Poisson Matting

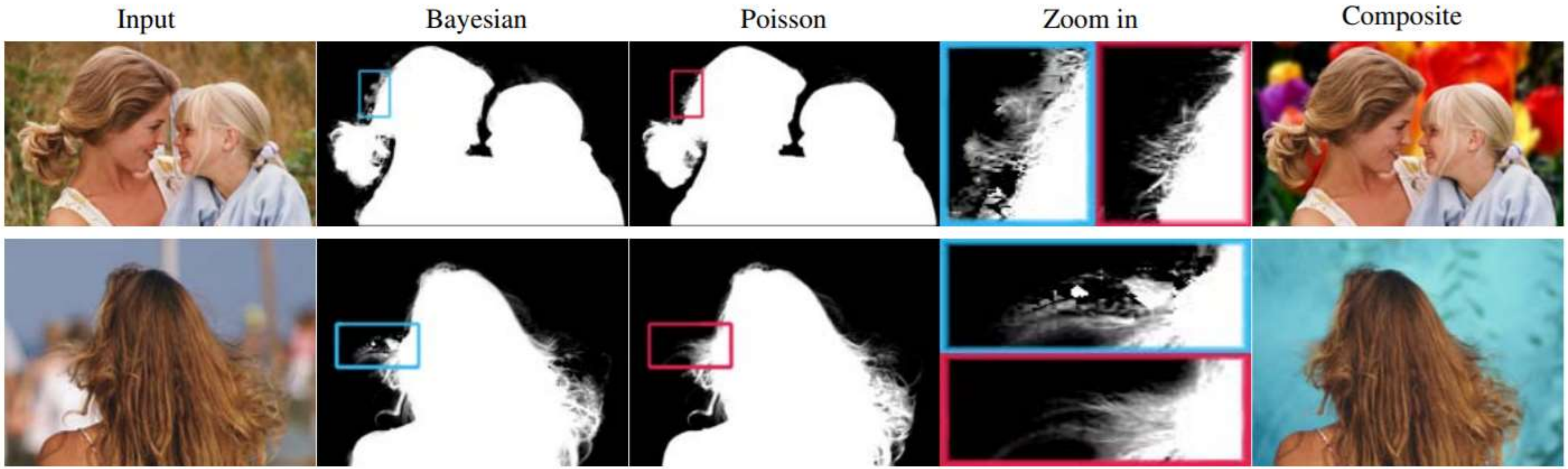
Jian Sun[1] Jiaya Jia[2]* Chi-Keung Tang[2] Heung-Yeung Shum[1]

[1]Microsoft Research Asia [2]Hong Kong University of Science and Technology

ACM Transactions on Graphics (TOG) Volume 23 Issue 3, August 2004

Pages 315-321

What is Poisson matting?



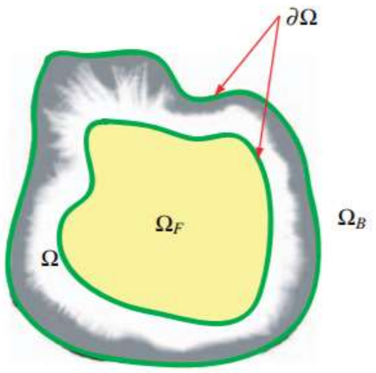
Poisson Matting

Global Poisson matting

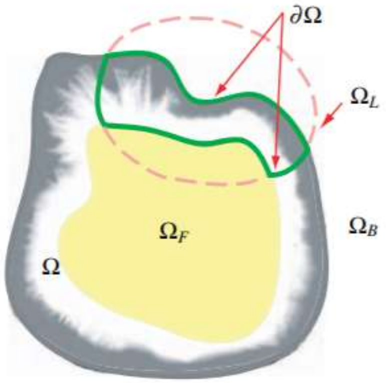
If it fails to produce high quality mattes

due to a complex background

Local Poisson matting



(a)



