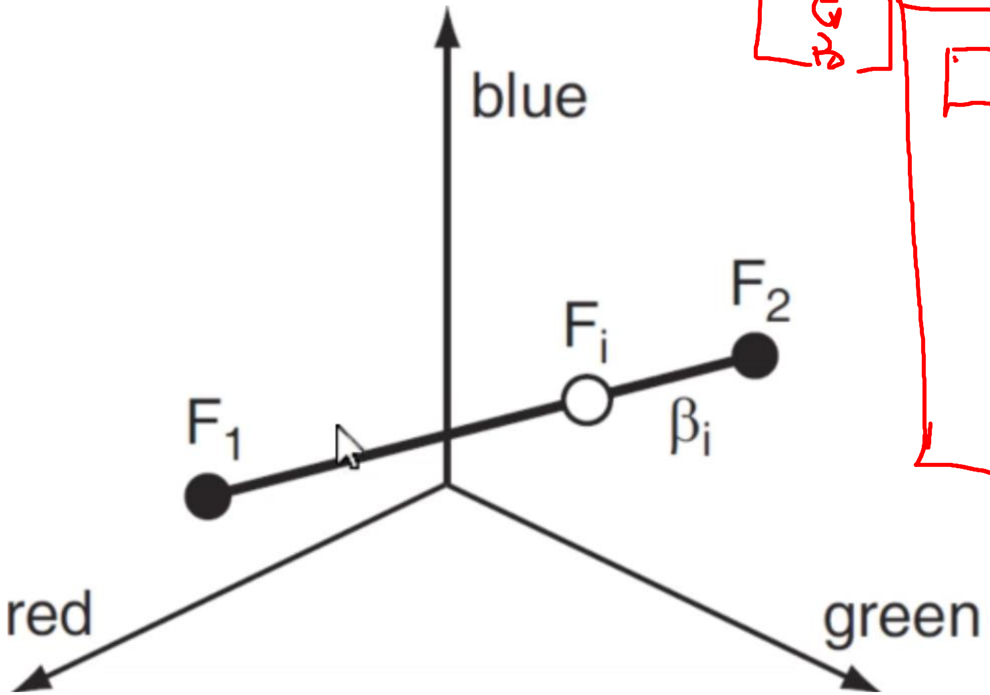
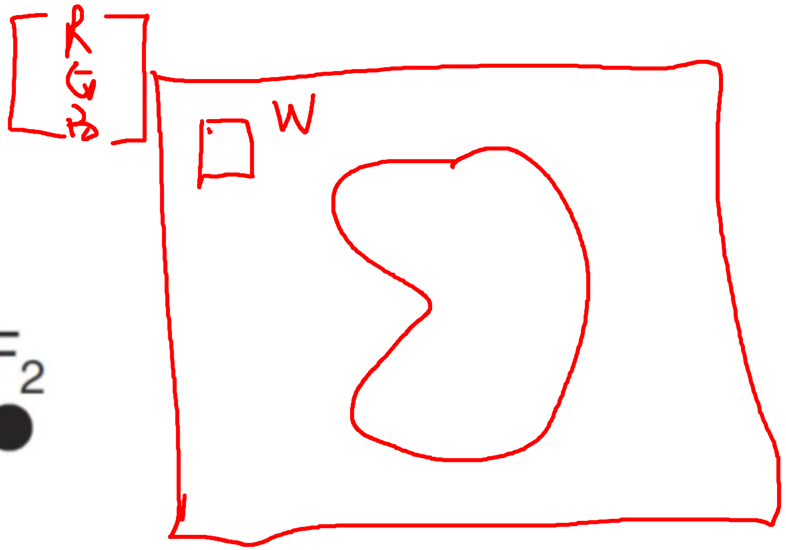
Lecture 24 Closed-form matting

