Karlsruhe Institute of Technology

Visual Computing: Matting

SS 2016

IVD - Institut für Visualisierung und Datenanalyse

Jun. Prof. Dr.-Ing. Boris Neubert
Karlsruhe Institut für Technologie

Bayesian Matting

- We need an accurate trimap for Bayesian Matting
- Solution to problem R. Radke Sec. 2 & R. Szeliski Sec. 10.2.

Bayesian Matting

- We need an accurate trimap for Bayesian Matting
- Solution to problem R. Radke Sec. 2 & R. Szeliski Sec. 10.2.

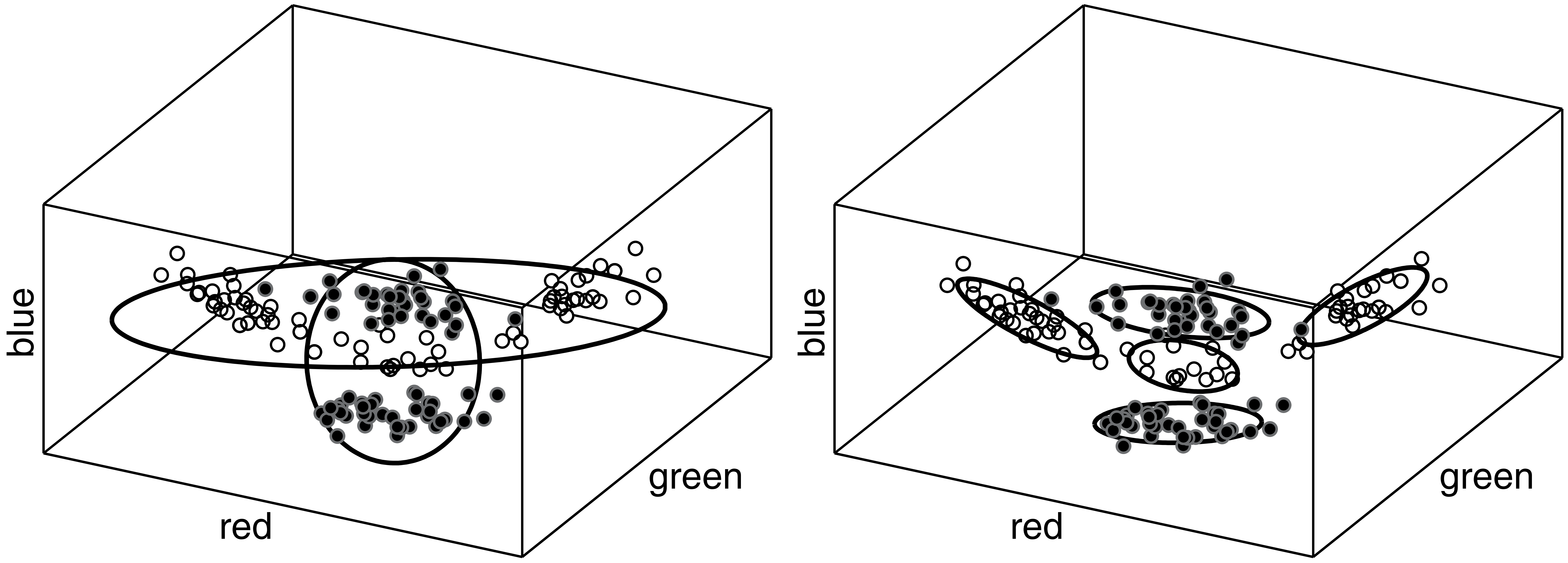
$$\arg\max_{\alpha,F,B} \; P(I|F,B,\alpha)P(F)P(B)P(\alpha)$$

Refinement & Extensions for BM

- suggestions?

