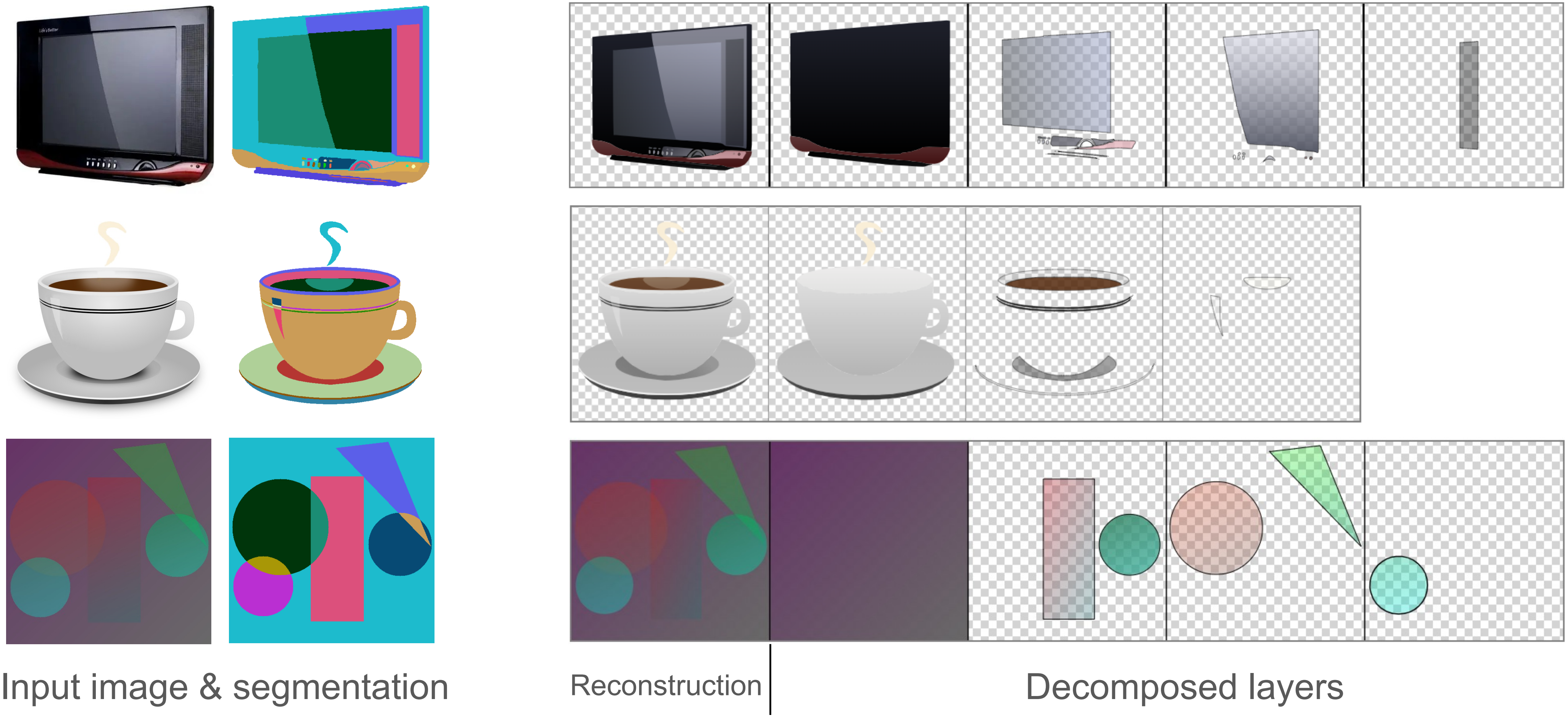
- Energy

$$E = w_r E_{\text{recon}} + w_g E_{\text{gamut}} + w_c E_{\text{compact}}$$

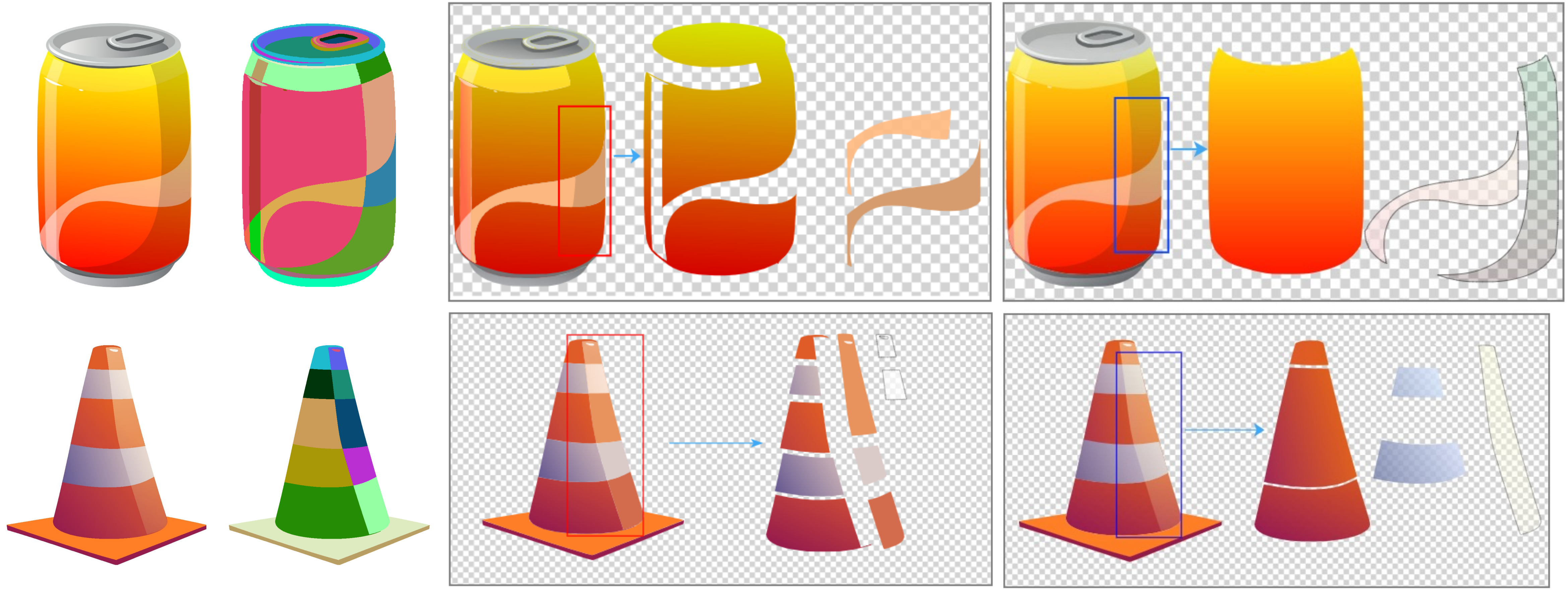


We set $w_r = 20, w_g = 10, w_c = 0.02$ in all our examples.

Results


Input image & segmentation
Reconstruction
Decomposed layers

Comparison



Input image & segmentation

[Favreau et al. 2016]

Ours

Conclusion

- We presented a fully automatic method for multi-layer vectorization from a segmented raster image.
- Perceptually-motivated rules reduce the search space, allowing us to find the globally optimal layer decomposition.
