Bayesian matting

**Introduction:**

We have successfully implemented a Bayesian matting algorithm in MATLAB, along with supporting functions such as extractNeighborhoodInfo, TransformMatrix, TransformPixels, likelihoodEstimation, and Mathcomputation. In our main script, we utilized these functions to process the original image, trimap, and ground truth data. Our process involved performing foreground-background (FB) split, generating alpha matte, and refining the foreground with the new background estimation.

Furthermore, we conducted a comparative study between Bayesian matting and Laplacian matting techniques. To evaluate the performance of our implementation, we employed two key performance metrics: Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE).

**Main Steps:**

* **Data Preparation:**
* Read the original image, trimap, and ground truth data.
* Bayesian Matting Algorithm:
* Perform FB split to separate the foreground and background regions.
* Generate alpha matte to capture the transparency levels of the image.
* Refine the foreground with the new background estimation.
* **Comparative Study:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Bayesian** | | | **Laplacian** | | |
| **Image Name** | **MSE** | **PSNR** | **Running Time** | **MSE** | **PSNR** | **Running**  **Time** |
| **Flower** | 0.0006 | 32.21895 | 28.7546 | 0.04187 | 13.78147 | 0.11465 |
| **Coil** | 0.01217 | 19.1484 | 69.8519 | 0.03814 | 14.18635 | 0.11717 |
| **Doll** | 0.01717 | 17.65264 | 168.7496 | 0.15603 | 8.06799 | 0.14342 |
| **Pumpkin** | 0.00052 | 32.86391 | 17.0989 | 0.02734 | 15.6315 | 0.14421 |

**Performance Evaluation:**

* The findings suggest that while Laplacian Matting may offer quicker processing times, the sacrifice in image quality renders Bayesian Matting a more favorable choice for tasks that prioritize output quality.