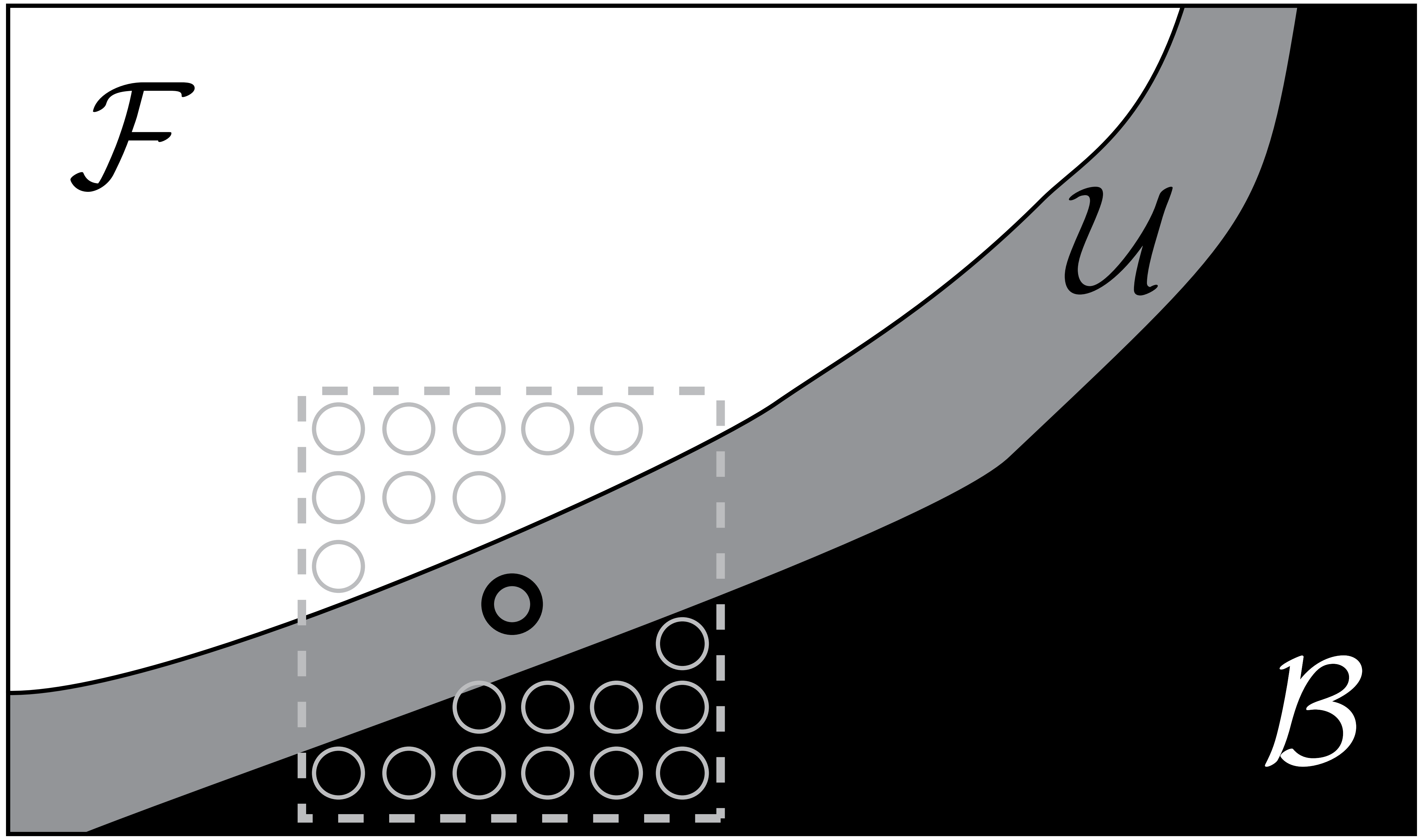
Refinement & Extensions

1) better FG and BG model



Refinement & Extensions

2) local estimation



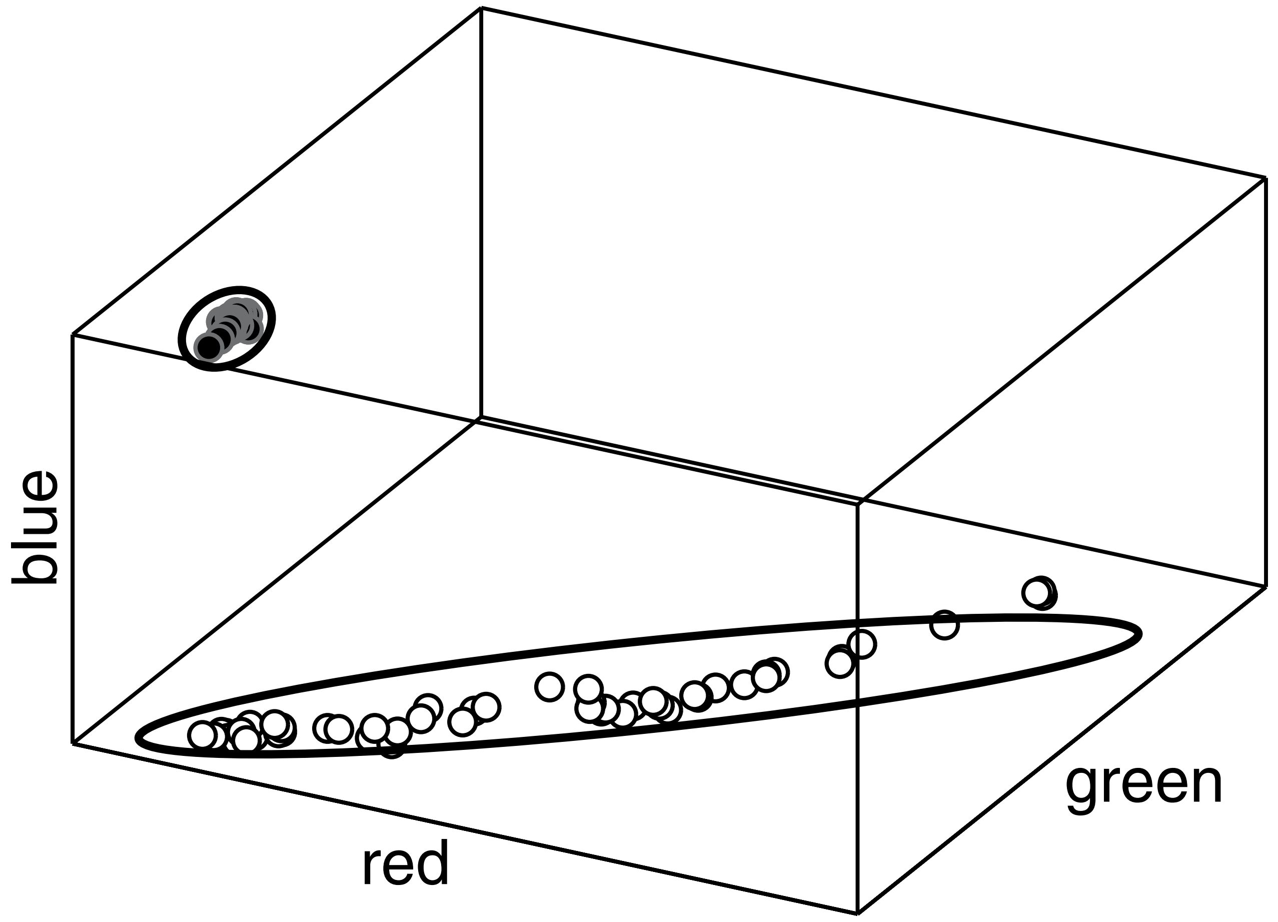
3) so far: constant distribution for alpha values