(b)

Global Poisson Matting

$$\nabla\alpha \approx \frac{1}{F-B}\nabla I$$

For each pixel $p = (x, y)$,

$$\alpha^* = \arg\min_{\alpha} \int \int_{p\in\Omega} \left\| \nabla\alpha_p - \frac{1}{F_p - B_p}\nabla I_p \right\|^2 dp$$

with $\alpha|_{\partial\Omega} = \hat{\alpha}|_{\partial\Omega}$

which is defined $\widehat{\alpha_p}|_{\partial\Omega} = \begin{cases} 1 & p \in \Omega_F \\ 0 & p \in \Omega_B \end{cases}$

$$\Delta\alpha = div\left(\frac{\nabla I}{F-B}\right)$$

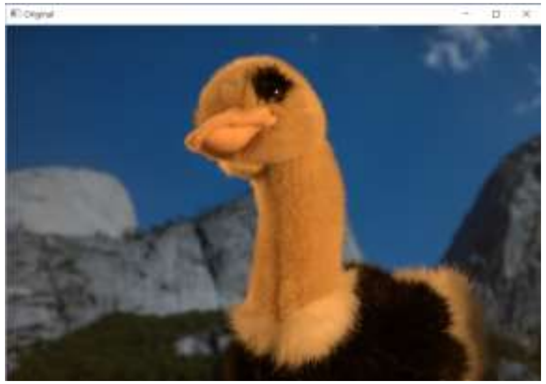
Iterative optimization

Global Poisson matting is an iterative optimization process:

1. (F - B) initialization : For each pixel p in Ω, F_p and B_p are approximated by corresponding the nearest foreground pixel in Ω_F and background pixel in Ω_B.
2. α reconstruction
3. F, B refinement

$$\Omega_F^+ = \{p \in \Omega | \alpha_p > 0.95, I_p \approx F_p\}$$

$$\Omega_B^+ = \{p \in \Omega | \alpha_p < 0.05, I_p \approx B_p\}$$




Alpha

Global Poisson Matting in the colourscale channel

Problem

$$\Delta\alpha = div(\frac{\nabla I}{F - B})$$

Grey Scale

Colour Scale

Solution

$$(F - B)\nabla\alpha = \nabla I$$

$$(F - B) \cdot (F - B)\nabla\alpha = (F - B) \cdot \nabla I$$

$$\nabla\alpha = \frac{(F - B) \cdot \nabla \mathrm{I}}{\left|\left|\mathrm{F} - \mathrm{B}\right|\right|^2}$$

