Matting Introduction

Ivanov Ilya

xx.05.2021

Content

- Task Statement
- Applications
- Metrics
- Classical methods
- Modern methods

What is matting?

What is matting?

- Alpha matting refers to the problem of softly extracting the foreground from a given image

$$\mathbf{C}_i = \boldsymbol{\alpha}_i \mathbf{F}_i + (1 - \boldsymbol{\alpha}_i)\mathbf{B}_i$$

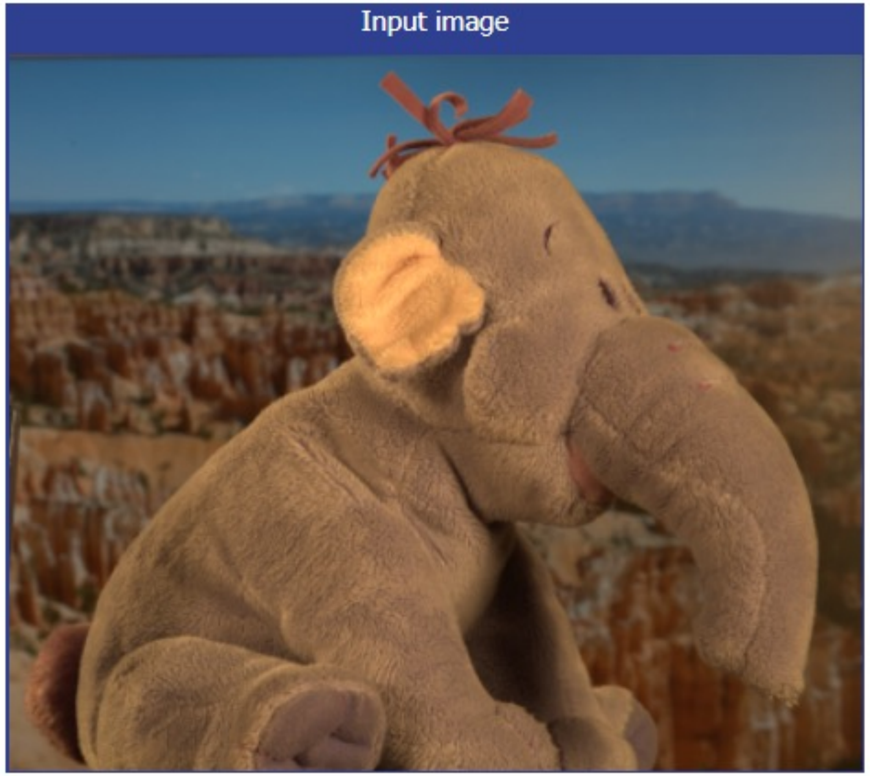
- C is the observed color
- F is the pixel color in the foreground
- B is the pixel color in the background
- α is the level of mixing between the two layers (α=0 means definite background, α=1 means definite foreground)
- 3 equations for RGB and 7 unknowns (matte, FG(3), BG(3))

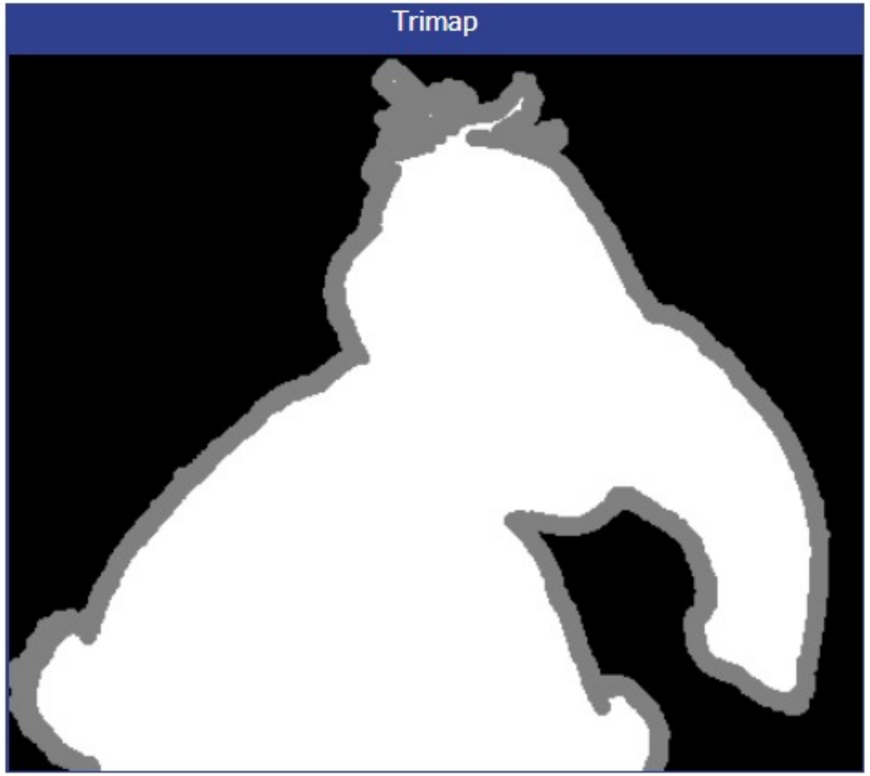
Сложности

- Пример тень
- Сложности для мэттинга человека: Волосы, очки, прозрачная одежда, бутылки и т.д

What is matting?