Closed-form Matting

$$I = \alpha F + (1 - \alpha) B$$

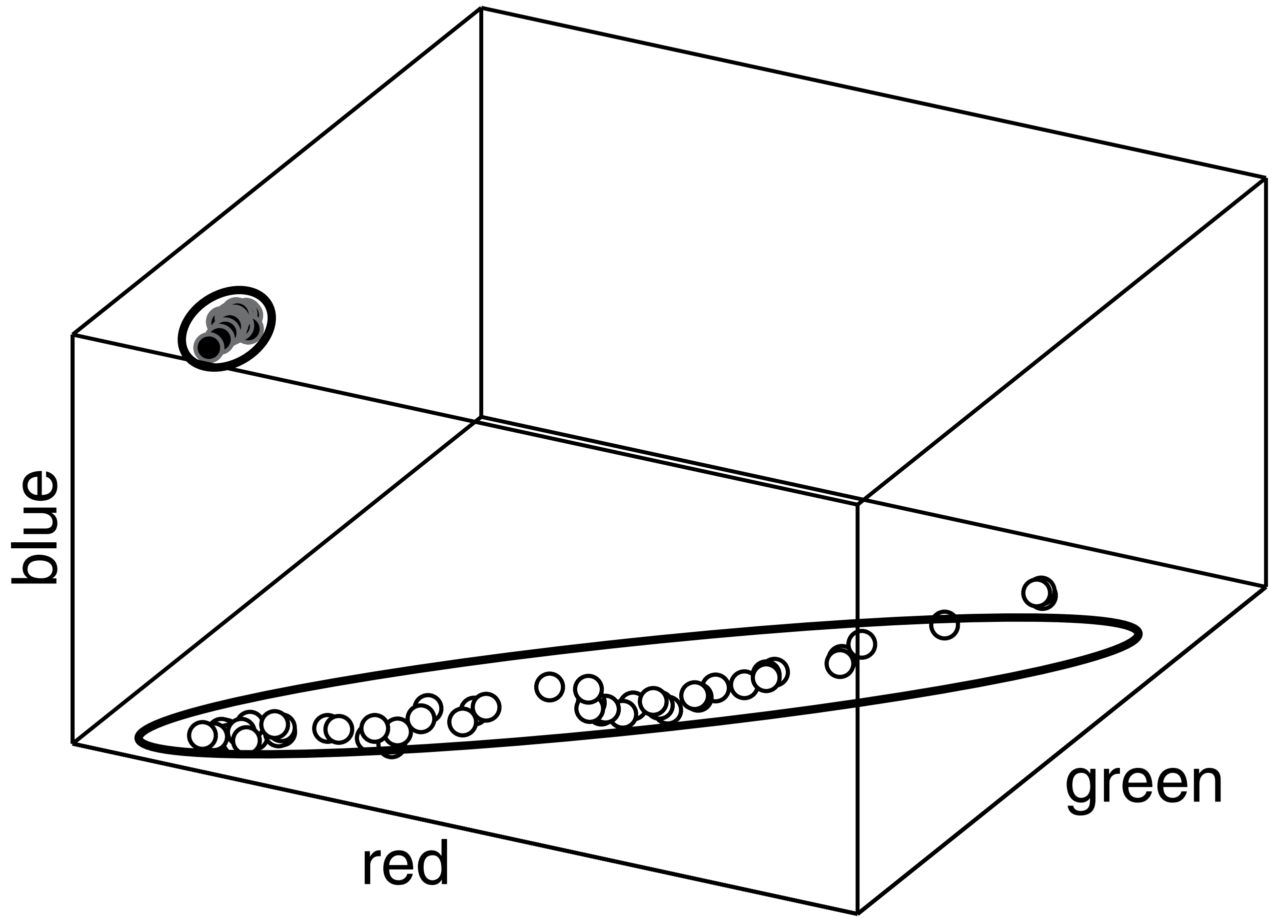
Closed-form Matting

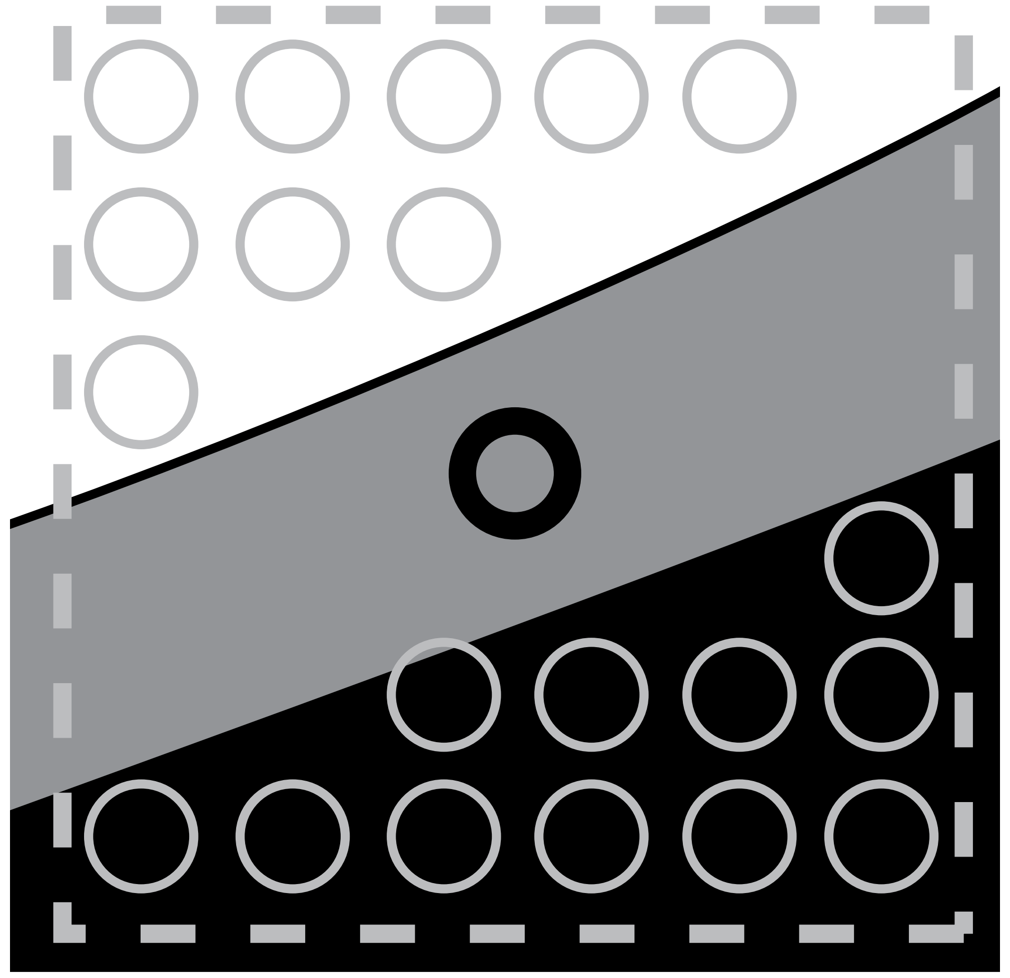
$$I = \alpha F + (1 - \alpha) B$$



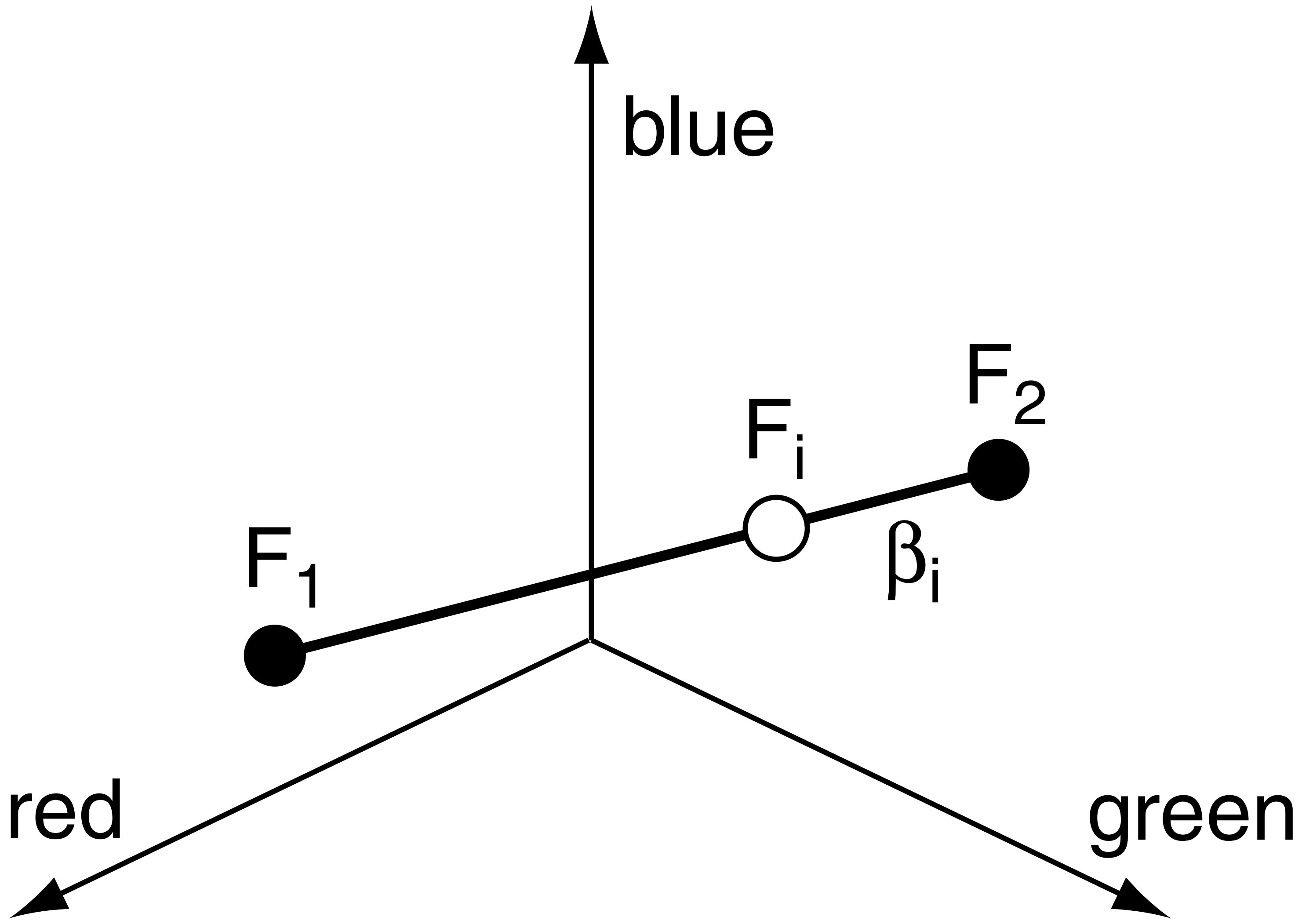
Closed-form Matting

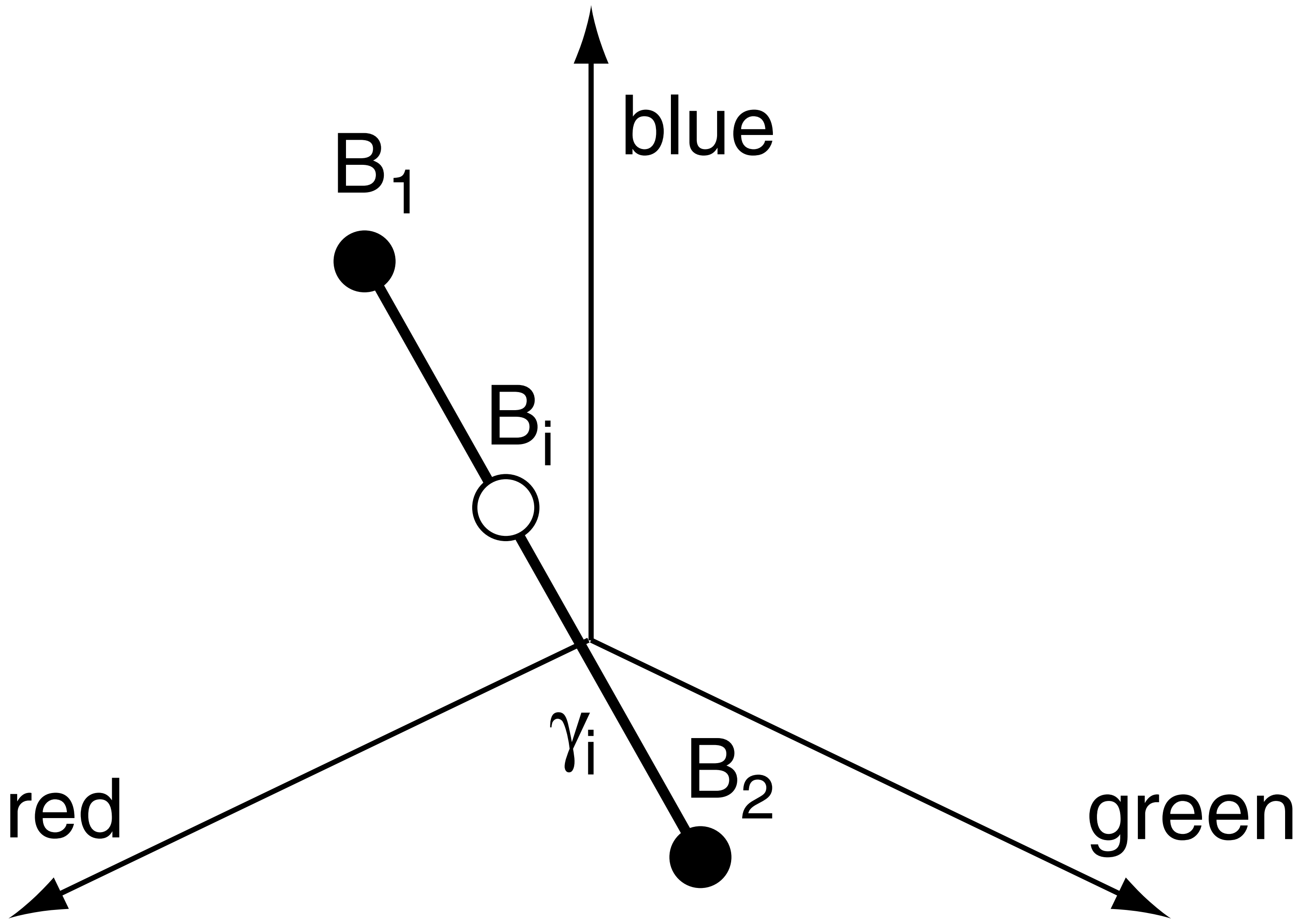
$$I = \alpha F + (1 - \alpha) B$$



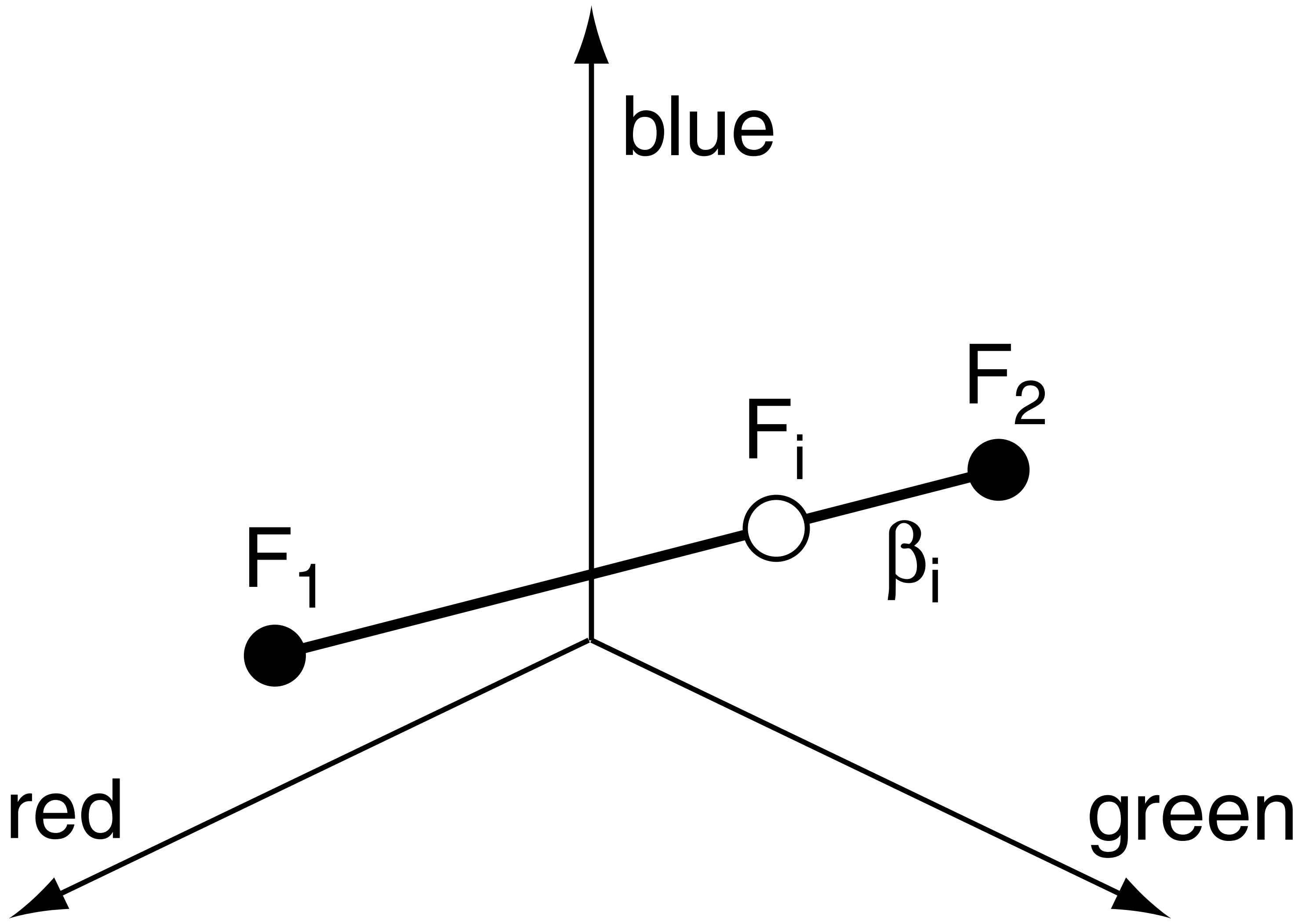


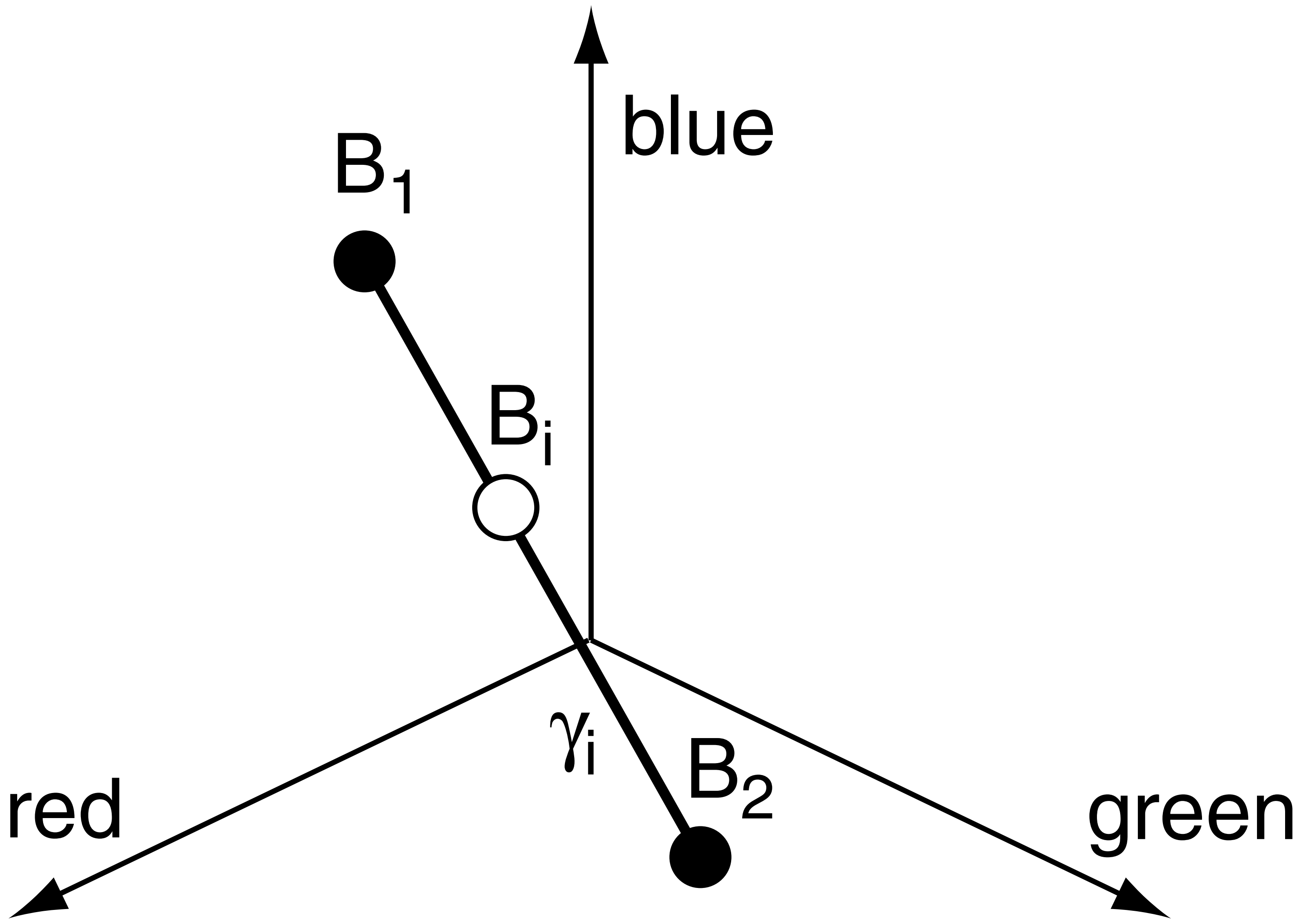
Color Line Assumption





Color Line Assumption





Color Line Assumption:
FG and BG colors for small neighborhoods lie on a line in RGB space.

Consequences for Matte?

$$F_i = \beta_i F_1 + (1 - \beta_i) F_2$$
$$B_i = \gamma_i B_1 + (1 - \gamma_i) B_2$$

