A foreground-background separation algorithm for image compression

Simard, P.Y. Malvar, H.S. Rinker, J. Renshaw, E.

Data Compression Conference, 2004. Proceedings. DCC 2004
pages 498- 507

Presented by Chen Lin

Contents

- Why separation?
- SLIm architecture
- Encode Foreground/Background
- Mask computation
- Comparisons with other separating algorithms

Why separation?

- Many bitmaps contain a mixture of text and images
- Text (Sharp edges) → compressed poorly by image codec
- Image → cannot be compressed by Binary codec

Segmented Layered Image (SLIm)

Segments the image into:

- Background
- Foreground
- Binary Mask (contains text)

Better compression: each codec only sees the kind of data for which it was designed.


LAMPS PLUS
SAVE $100
$199.95
1/2 PRICE DAYS & Sale
SAVE 10-50%
On selected sale items throughout the store!
1/3 OFF
$99.95
1/3 OFF
$49.95
1/3 OFF
$59.99
SAVE 40%
$14.95


LAMPS PLUS
SAVE $100
$199.95
1/2 PRICE DAYS & Sale
SAVE 10-50%
On selected sale items throughout the store!
1/3 OFF
$99.95
1/3 OFF
$49.95
1/3 OFF
$59.99
SAVE 40%
$14.95


1/2 PRICE DAYS & Sale

SLIm Architecture

- Mask Computation
- Foreground/background layering
- Image codec
- Mask codec

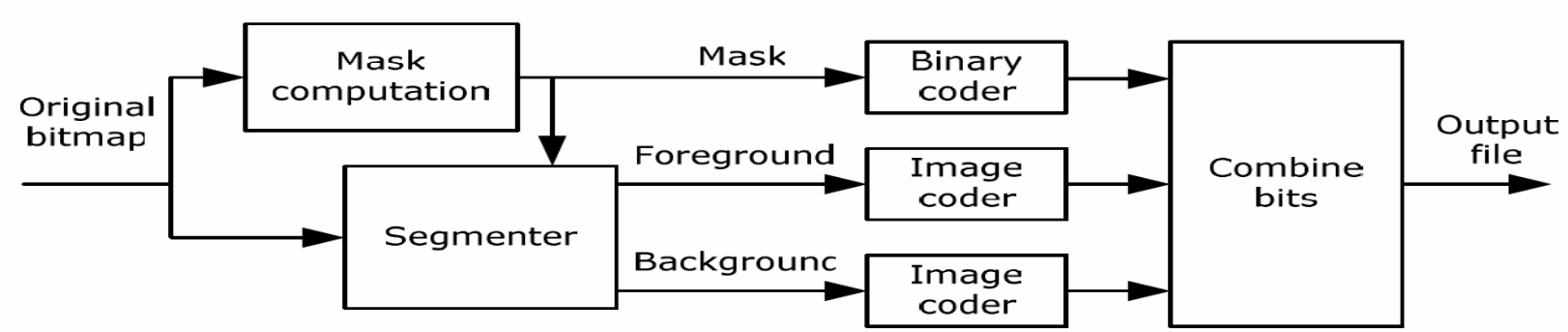


Figure 1. Simplified block diagram of the SLIm encoder.