- Typical input is trimap and image



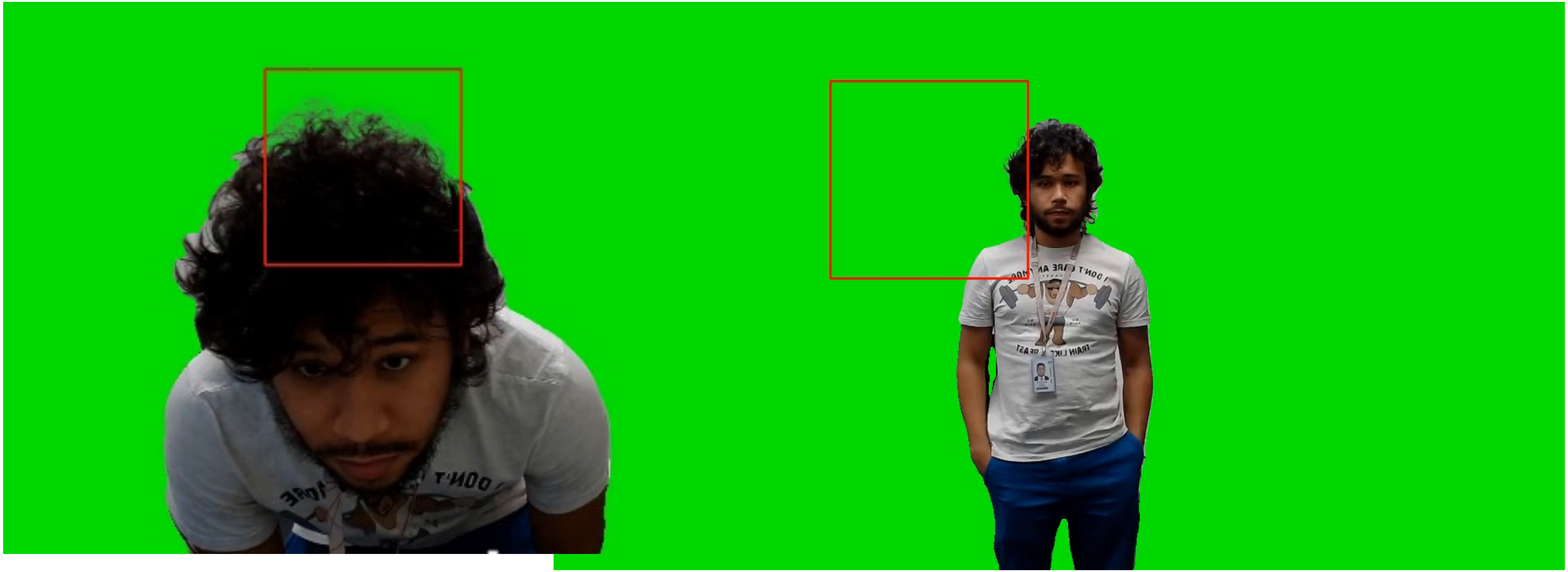


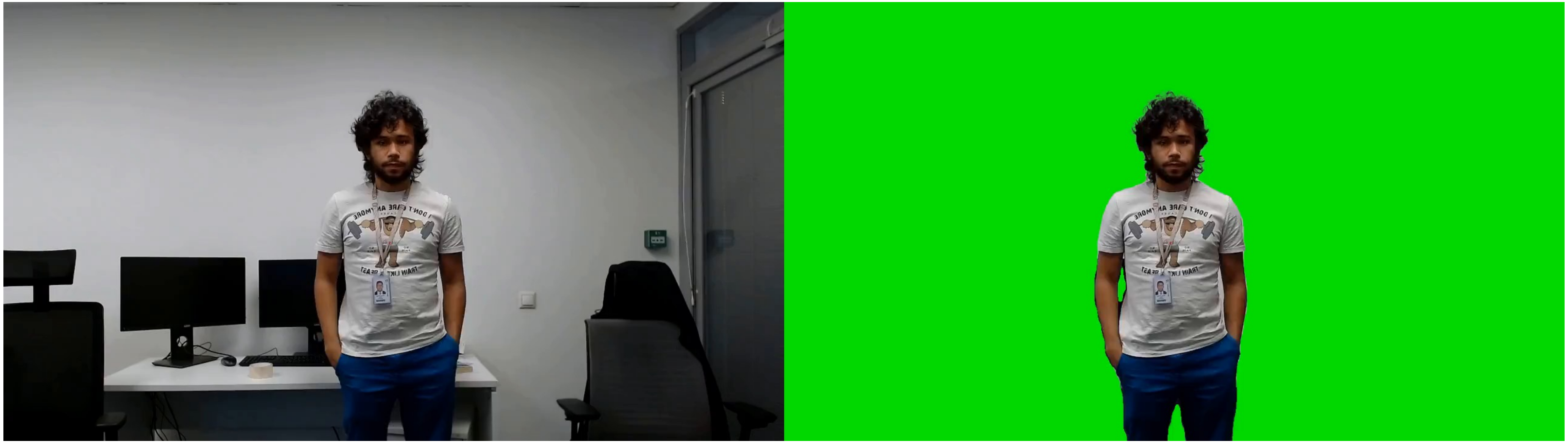
Why we need fractions alphas

- Finite pixel size
- Finite shutter speed
- Motion blur
- Translucency

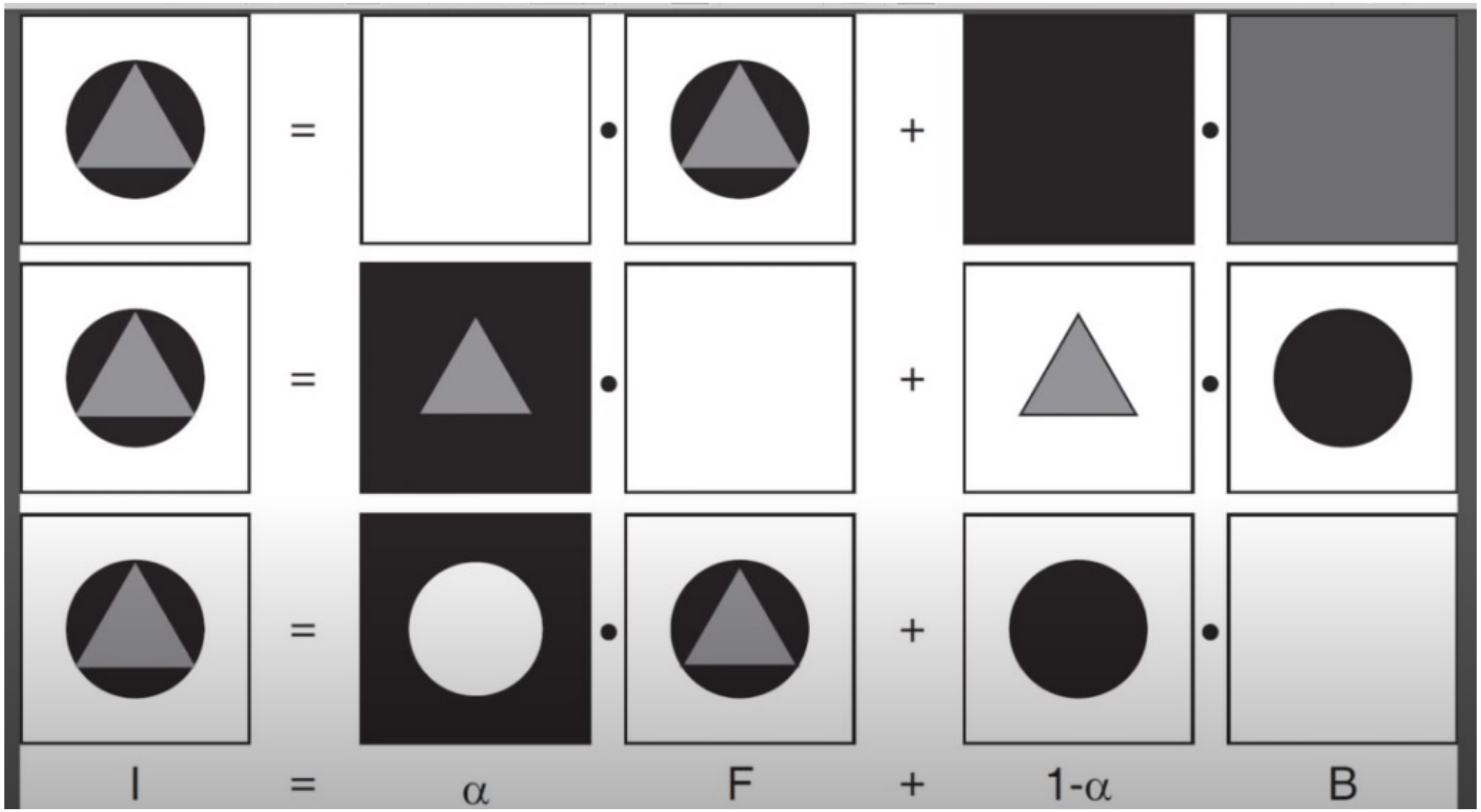
Hard segmentation vs soft segmentation

Hard segmentation vs soft segmentation





Matting ambiguity



Possible solutions

- Trimaps
- Strokes

Applications

- Used in movies
- Communication in Zoom
- Background editing
- Portrait mode in smarphone
- Bokeh effect
- Etc.