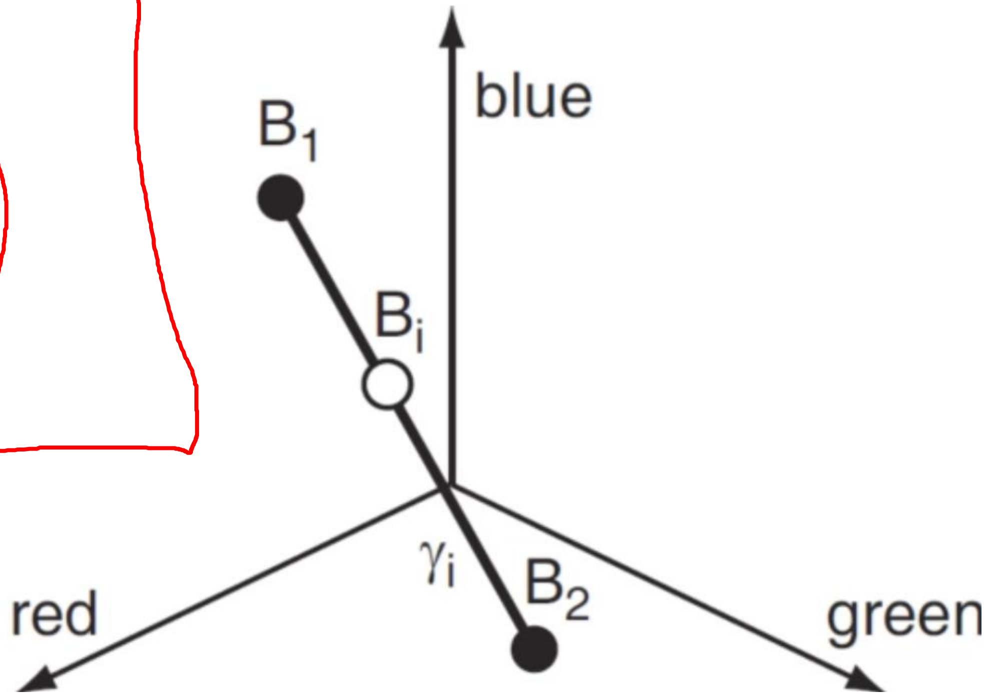
Yuyao Zhang PhD
zhangyy8@shanghaitech.edu.cn
SIST Building 2 302-F

Closed-form matting

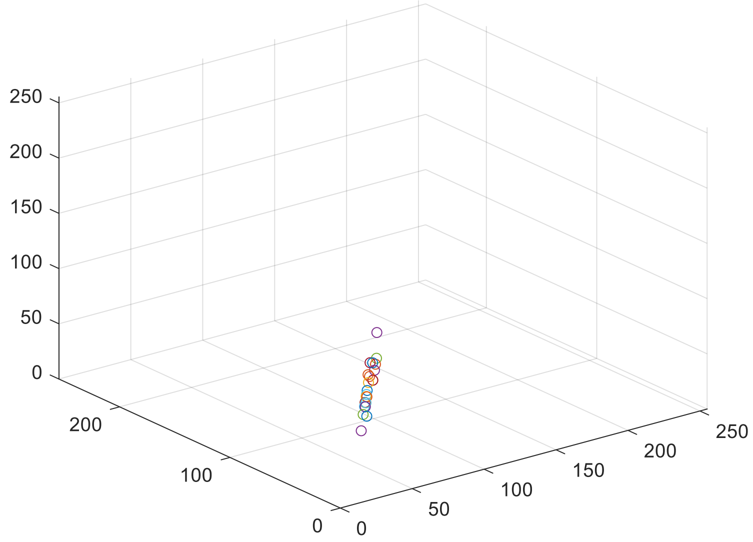
$$I = \alpha \cdot F + [1 - \alpha] \cdot B$$