Global Poisson Matting

1. Set ∇I

```
//gradient x, y
Mat Ix, Iy;
gradient(img, Ix, Iy);
```

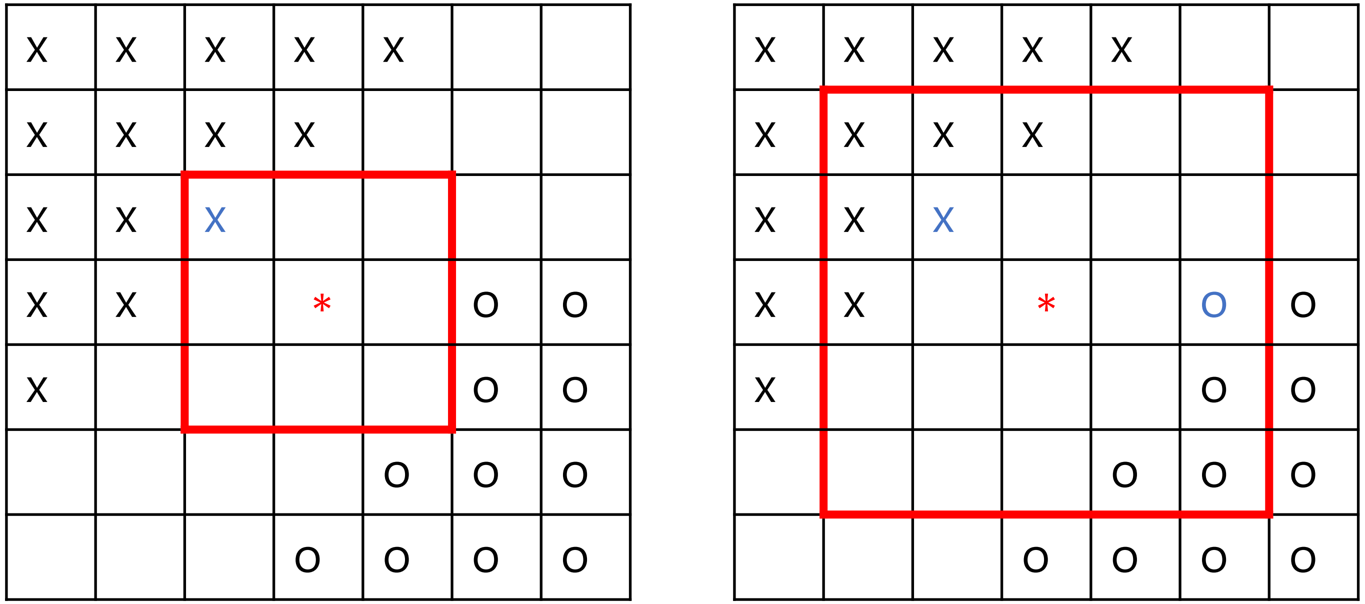
```
void gradient(cv::Mat src, cv::Mat& x, cv::Mat& y) {
        x.create(src.size(), CV_32FC3);
        y.create(src.size(), CV_32FC3);

        for (int i = 0; i < src.rows; i++){
                for (int j = 0; j < src.cols; j++){
                        if (i == 0) y.at < Vec3f >(i, j) = src.at<Vec3f>(i + 1, j) - src.at<Vec3f>(i, j);
                        else if (i == src.rows - 1) y.at<Vec3f>(i, j) = src.at<Vec3f>(i, j) - src.at<Vec3f>(i - 1, j);
                        else y.at<Vec3f>(i, j) = (Vec3f)(src.at<Vec3f>(i + 1, j) - src.at<Vec3f>(i - 1, j)) / 2;

                        if (j == 0) x.at<Vec3f>(i, j) = src.at<Vec3f>(i, j + 1) - src.at<Vec3f>(i, j);
                        else if (j == src.cols - 1) x.at<Vec3f>(i, j) = src.at<Vec3f>(i, j) - src.at<Vec3f>(i, j - 1);
                        else x.at<Vec3f>(i, j) = (Vec3f)(src.at<Vec3f>(i, j + 1) - src.at<Vec3f>(i, j - 1)) / 2;
                }}}
```

Global Poisson Matting

2. Find the nearest F and B from the unknown pixel and initialise (F-B) value



x : Definitely background

o : Definitely foreground

<How to find the nearest F and B>

Global Poisson Matting

3. α reconstruction

For each pixel $p = (x, y)$,

$$\alpha^* = \arg\min_{\alpha} \int \int_{p \in \Omega} \left\| \nabla \alpha_p - \frac{(F - B) \cdot \nabla \mathrm{I}}{||\mathrm{F} - \mathrm{B}||^2} \right\|^2 dp$$

with $\alpha|_{\partial\Omega} = \hat{\alpha}|_{\partial\Omega}$

which is defined $\widehat{\alpha_p}|_{\partial\Omega} = \begin{cases} 1 & p \in \Omega_F \\ 0 & p \in \Omega_B \end{cases}$

Discrete Poisson solver

$$\min_{f|_\Omega} \sum_{<p,q> \cap \Omega \neq \emptyset} \left(f_p - f_q - v_{pq}\right)^2$$