How to compare matting models?

- SAD
- MSE
- Grad
- Connectivity

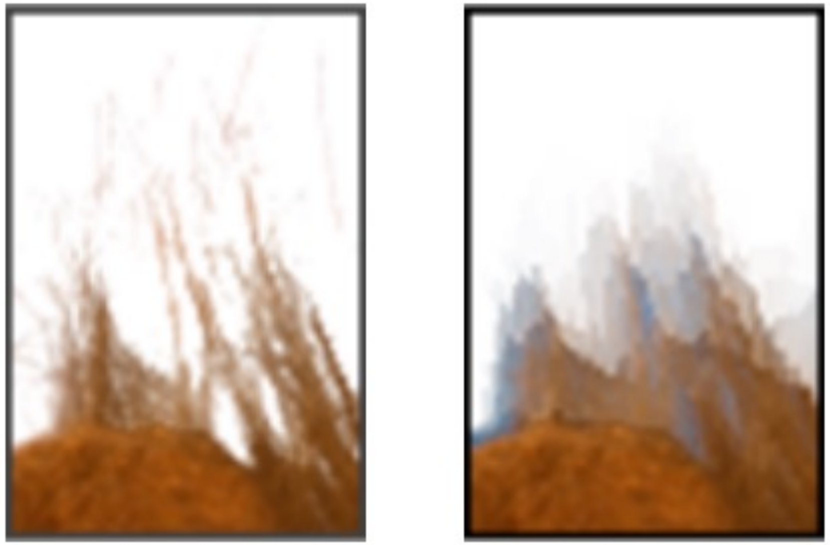
Metrics

- SAD $= \sum_i |\boldsymbol{\alpha_i} - \boldsymbol{\widehat{\alpha}_i}| \cdot [\boldsymbol{trimap_i} == \boldsymbol{gray}]/\mathbf{1000}$
- MSE $= \frac{\sum_i (\boldsymbol{\alpha_i} - \boldsymbol{\widehat{\alpha}_i})^{\mathbf{2}} \cdot [\boldsymbol{trimap_i} == \boldsymbol{gray}]}{\sum_i [\boldsymbol{trimap_i} == \boldsymbol{gray}]}$

\+

- Gradient loss
- Connectivity error

Why we need gradient and connectivity?