SLIm Architecture

- Mask Computation
- Foreground/background layering
- Image codec
- Mask codec

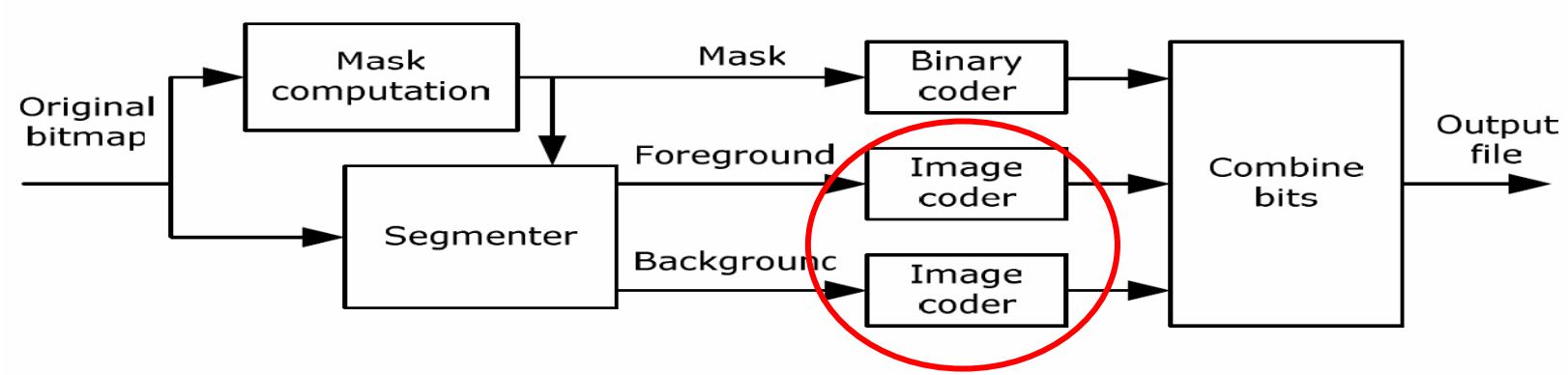


Figure 1. Simplified block diagram of the SLIm encoder.

SLIm Architecture

- Mask Computation
- Foreground/background layering
- Image codec
- Mask codec

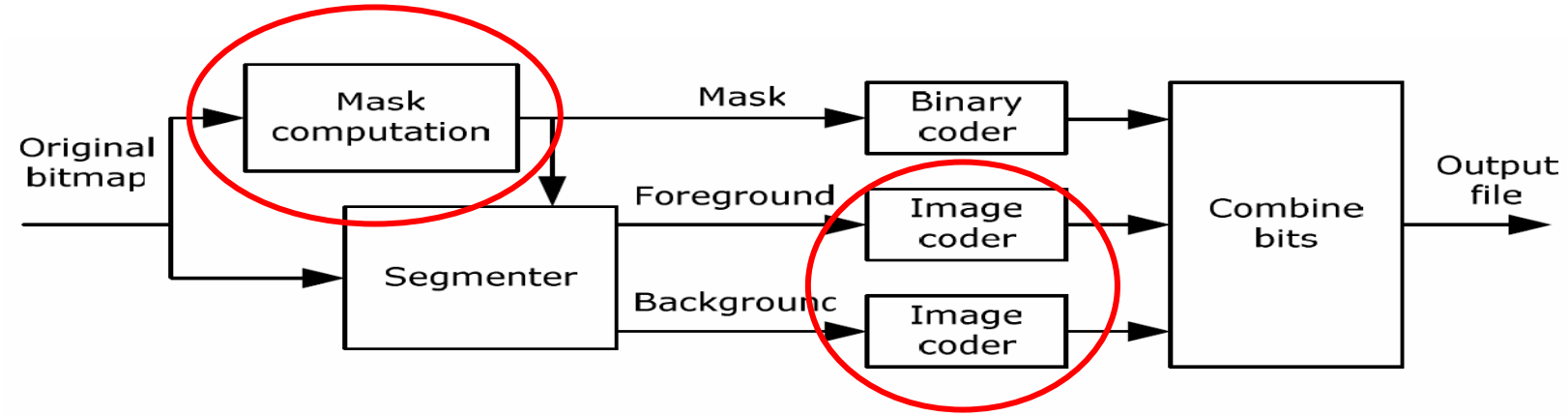
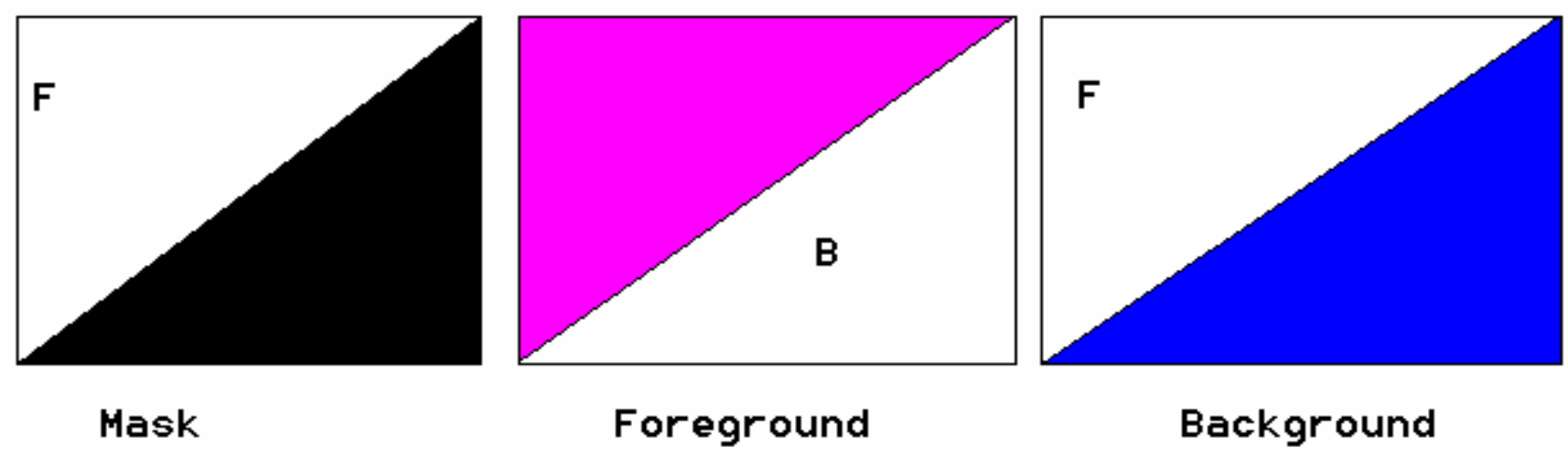