$with\ f_p = f_p^*, for\ all\ p \in \partial\Omega$

$For\ all\ p \in \Omega,$

$$\alpha_p = \frac{1}{|N_p|} \left(\sum_{\partial\Omega} \alpha_q^* + \sum_{\mathrm{N}} \alpha_q^0 + \sum \nabla \alpha_{pq}\right)$$

Patrick P´erez∗, Michel Gangnet†, Andrew Blake‡, "Poisson Image Editing ", Microsoft Research UK, 2003

Global Poisson Matting

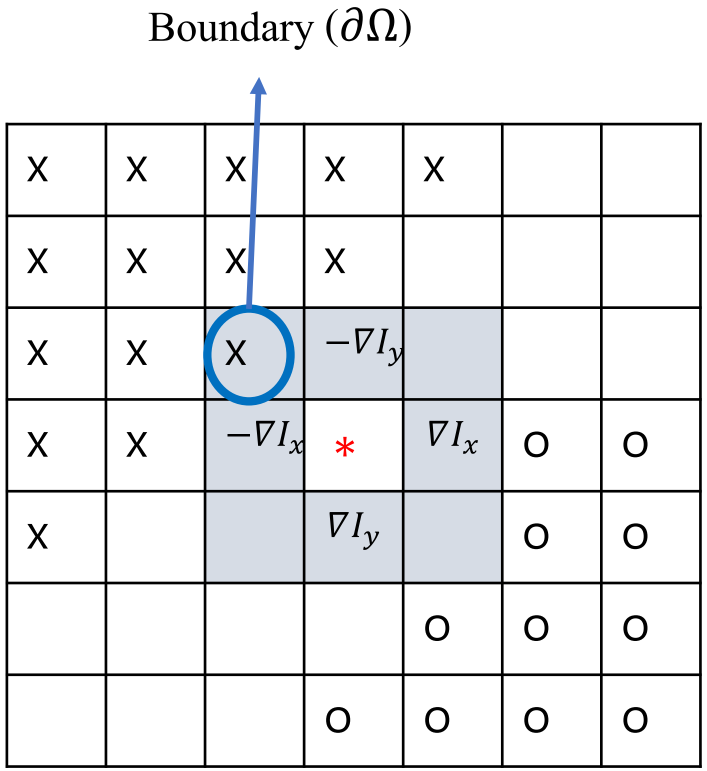
Discrete Poisson solver

For all $p \in \Omega$,

$$\alpha_p = \frac{1}{|N_p|}\left(\sum_{\partial\Omega} \alpha_q^* + \sum_{\mathrm{N}} \alpha_q^0 + \sum \nabla\alpha_{pq}\right)$$

N_p : the set of its 8-connected neighbors

: the set of q



x : Definitely background

o : Definitely foreground

<u>Calculate iteratively the value of α until the difference is so small</u>

Global Poisson Matting

4. F and B refinement

$$\Omega_F^+ = \{p \in \Omega | \alpha_p > 0.95, I_p \approx F_p\}$$
$$\Omega_B^+ = \{p \in \Omega | \alpha_p < 0.05, I_p \approx B_p\}$$

```
Vec3f second_con;

if (1> alpha1.at<float>(y, x)&&alpha1.at<float>(y, x) > 0.95) {
        for (auto& NB_pixel : neighbour_pixels) {
                if (NB_pixel.p == Point(x, y)) {
                        second_con = img.at<Vec3f>(y, x) - NB_pixel.F_pixel;
                        if (dot(second_con, second_con) < 0.0001) {
                                trimap.at<uchar>(y, x) = WHITE;}
                        break;}}}
if (0 < alpha1.at<float>(y, x) && alpha1.at<float>(y, x)< 0.05) {
        for (auto& NB_pixel : neighbour_pixels) {
                if (NB_pixel.p == Point(x, y)) {
                        second_con = img.at<Vec3f>(y, x) - NB_pixel.B_pixel;
                        if (dot(second_con, second_con) < 0.0001) {
                                trimap.at<uchar>(y, x) = BLACK;}
                break;}}}
```