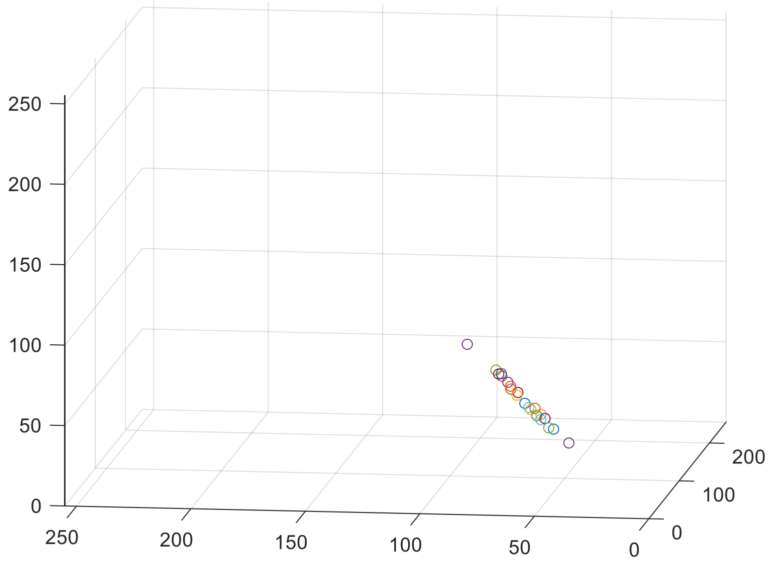




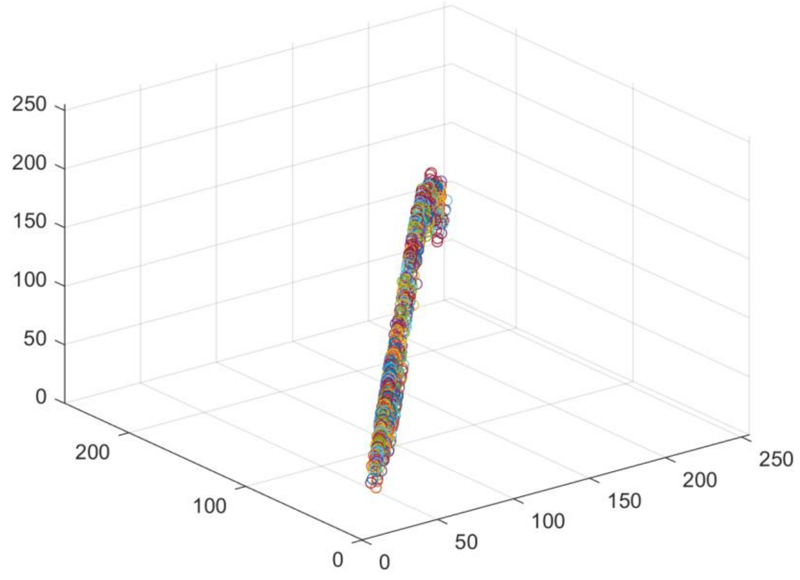


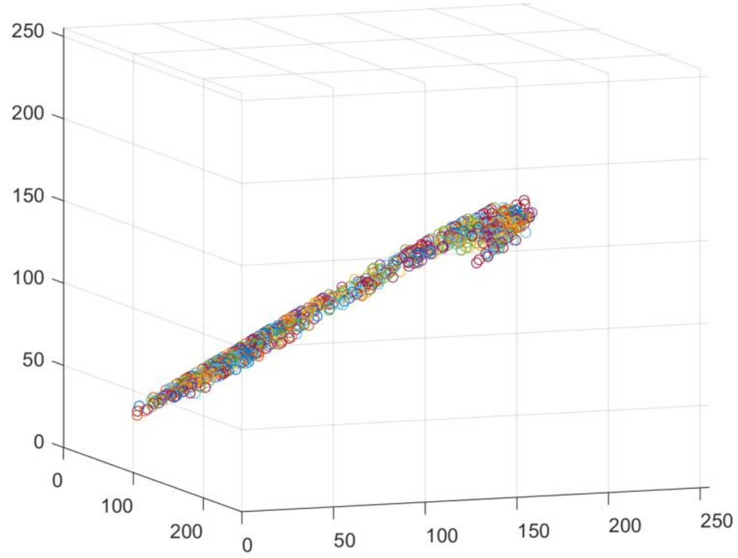
Color line assumption





上海科技大学
ShanghaiTech University

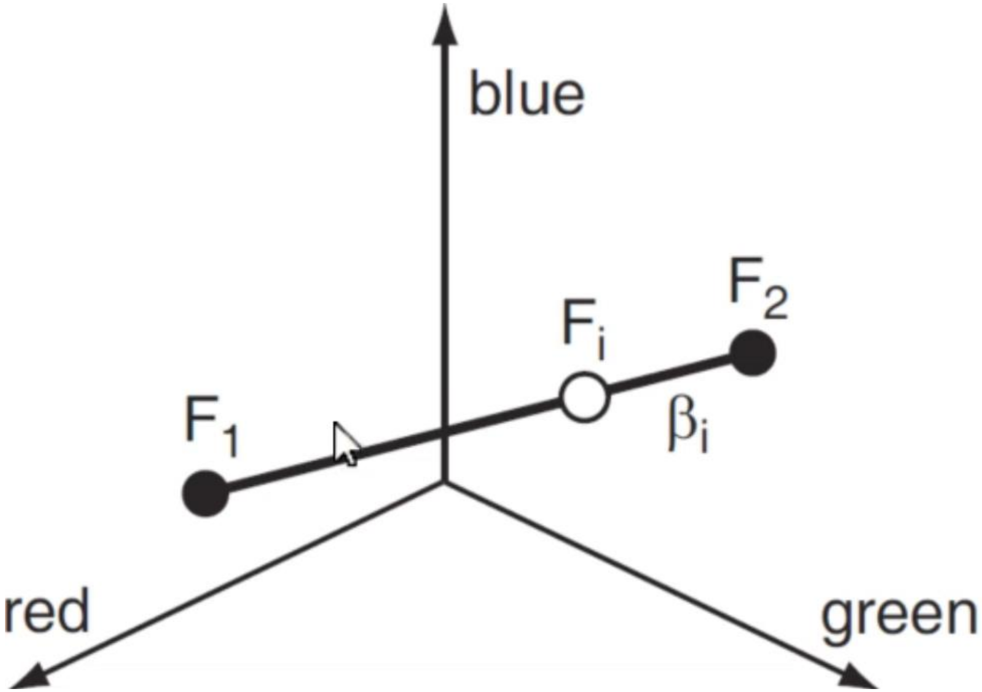

250
200
150
100
50
0
200
100
0
0
50
100
150
200
250


250
200
150
100
50
0
0
100
200
0
50
100
150
200
250

Closed-form matting

- Color line assumption:
- FG and BG colors in a small window lie on a straight line in RGB space.
- Line depends on which window we chose.

Closed-form matting



$$F_i = \beta_i F_1 + (1-\beta_i)\ F_2$$

$$B_i = \gamma_i B_1 + (1-\gamma_i)\ B_2$$

If color line assumption holds, then the true matte (α) satisfies

$$\alpha_i = a^T I_i + \mathrm{b}$$

for all pixels in the window.

Prove for the affine transformation

$$\underset{1\times1}{\alpha_i} = \underset{3\times1}{a}^T \underset{3\times1\ [R,G,B]^T}{I_i} + \underset{1\times1 \leftarrow \text{scale}}{b}$$

We combine the matting equation and the color line assumption and get:

$$I_i = \alpha_i F_i + (1-\alpha_i) B_i$$

$$= \alpha_i[\beta_i F_1 + (1-\beta_i)F_2] + (1-\alpha_i)[\gamma_i B_1 + (1-\gamma_i) B_2]$$