Figure 1. Simplified block diagram of the SLIm encoder.

Encode Fore/Background



- Masked wavelet coder
- Encode the whole image

Encode Fore/Background

How to fill the masked pixels with values that don’t introduce sharp edges?

- Voronoi: set the value of masked pixel to the value of closest visible pixel
- Filtering: running an averaging filter that scans the image from left to right and top to bottom and replace each masked pixels by a linear combination of the left and above pixel

Mask Computation -- assumptions

- The bitmap has N pixels → 2^N possible masks
- Compute the mask using only a gray level version of image. (text has a high contrast in the Y component of YUV)
- Foreground and background are constant over small regions. Thus we look for a mask that minimizes the variance.

Mask Computation -- variance

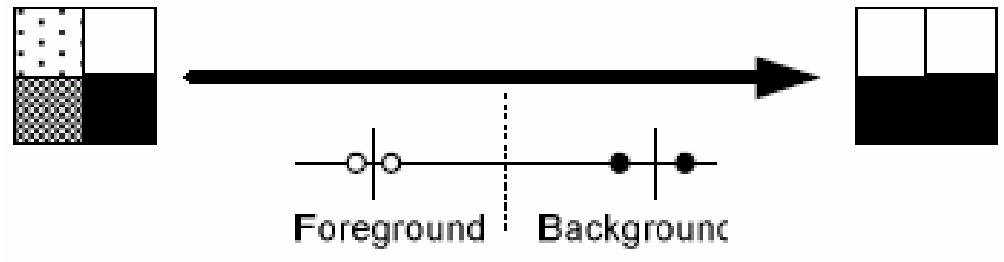
$$v_F = \sum_{x \in F} (f(x) - \mu_F)^2, \quad v_B = \sum_{x \in B} (f(x) - \mu_B)^2$$

$$\mu_F = \frac{1}{N_F} \sum_{x \in F} f(x) \quad \text{and} \quad \mu_B = \frac{1}{N_B} \sum_{x \in B} f(x)$$