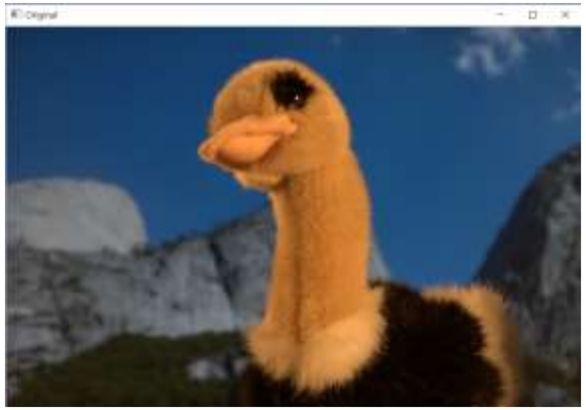
Global Poisson Matting

5. Iterative optimization

```
Mat alpha;
trimap.convertTo(alpha, CV_32F, 1 / 255.f); //from 0 to 1
int result;
for (int i = 0; i < 50; i++) {
		result = Iterative_opti(trimap, img, alpha);
		if (result == 1) break;

		imshow("Alpha", alpha);
		waitKey(10);
}
```

1. Set ∇I
2. Find the nearest F and B from the unknown pixel and initialise (F-B) value
3. α reconstruction
4. F and B refinement
5. Iterative optimization