$$= \alpha_i\beta_i F_1 + \alpha_i F_2 - \alpha_i\beta_i F_2 + \gamma_i B_1 - \alpha_i\gamma_i B_1 + B_2 - \alpha_i B_2 - \gamma_i B_2 + \alpha_i\gamma_i B_2$$

$$I_i - B_2 = [F_2-B_2, F_1-F_2, B_1-B_2]\begin{bmatrix}\alpha_i \\ \alpha_i\beta_i \\ \gamma_i - \alpha_i\gamma_i\end{bmatrix}$$

$$\Rightarrow \underset{3\times1}{\begin{bmatrix}\alpha_i \\ \alpha_i\beta_i \\ \gamma_i - \alpha_i\gamma_i\end{bmatrix}} = \underset{3\times3}{[\overset{3\times1}{F_2-B_2}, F_1-F_2, B_1-B_2]^{-1}} \underset{3\times1}{[I_i - B_2]}$$

$$\begin{bmatrix} \overset{1\times3}{r_1^T} \\ \\ \\ \end{bmatrix}^{-1} [I_i - B_2]$$

$$\Rightarrow \alpha_i = \underset{3\times1}{r_1}^T (I_i - B_2)$$

$$= r_1^T I_i - \underset{1\times1}{r_1^T B_2}$$

$$\Downarrow$$

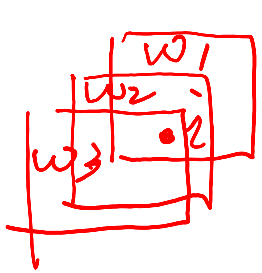
$$\alpha_i = a^T I_i + b$$

Cost function

$$J(\alpha_i, a_i, b_i) = \sum_{j=1}^{N} \sum_{i \in window\, j} (\alpha_i - (a_j^T I_i + b_j))^2$$

- i is the index of every pixel
- j is the index of every window
- For every pixel, we need to determine α_i
- For every window, we need to determine a_j & b_j