Consequences for Matte?

$$F_i = \beta_i F_1 + (1 - \beta_i) F_2$$
$$B_i = \gamma_i B_1 + (1 - \gamma_i) B_2$$

If CLA holds:

Consequences for Matte?

$$F_i = \beta_i F_1 + (1 - \beta_i) F_2$$
$$B_i = \gamma_i B_1 + (1 - \gamma_i) B_2$$

If CLA holds:

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

Consequences for Matte?

$$F_i = \beta_i F_1 + (1 - \beta_i) F_2$$
$$B_i = \gamma_i B_1 + (1 - \gamma_i) B_2$$

If CLA holds:

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

for all pixels in window.

Proof

Proof

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

Proof

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

$$F_i = \beta_i F_1 + (1 - \beta_i) F_2$$
$$B_i = \gamma_i B_1 + (1 - \gamma_i) B_2$$

Visual Break!

http://www.dezeen.com/2015/11/02/in-the-eyes-of-the-animal-virtual-reality-installation-marshmallow-laser-feast-abandon-normal-devices-festival-england/

VFX Examples

http://www.premiumbeat.com/blog/vfx-didnt-realize-happening/

Compute Matte

- Setup a cost function J
- find alpha for each pixel and a,b for each window, s.t. the CLA is satisfied.

Cost Function J

Cost Function J

$$J(\alpha, a, b) = \sum_{j=1}^{N} \sum_{i \in w_j} (\alpha_i - (a_j^T I_i + b_j))^2$$

Cost Function J

$$J(\alpha, a, b) = \sum_{j=1}^{N} \sum_{i \in w_j} (\alpha_i - (a_j^T I_i + b_j))^2$$

Q: number of unknowns? knowns?
Q: Why better then BM?

Reformulation of J

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| \begin{bmatrix} I_1^T & 1 \\ \vdots & \vdots \\ I_W^T & 1 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_W \end{bmatrix} \right\|_2^2$$

Add Regularization Term

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| \begin{bmatrix} I_1^T & 1 \\ \vdots & \vdots \\ I_W^T & 1 \\ \sqrt{\epsilon} I_{3\times 3} & 0 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_W \\ 0 \end{bmatrix} \right\|_2^2$$

Add Regularization Term

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| \begin{bmatrix} I_1^T & 1 \\ \vdots & \vdots \\ I_W^T & 1 \\ \sqrt{\epsilon} I_{3\times 3} & 0 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_W \\ 0 \end{bmatrix} \right\|_2^2$$

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| G_i \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \alpha \right\|_2^2$$

Add Regularization Term

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| \begin{bmatrix} I_1^T & 1 \\ \vdots & \vdots \\ I_W^T & 1 \\ \sqrt{\epsilon} I_{3\times3} & 0 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_W \\ 0 \end{bmatrix} \right\|_2^2$$

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| G_i \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \alpha \right\|_2^2$$

minimizing each of these is a LSP.