$$E = v_F + v_B$$

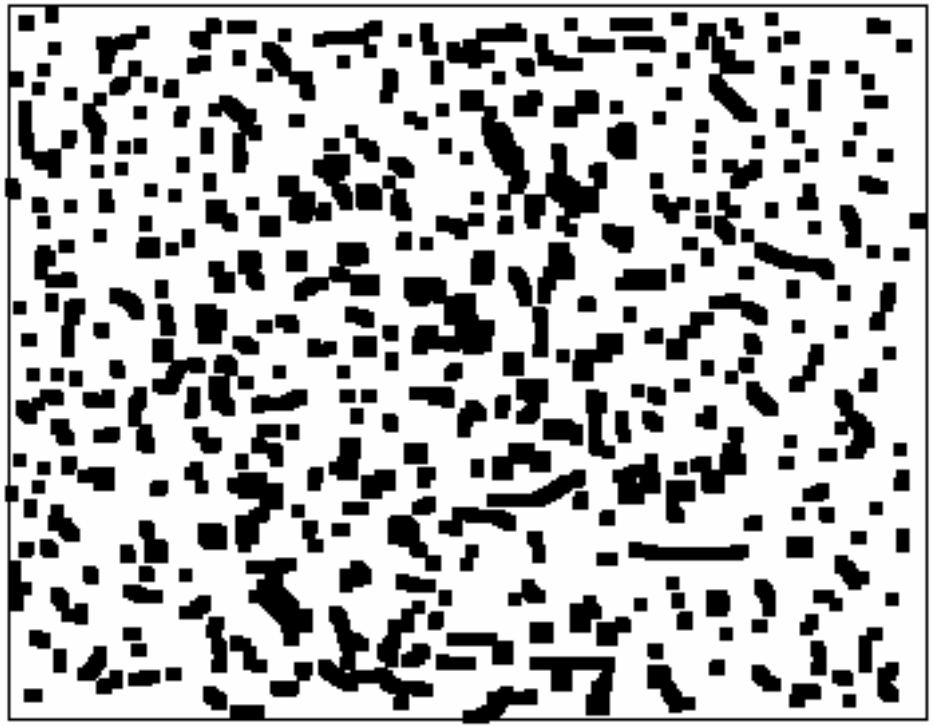
Mask Computation – Step 1

- Partition the image into 2×2 pixel sub-images.
- There are only $2^4 = 16$ possible masks.
- Find the optimal F and B that minimize energy: $E = v_F + v_B$



Result of Step 1

- Salt and Pepper



Mask Computation – Step 2

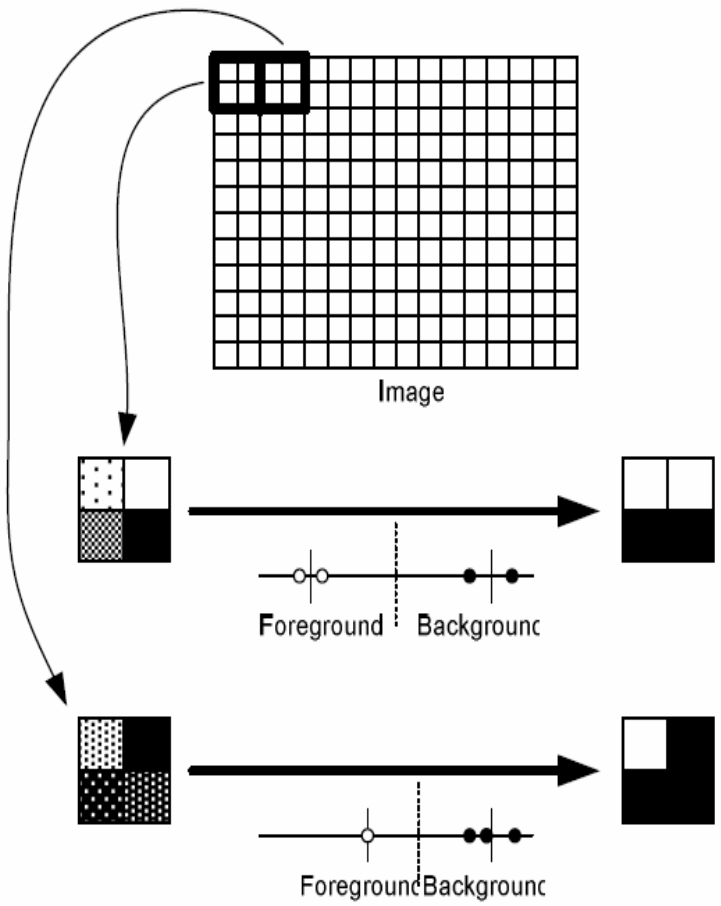
- Combing adjacent regions:

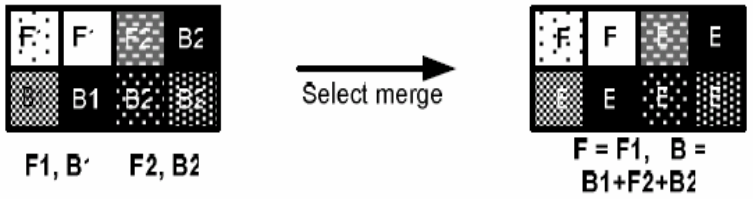
1) Label 2 adjacent regions 1 and 2, and their corresponding foreground and background, F1, B1 and F2, B2

2) 7 possible combinations (next Fig.)