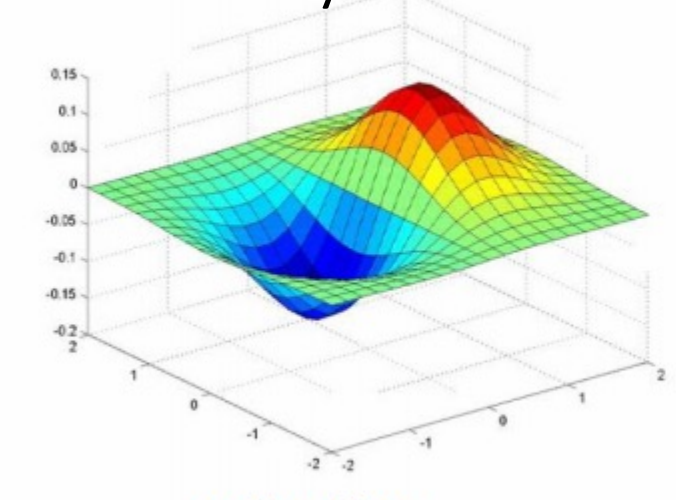


(g) SAD: 1215 (h) SAD: 806

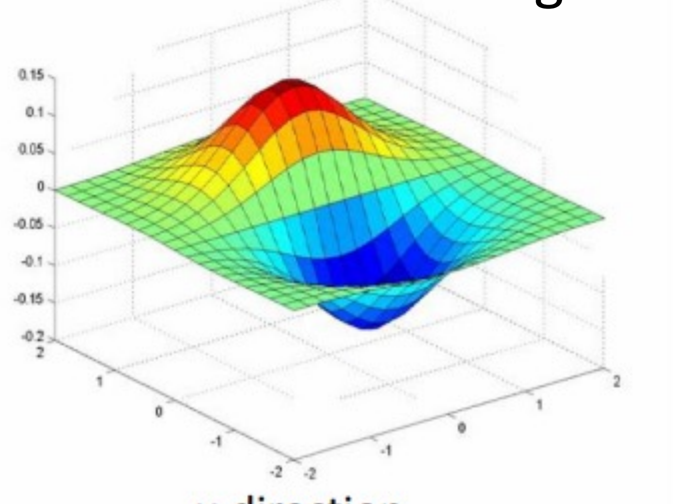
SAD and MSE not always correlate with visual quality

Gradient loss

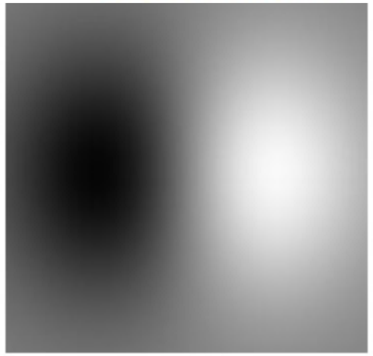
- Gradient loss $= \sum_i (\boldsymbol{\nabla\alpha_i} - \boldsymbol{\nabla\hat{\alpha}_i})^2 [\boldsymbol{trimap_i == gray}]$
  - Gradient $\nabla\alpha$ is calculated by convolution of α with derivative of gaussian kernel:

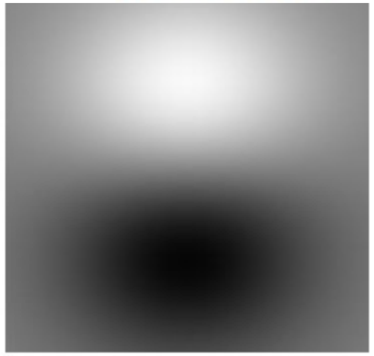


x-direction

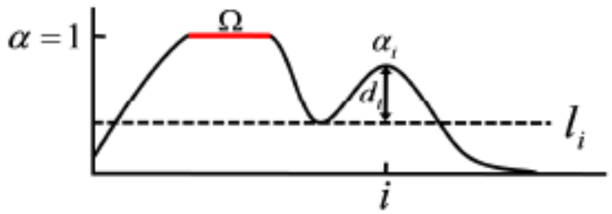


y-direction

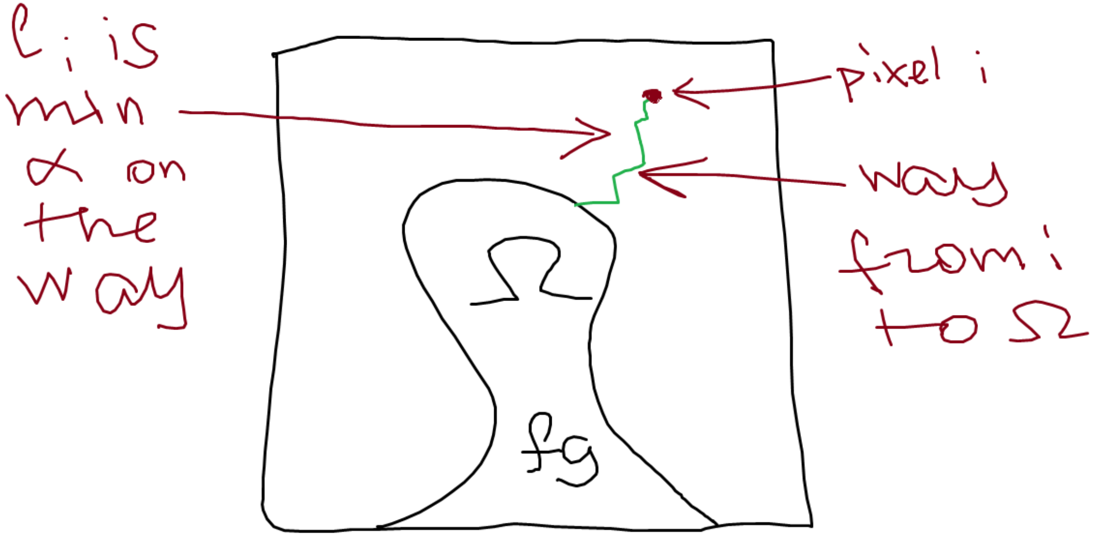


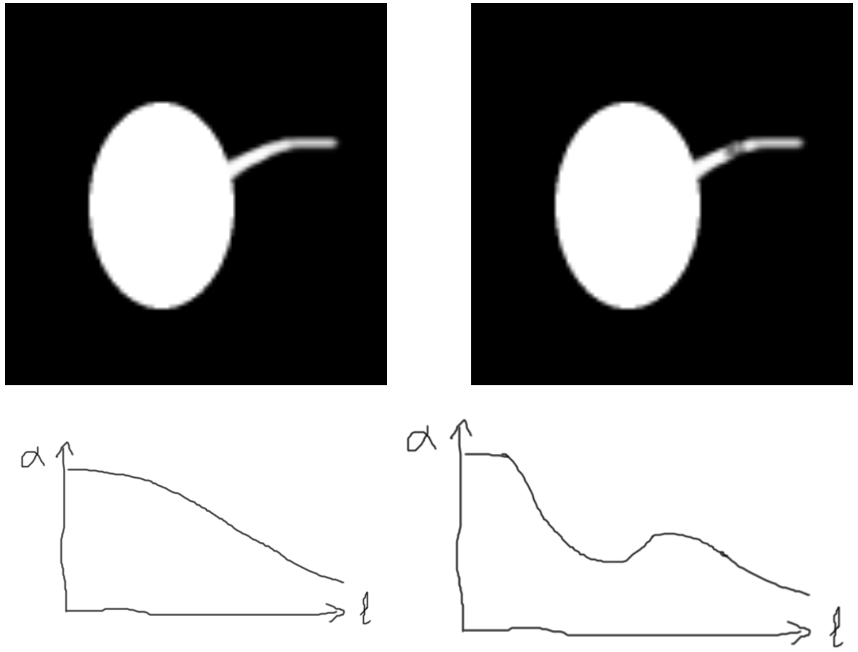


Connectivity Error



- $d_i = \alpha_i - l_i$
- Degree of connectivity: $\boldsymbol{\varphi(\alpha)_i = 1 - d_i}$
  - Actually $\boldsymbol{\varphi(\alpha)_i = 1 - d_i} \cdot [d_i \geq 0.15]$ to neglect small variations
- Conn. error $= \sum_i |\boldsymbol{\varphi(\alpha)_i - \varphi(\hat{\alpha})_i}| \cdot [\boldsymbol{trimap_i == gray}]$



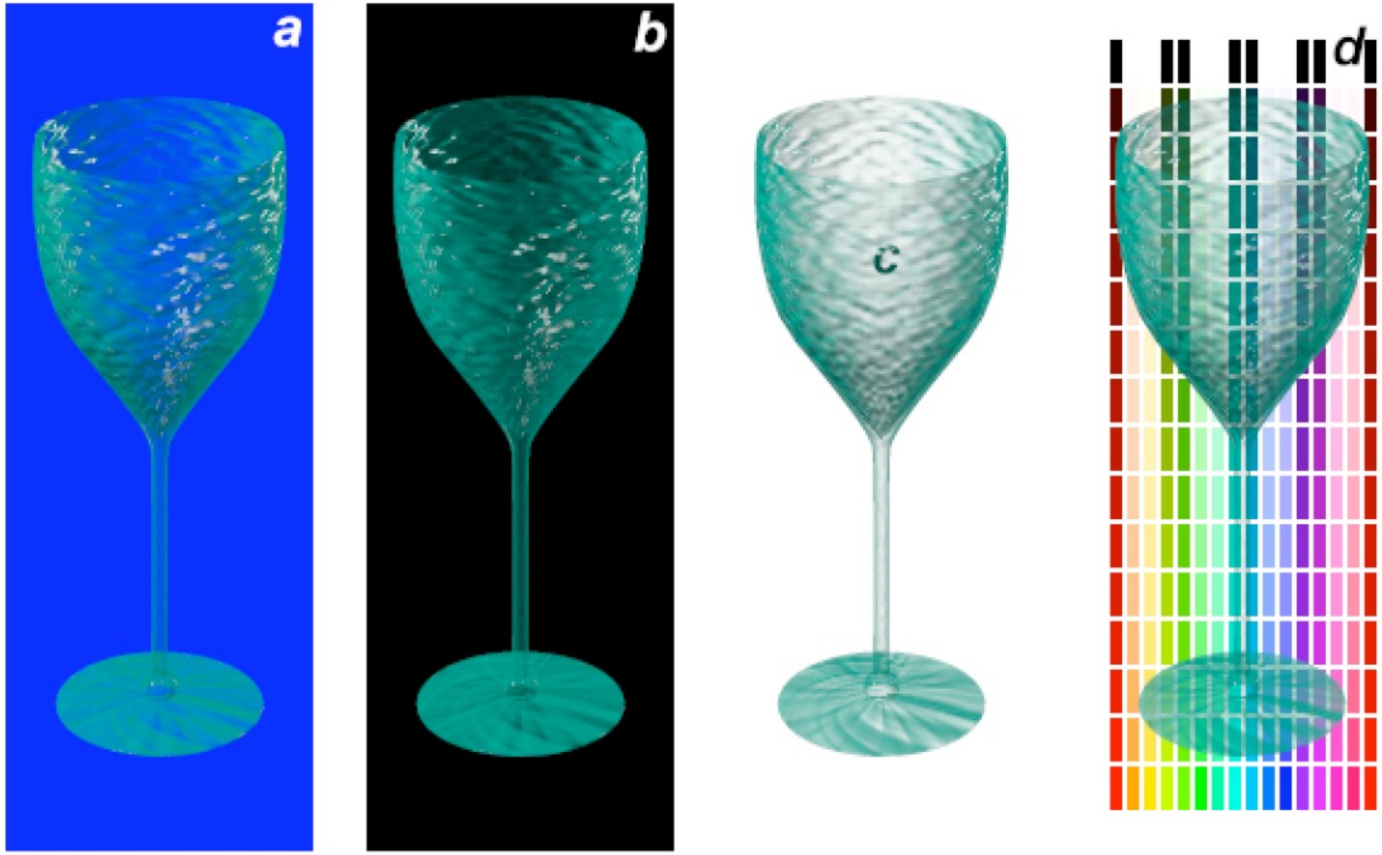


Classical Methods

- Bluescreen matting
- Closed-form matting

Blue Screen Matting

- It's impossible to solve equations for one BG
- But it's possible for 2 BG



Closed-form matting: Gray-scale case

- I is gray-scale image

$$\alpha_i \approx aI_i + b, \quad \forall i \in w,$$

where $a = \frac{1}{F-B}$, $b = -\frac{B}{F-B}$ and w is a small image window.