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| G_i \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \alpha \right\|_2^2$$

$$J(\alpha, a, b) = \sum_{j=1}^{N} \left\| G_i \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \alpha \right\|_2^2$$

$$\begin{bmatrix} a_j^* \\ b_j^* \end{bmatrix} = (G_j^\top G_j)^{-1} G_j^\top \bar{\alpha}_j$$

L - Matting Laplacian

$$J(\alpha) = \sum_{j=1}^{N} \|G_j(G_j^T G_j)^{-1} G_j^T \alpha_i - \alpha_i\|^2$$

L - Matting Laplacian

$$J(\alpha) = \sum_{j=1}^{N} \|G_j (G_j^T G_j)^{-1} G_j^T \alpha_i - \alpha_i\|^2$$

$$J(\alpha) = \alpha^T L \alpha$$

L - Matting Laplacian

$$J(\alpha) = \sum_{j=1}^{N} \|G_j(G_j^T G_j)^{-1} G_j^T \alpha_i - \alpha_i\|^2$$

$$J(\alpha) = \alpha^T L \alpha$$

minimize J as

L - Matting Laplacian

$$J(\alpha) = \sum_{j=1}^{N} \|G_j (G_j^T G_j)^{-1} G_j^T \alpha_i - \alpha_i\|^2$$

$$J(\alpha) = \alpha^T L \alpha$$

minimize J as

$$2L\alpha = 0$$

Karlsruhe Institute of Technology

L - Matting Laplacian

$$J(\alpha) = \sum_{j=1}^{N} \|G_j(G_j^T G_j)^{-1} G_j^T \alpha_i - \alpha_i\|^2$$

$$J(\alpha) = \alpha^T L \alpha$$

minimize J as

$$2L\alpha = 0$$

Null vector of L solves matting equation.

Adding Constrains

force some pixel to have

$$\alpha = 0 \quad (BG)$$

$$\alpha = 1 \quad (FG)$$

Adding Constrains

force some pixel to have

$$\alpha = 0 \quad (BG)$$
$$\alpha = 1 \quad (FG)$$

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}^\top L \boldsymbol{\alpha} \\ s.t. \quad & \alpha_i = 1 \quad i \in \mathcal{F} \\ & \alpha_i = 0 \quad i \in \mathcal{B} \end{aligned}$$

Adding Constrains

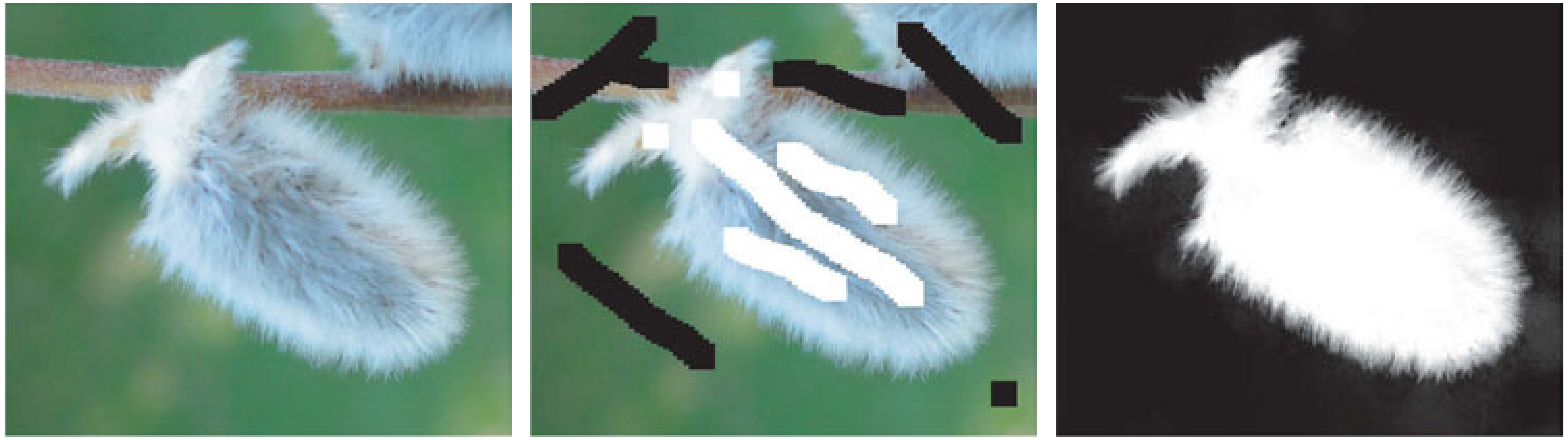
force some pixel to have

$$\alpha = 0 \quad (BG)$$
$$\alpha = 1 \quad (FG)$$

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}^\top L \boldsymbol{\alpha} \\ s.t. \quad & \alpha_i = 1 \quad i \in \mathcal{F} \\ & \alpha_i = 0 \quad i \in \mathcal{B} \end{aligned}$$

$$\min \boldsymbol{\alpha}^\top L \boldsymbol{\alpha} + \lambda(\boldsymbol{\alpha} - \boldsymbol{\alpha}_K)^\top D(\boldsymbol{\alpha} - \boldsymbol{\alpha}_K)$$

Results



References

A Closed Form Solution to Natural Image Matting

Anat Levin   Dani Lischinski   Yair Weiss
School of Computer Science and Engineering
The Hebrew University of Jerusalem
{*alevin,danix,yweiss*}*@cs.huji.ac.il*

Abstract

Interactive digital matting, the process of extracting a foreground object from an image based on limited user input, is an important task in image and video editing. From a computer vision perspective, this task is extremely chal-

color image, at each pixel there are 3 equations and 7 unknowns.

Obviously, this is a severely under-constrained problem, and user interaction is required to extract a good matte. Most recent methods expect the user to provide a *trimap*

- [CVFX] Computer Vision for Visual Effects, R.Radke, Image Matting Ch. 2. (**2.4. Closed Form Matting** 2.4.1 The Color Line Assumption, 2.4.3. Constraining the Matte)