Cost function



$a_j \quad b_j$

There exists a tight constrain between what happen on each pixel.

pixel

$$\arg\min J(\alpha_i, a_i, b_i) = \sum_{j=1}^{N} \sum_{i \in window\, j} (\alpha_i - (a_j^T I_i + b_j))^2$$

window

$$\sum_{j=1}^{N} \left\| \begin{bmatrix} I_1^{jT} & 1 \\ I_2^{jT} & 1 \\ \vdots & \vdots \\ I_w^{jT} & 1 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \begin{bmatrix} \alpha_1^j \\ \alpha_2^j \\ \vdots \\ \alpha_w^j \end{bmatrix} \right\|^2$$

Image color

$$= \sum_{j=1}^{N} \left\| G_j \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \bar{\alpha}_j \right\|^2$$

unknown

上海科技大学 ShanghaiTech University

A relaxation for optimization

my color

unknow

$$\operatorname{argmin} \sum_{j=1}^{N} \left\| G_j \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \alpha_j \right\|^2$$

pixel window

Minimizing each of these equations is a linear least square problem.

Suppose we know the $\alpha's$ and $a\&b$ that make $\partial \left\| G_j \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \alpha_j \right\|^2 = 0$ as small as possible:

∂ ~~a~~ min

$$\Rightarrow \quad \begin{bmatrix} a_j \\ b_j \end{bmatrix}^* = (G_j^T G_j)^{-1} G_j^T \alpha_j$$

Then we have $a\&b$ are functions of α!

上海科技大学 ShanghaiTech University

Matting Laplacian

- So the whole optimization is a function of only α.

$$\begin{bmatrix} a_j \\ b_j \end{bmatrix}^* = (G_j^T G_j)^{-1} G_j^T \alpha_j \quad ①$$

$$② \quad \arg\min J(\alpha_i, a_i, b_i) = \sum_{j=1}^{N} \sum_{i \in window\, j} (\alpha_i - (a_j^T I_i + b_j))^2$$

① ② ⇒

$$\arg\min J(\alpha_j) = \sum_{j=1}^{N} \| \underbrace{G_j (G_j^T G_j) G_j^T}_{\text{Image color}} \alpha_j - \underbrace{\alpha_j}_{\text{matte}} \|^2$$

$$= [\alpha_1 \quad \cdots \quad \alpha_N] L \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}$$

$$= \alpha^T L \alpha$$

→ Matting Laplacian.

N = # of image pixel

W = # of pixels in a window

Solution

L_{ij} $N \times N$

$i \in [1, N]$

$j \in [1, N]$

$|i-j| < |W|-1$

$$\arg\min J(\alpha) = \underbrace{\alpha^T}_{1\times N} L \underbrace{\alpha}_{N\times 1}$$

$$\frac{d}{d\alpha} = 2L\alpha = 0$$

$$\boxed{L\alpha = 0}$$

- Null vectors of L solves matting equation.
- Bad news: many 0-eigenvectors
- To constrain the matte, we need user input (scribbles).
- i.e. some pixels are forced to have $\alpha = 0$ for BG and $\alpha = 0$ for FG.

Solution

- So we actually solve:

$$\arg\min \alpha^T L\alpha + \lambda\Big(\sum_{i\epsilon FG}(1-\alpha_i) + \sum_{i\epsilon BG}\alpha_i\Big)^2$$

- We seek $\alpha's$ eigenvectors of L with eigenvalue 0 (null vectors).
- There are many such eigenvectors. And the matte we look for is a linear combination of these eigenvectors since:

if $Lv = 0$, and $w = \beta_1 v_1 + \beta_2 v_2 + \cdots + \beta_k v_k$ $= 0$

Then $Lw = 0$

$LV = 0$

Solution

- Idea: we have 100 grayscale eigenvectors, we can combine them to build the binary matte.
- This is called “spectral matting”.
- Cost function:

L 0 $\alpha \rightarrow$

$$\min J(\alpha) = \min \sum_{i,k} |1 - \alpha_i^k|^{\gamma} + |\alpha_i^k|^{\gamma}$$

s.t. $$\alpha^k = E\beta^k$$

matting

* null-vectors of L

$L \leftarrow$ null vectors

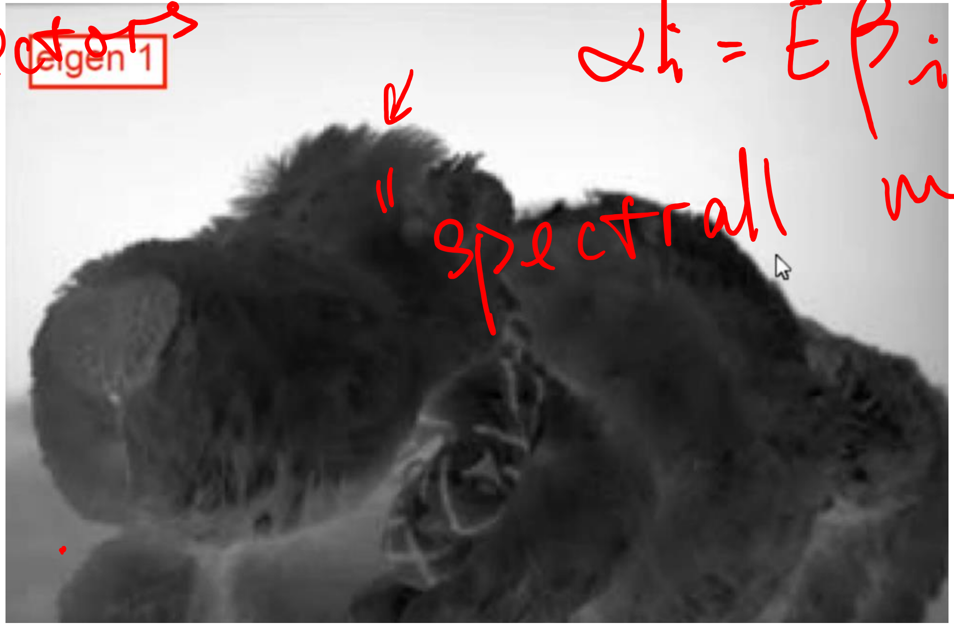
$\alpha_i = E\beta_i$

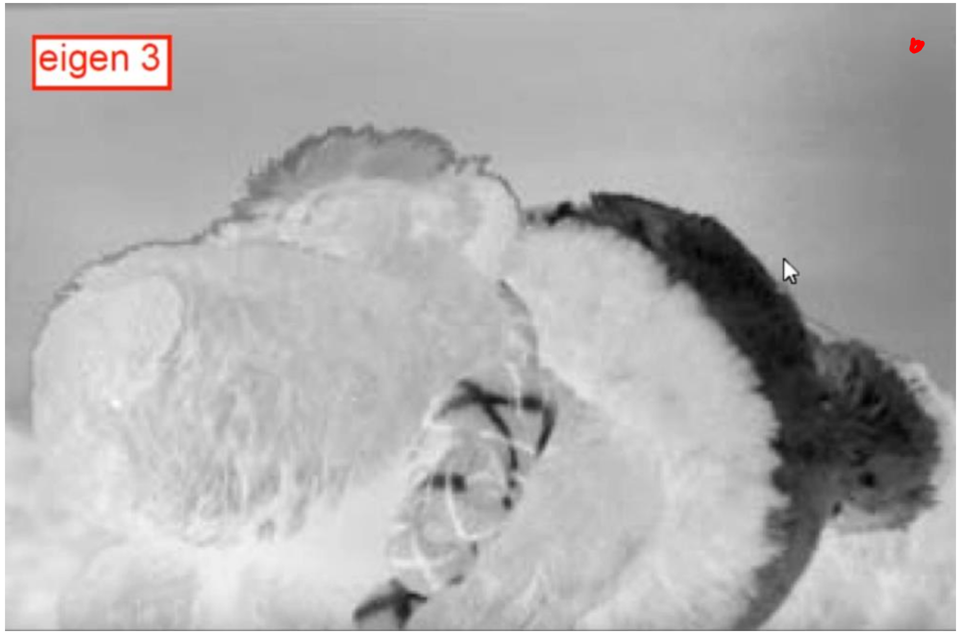
"spectral matting"

$N \times N$

α

$N \times 1$





Other application

$$I = Jt + (1 - t)A$$

L

α

Take home message

- Bayesian image matting
- Closed-form image matting
- http://people.csail.mit.edu/alevin/papers/Matting-Levin-Lischinski-Weiss-PAMI-o.pdf
- https://arxiv.org/pdf/2004.00626v2.pdf
- https://grail.cs.washington.edu/projects/background-matting/