- Optimization problem

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - a_j I_i - b_j)^2 + \varepsilon a_j^2 \right)$$

Theorem 1 *Define* $J(\alpha)$ *as*

$$J(\alpha) = \min_{a,b} J(\alpha, a, b).$$

Then

$$J(\alpha) = \alpha^T L \, \alpha, \qquad (4)$$

where L *is an* $N \times N$ *matrix, whose* (i, j)*-th entry is:*

$$\sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + \frac{1}{\frac{\varepsilon}{|w_k|} + \sigma_k^2} (I_i - \mu_k)(I_j - \mu_k) \right) \right) \qquad (5)$$

Here δ_{ij} *is the Kronecker delta,* μ_k *and* σ_k^2 *are the mean and variance of the intensities in the window* w_k *around* k*, and* $|w_k|$ *is the number of pixels in this window.*

Closed-form matting: Color case

- Color line model $\alpha_i \approx \sum_c a^c I_i^c + b, \quad \forall i \in w$ $\Longleftrightarrow$ $F_i = \beta_i F_1 + (1 - \beta_i) F_2$

Using the 4D linear model (9) we define the following cost function for matting of RGB images:

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} \left(\alpha_i - \sum_c a_j^c I_i^c - b_j \right)^2 + \varepsilon \sum_c a_j^{c^2} \right) \tag{10}$$

Similarly to the grayscale case, a^c and b can be eliminated from the cost function, yielding a quadratic cost in the α unknowns alone:

$$J(\alpha) = \alpha^T L \, \alpha. \tag{11}$$

Here L is an $N \times N$ matrix, whose (i, j)-th element is:

$$\sum_{k | (i,j) \in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + (I_i - \mu_k)(\Sigma_k + \frac{\varepsilon}{|w_k|} I_3)^{-1} (I_j - \mu_k) \right) \right) \tag{12}$$

Theorem 3 *Let I be an image formed from F and B according to (1), and let α^* denote the true alpha matte. If F and B satisfy the color line model in every local window w_k, and if the user-specified constraints S are consistent with α^*, then α^* is an optimal solution for the system (13), where L is constructed with $\varepsilon = 0$.*

Proof: Since $\varepsilon = 0$, if the color line model is satisfied in every window w_k, it follows from the definition (10) that $J(\alpha^*, a, b) = 0$, and therefore $J(\alpha^*) = \alpha^{*T} L \, \alpha^* = 0$. $\square$

Modern methods

- FBA
- Background matting
- Modnet

FBA Matting:Network architecture

- 9 input channels: 3 – image, 6 – definite foreground and background with Gaussian blurs at three different scales
- 7 output channels: 1 for alpha, 3 for F, 3 for B
- Encoder-decoder with Unet style
- Encoder – Resnet50 with removed striding from layers 3 and 4, and increased the dilation to 2 and 4 respectively
- Decoder – Pyramid Pooling layer and some convolutions
- Hardtanh activation to predict alpha, sigmoid for F and B

Batch normalization vs Group Normalization

Model	Norm.	Batch-Size	Loss	MSE	SAD	GRAD	CONN
Training at 20 epochs:							
(1)	BN	6	$\mathcal{L}_1^\alpha$	11.2	36.3	14.9	32.5
(2)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha$	9.1	34.5	15.0	31.3
(3)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha$	7.4	33.5	12.9	28.5
(4)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	8.1	36.3	13.8	32.0
(5)	GN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	10.3	36.2	15.1	32.0
(6)	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	7.2	32.8	13.3	28.6
(7)	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha + \text{clip}_\alpha$	6.9	31.2	12.9	27.1
Training at 45 epochs:							
Ours$_\alpha$	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha + \text{clip}_\alpha$	5.3	26.5	10.6	21.8

F, B, α Losses

$\boldsymbol{\alpha}$ Losses	$\mathbf{F}, \mathbf{B}$ Losses
$\mathcal{L}_1^\alpha = \sum_i \lVert \hat{\boldsymbol{\alpha}}_i - \boldsymbol{\alpha}_i \rVert_1$	$\mathcal{L}_1^{\mathrm{FB}} = \sum_i \left\lVert \hat{\mathbf{F}}_i - \mathbf{F}_i \right\rVert_1 + \left\lVert \hat{\mathbf{B}}_i - \mathbf{B}_i \right\rVert_1$
$\mathcal{L}_c^\alpha = \sum_i \lVert \mathbf{C}_i - \hat{\boldsymbol{\alpha}}_i \mathbf{F}_i - (1 - \hat{\boldsymbol{\alpha}}_i)\mathbf{B}_i \rVert_1$	$\mathcal{L}_{\mathrm{excl}}^{\mathrm{FB}} = \sum_i \lVert \nabla \mathbf{F}_i \rVert_1 \lVert \nabla \mathbf{B}_i \rVert_1$
$\mathcal{L}_{\mathrm{lap}}^\alpha = \sum_{s=1}^{5} 2^{s-1} \left\lVert L_{\mathrm{pyr}}^s(\boldsymbol{\alpha}) - L_{\mathrm{pyr}}^s(\hat{\boldsymbol{\alpha}}) \right\rVert_1$	$\mathcal{L}_c^{\mathrm{FB}} = \sum_i \left\lVert \mathbf{C}_i - \boldsymbol{\alpha}_i \hat{\mathbf{F}} - (1 - \boldsymbol{\alpha}_i)\hat{\mathbf{B}} \right\rVert_1$
$\mathcal{L}_g^\alpha = \sum_i \lVert \nabla \hat{\boldsymbol{\alpha}}_i - \nabla \boldsymbol{\alpha}_i \rVert_1$	$\mathcal{L}_{\mathrm{lap}}^{\mathrm{FB}} = \mathcal{L}_{\mathrm{lap}}^{\mathbf{F}} + \mathcal{L}_{\mathrm{lap}}^{\mathbf{B}}$