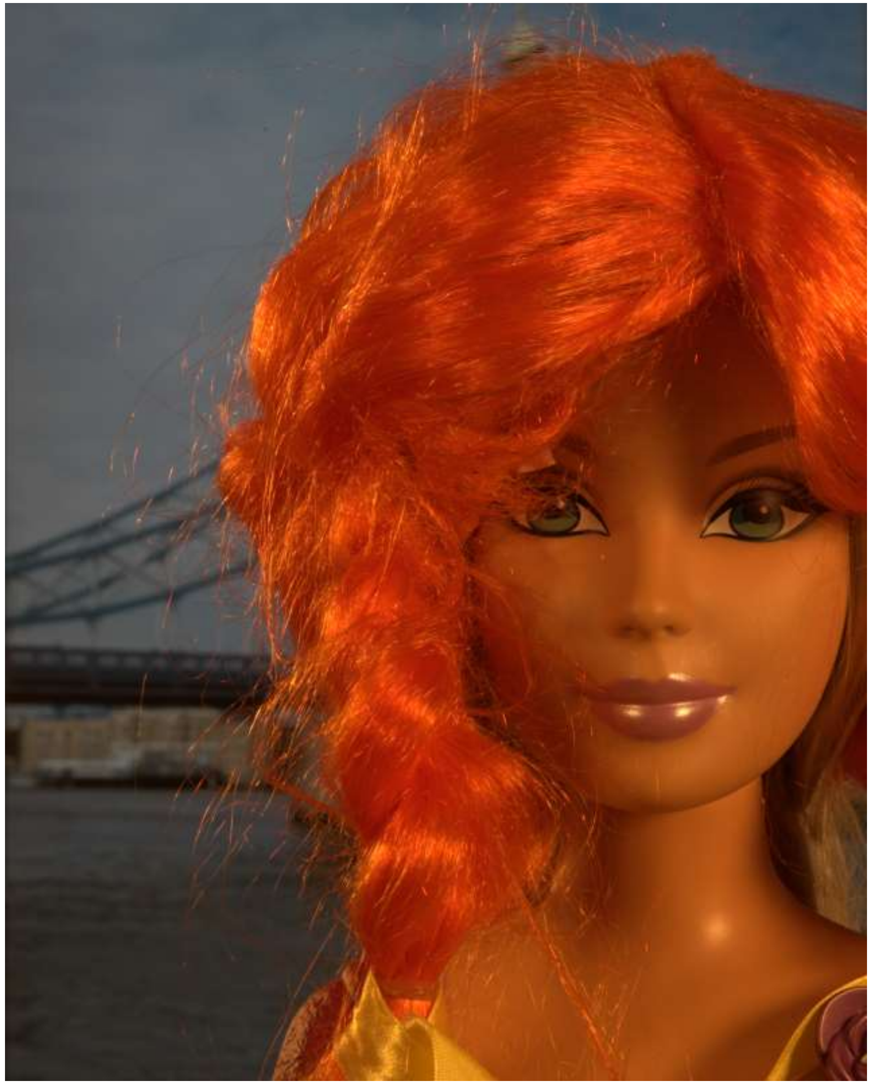



Alpha

Alpha mattes based on the greyscale and the colourscale

Grey

Colour



Local Poisson Matting

Channel selection

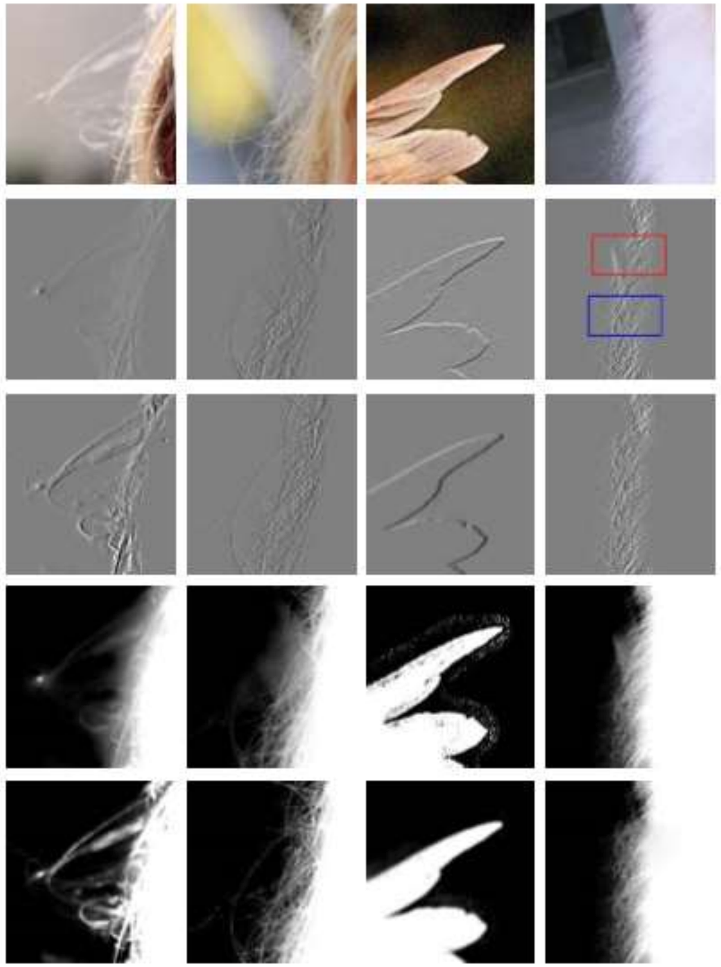
To reduce the error of D

Local filtering

To manipulate A and D

1. Boosting brush
2. Highpass filtering
3. Diffusion filtering
4. Clone brush

Erase brush / Inverse Brush



(a) Boosting (b) Highpass (c) Diffusion (d) Clone

Local Poisson Matting

Channel Selection

$$\min_{a,b,c} \sum_i [(a\ b\ c) \cdot (R_i\ G_i\ B_i)^T - (a\ b\ c) \cdot (\bar{R}\ \bar{G}\ \bar{B})\ ]^2 \ \ s.t.\ a+b+c=1$$

(a) (b) (c)

A new channel $\gamma = aR + bG + cB$

The error of D is reduced and the hair shape is better recovered.