- The decomposed layers better reflect the shape and hierarchical structure of the input than previous methods.

THANK YOU!

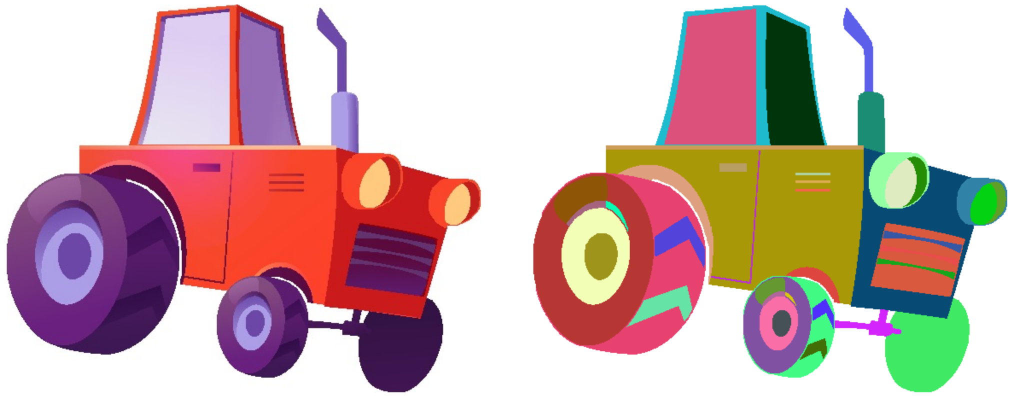
- **Contact Information:**

  - Zheng-Jun Du: dzj@qhu.edu.cn
  - Liang-fu Kang: kanglf20@mails.tsinghua.edu.cn
  - Jianchao Tan: tanjianchaoustc@gmail.com
  - Yotam Gingold: ygingold@gmu.edu
  - Kun Xu: xukun@tsinghua.edu.cn

Supplementary

Effectiveness of the rules

- The effectiveness of these rules in reducing search space.



#region: 45,   #edge: 171

in the initial region adjacency graph

Condition	#tree
w/o the surrounding rule	672
w/o the size rule	1,152
w/o the adj. strength rule	10,814
with all rules	112