$$\mathcal{L}_{FB\alpha} = \mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_g^\alpha + \mathcal{L}_{\mathrm{lap}}^\alpha + 0.25\left(\mathcal{L}_1^{\mathrm{FB}} + \mathcal{L}_{\mathrm{lap}}^{\mathrm{FB}} + \mathcal{L}_{\mathrm{excl}}^{\mathrm{FB}} + \mathcal{L}_c^{\mathrm{FB}}\right)$$

F, B, α Fusion

- We can improve the prediction $\mathbf{C}_i = \boldsymbol{\alpha}_i \mathbf{F}_i + (1 - \boldsymbol{\alpha}_i)\mathbf{B}_i$
- Simplified likelihood model:

$$p(\boldsymbol{\alpha}, \mathbf{F}, \mathbf{B} | \hat{\boldsymbol{\alpha}}, \hat{\mathbf{F}}, \hat{\mathbf{B}}) \propto p(\boldsymbol{\alpha}|\hat{\boldsymbol{\alpha}}) p(\mathbf{F}|\hat{\mathbf{F}}) p(\mathbf{B}|\hat{\mathbf{B}}) p(\boldsymbol{\alpha}, \mathbf{F}, \mathbf{B})$$

- Assuming Gaussian distribution for errors:

$$p(\mathbf{F}|\hat{\mathbf{F}}) \propto \exp\left(-\frac{\|\mathbf{F} - \hat{\mathbf{F}}\|_2^2}{2\sigma_{FB}^2}\right) \qquad p(\mathbf{B}|\hat{\mathbf{B}}) \propto \exp\left(-\frac{\|\mathbf{B} - \hat{\mathbf{B}}\|_2^2}{2\sigma_{FB}^2}\right)$$

$$p(\boldsymbol{\alpha}|\hat{\boldsymbol{\alpha}}) \propto \exp\left(-\frac{(\boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}})^2}{2\sigma_{\alpha}^2}\right) \qquad p(\boldsymbol{\alpha}, \mathbf{F}, \mathbf{B}) \propto \exp\left(-\frac{\|\mathbf{C} - \boldsymbol{\alpha}\mathbf{F} - (1 - \boldsymbol{\alpha})\mathbf{B}\|_2^2}{2\sigma_C^2}\right)$$

F, B, α Fusion

Update step (even one step is enough):

$$\hat{\mathbf{F}}^{(n+1)} = \hat{\mathbf{F}} + \frac{\sigma_F^2}{\sigma_{\mathbf{C}}^2}\hat{\boldsymbol{\alpha}}^{(n)}\left(\mathbf{C} - \hat{\boldsymbol{\alpha}}^{(n)}\hat{\mathbf{F}}^{(n)} - (1 - \hat{\boldsymbol{\alpha}}^{(n)})\hat{\mathbf{B}}^{(n)}\right)$$

$$\hat{\mathbf{B}}^{(n+1)} = \hat{\mathbf{B}} + \frac{\sigma_B^2}{\sigma_{\mathbf{C}}^2}(1 - \hat{\boldsymbol{\alpha}}^{(n)})\left(\mathbf{C} - \hat{\boldsymbol{\alpha}}^{(n)}\hat{\mathbf{F}}^{(n)} - (1 - \hat{\boldsymbol{\alpha}}^{(n)})\hat{\mathbf{B}}^{(n)}\right)$$

$$\hat{\boldsymbol{\alpha}}^{(n+1)} = \frac{\hat{\boldsymbol{\alpha}}^{(n)} + \frac{\sigma_\alpha^2}{\sigma_{\mathbf{C}}^2}(\mathbf{C} - \hat{\mathbf{B}}^{(n+1)})^\top(\hat{\mathbf{F}}^{(n+1)} - \hat{\mathbf{B}}^{(n+1)})}{1 + \frac{\sigma_\alpha^2}{\sigma_{\mathbf{C}}^2}(\hat{\mathbf{F}}^{(n+1)} - \hat{\mathbf{B}}^{(n+1)})^\top(\hat{\mathbf{F}}^{(n+1)} - \hat{\mathbf{B}}^{(n+1)})}$$

Ablation study

Model	$+\mathcal{L}_{FB}$	$+\mathcal{L}_{\text{excl}}$	output	$\alpha\mathbf{F}$ SAD	$\alpha\mathbf{F}$ MSE	α SAD	α MSE
Closed-form Matting [20]				251.67	22.96	161.3	85.3
Context-Aware Matting [13]				70.00	11.49	38.1	8.9
Training at 20 epochs:							
(6)	N	N	sigmoid	-	-	32.8	7.2
(8)	Y	N	sigmoid	53.64	9.04	32.7	9.0
(9)	Y	Y	sigmoid	52.87	8.88	31.8	8.9
(7)	N	N	clip	-	-	31.2	6.9
(10)	Y	Y	clip	50.69	8.64	31.3	8.6
(11)	Y*	Y	clip	50.29	8.48	32.1	8.5
Training at 45 epochs:							
(11)	Y*	Y	clip	42.19	6.50	26.5	5.4
Ours$_{\text{FB}\alpha}$	Y*	Y	clip +fusion	39.21	6.19	26.4	5.4
Ours$_{\text{FB}\alpha}$	Y*	Y	clip +fusion +TTA	38.81	5.98	25.8	5.2

