

CM

```
% Input
% read RGB images and Trimap
% Use GUI
% core: imread
function [RGB, Trimap] = read_image(RGB_path, Trimap_path)

% test: resolution, RGB channel([0, 255]), trimap channel([1,3])
```

```
% match two images
function [coF, coB] = match_images(RGB, Trimap)
```

```
% Bayesian Matting
function [position_x, position_y, alpha] = bayesian_matting(coF, coB)

% test: 0 <= position_x, position_y <= resolution,
% test: 0 <= alpha <= 1
```

```
% Laplacian Matting
% https://github.com/frcs/alternative-matting-laplacian
function [laplacian_image] = laplacian_matting(path)
```

```
% Evaluation
% Mean Absolute Deviation (MAD): In  $[0, \infty)$ , the smaller the better
% Root Mean Squared Error (RMSE): In  $[0, \infty)$ , the smaller the better
% Median Absolute Error (MAE): In  $[0, \infty)$ , the smaller the better
% Mean Squared Log Error (MSLE): In  $[0, \infty)$ , the smaller the better
%  $R^2$ , coefficient of determination: In  $(-\infty, 1]$ , not necessarily 1
function [MAD, RMSE, MAE, MSLE,  $R^2$ ] = evaluation(matting_image, ground_truth)
```

```
% Output  
function save(path, matting_image)
```