3) Winning combination is the one has the lowest energy: $E = v_F + v_B$

- Merge till there is only one region

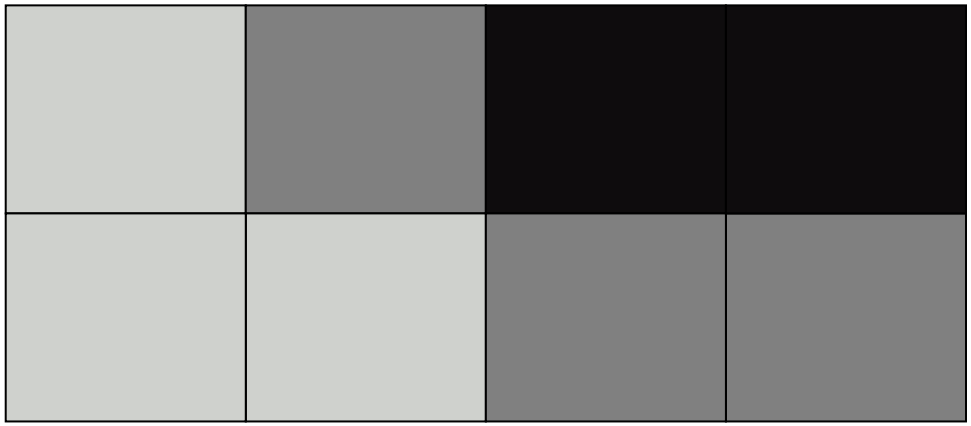




Possible merges:

New F	New B
F1	B1,F2,B2
F1,B1,F2	B2
F1,F2,B2	B´
F2	F1,B1,B2
F1,F2	B1,B2
F1,B1	F1,F2
F1,B2	B1,F2

Result of Step 2

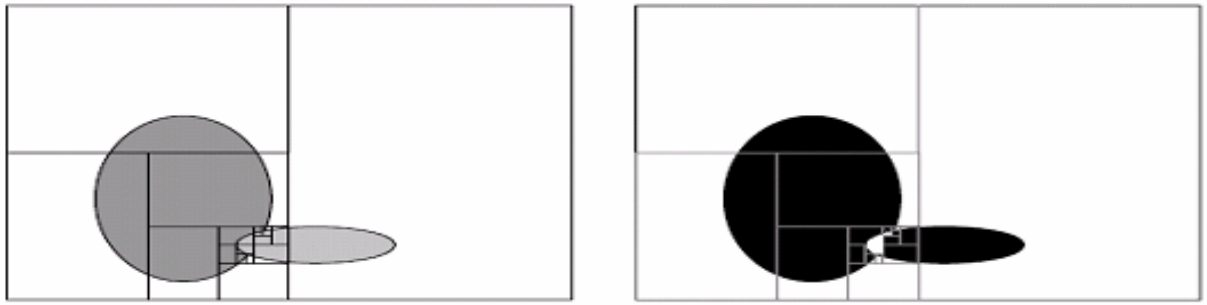


A block has several gray levels → more than two colors are grouped in a region

Mask Computation – Step 3

In order not to merge two colors in a region:

- If two colors are merged in one region, → energy E exceeds a certain threshold T.
- So we do a top-down recursive sweep, split the region if the energy E exceeds T.
- Compare each pixel value with $(\mu_F + \mu_B)/2$



Comparisons with other method

JPEG (200KB)

TIFF-FX (135KB)

SLlm (100KB)

DjVu (65KB)

Conclusion

- Good Compression
- Apply off-the-shelf image codec
- Speed: less than 10 operations per pixel