Ablation Study

Model	Norm.	Batch-Size	Loss	MSE	SAD	GRAD	CONN
Training at 20 epochs:							
(1)	BN	6	$\mathcal{L}_1^\alpha$	11.2	36.3	14.9	32.5
(2)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha$	9.1	34.5	15.0	31.3
(3)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha$	7.4	33.5	12.9	28.5
(4)	BN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	8.1	36.3	13.8	32.0
(5)	GN	6	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	10.3	36.2	15.1	32.0
(6)	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha$	7.2	32.8	13.3	28.6
(7)	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha + \text{clip}_\alpha$	6.9	31.2	12.9	27.1
Training at 45 epochs:							
Ours$_\alpha$	GN	1	$\mathcal{L}_1^\alpha + \mathcal{L}_c^\alpha + \mathcal{L}_{\text{lap}}^\alpha + \mathcal{L}_g^\alpha + \text{clip}_\alpha$	5.3	26.5	10.6	21.8

Results

Method	SAD	MSE x10^3	Gradient	Connectivity
Closed-Form Matting [20]	168.1	91.0	126.9	167.9
KNN-Matting [4]	175.4	103.0	124.1	176.4
DCNN Matting [5]	161.4	87.0	115.1	161.9
Information-flow Matting [1]	75.4	66.0	63.0	-
Deep Image Matting [37]	50.4	14.0	31.0	50.8
AlphaGan-Best [25]	52.4	30.0	38.0	-
IndexNet Matting [24]	45.8	13.0	25.9	43.7
VDRN Matting [33]	45.3	11.0	30.0	45.6
AdaMatting [3]	41.7	10.2	16.9	-
Learning Based Sampling [34]	40.4	9.9	-	-
Context Aware Matting [13]	35.8	8.2	17.3	33.2
GCA Matting [21]	35.3	9.1	16.9	32.5
Ours$_\alpha$	26.5	5.3	10.6	21.8
Ours$_{FB\alpha}$	26.4	5.4	10.6	21.5
Ours$_{FB\alpha}$ **TTA**	**25.8**	**5.2**	**10.6**	**20.8**

Background Matting

- https://grail.cs.washington.edu/projects/background-matting-v2/

ModNet

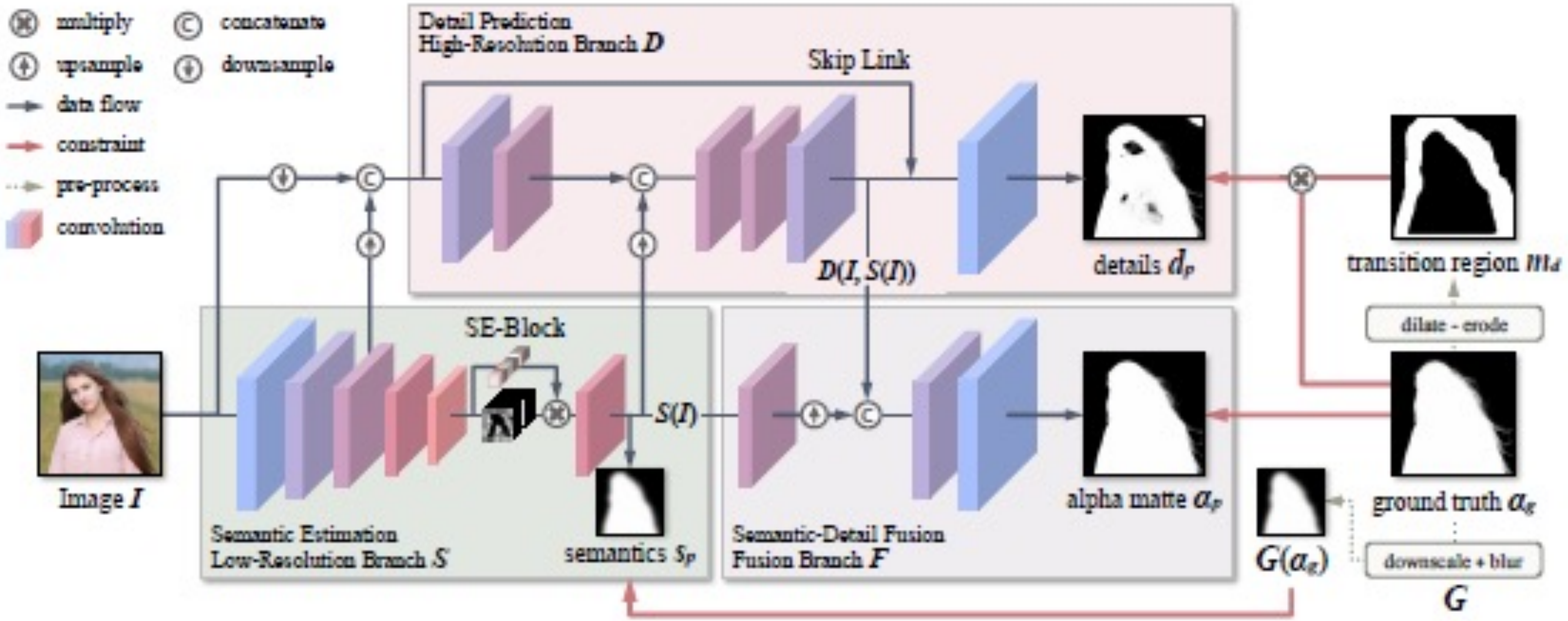


Figure 2. **Architecture of MODNet.** Given an input image I, MODNet predicts human semantics s_p, boundary details d_p, and final alpha matte α_p through three interdependent branches, S, D, and F, which are constrained by specific supervisions generated from the ground truth matte α_g. Since the decomposed sub-objectives are correlated and help strengthen each other, we can optimize MODNet end-to-end.