

Name (netid): Your Name (Your Netid)
CS 445 - Project 3: Gradient Domain Fusion

Complete the claimed points and sections below.

Total Points Claimed [100] / 160

Core

- | | |
|--------------------------------|-------------|
| 1. Toy Problem | [20] / 20 |
| 2. Poisson blending | [50] / 50 |
| 3. Mixed gradients | [20] / 20 |
| 4. Quality of results / report | [10] / 10 |

B&W

- | | |
|------------------------------------|----------|
| 5. Color2Gray | [] / 20 |
| 6. Laplacian Pyramid Blending | [] / 20 |
| 7. More gradient domain processing | [] / 20 |

1. Toy problem

Include

- The “Max error is: xxxx” line
 - Max error is: 7.420508651989621e-12

2. Poisson blending

Using your own images (not sample images), include:

- Your favorite blending result, including: (1) background and object images; (2) pasted image with source pixels directly copied onto target background region (can use `utils.get_combined_img`); (3) final blend result. (30 pts)

background image:



object image:



pasted image:

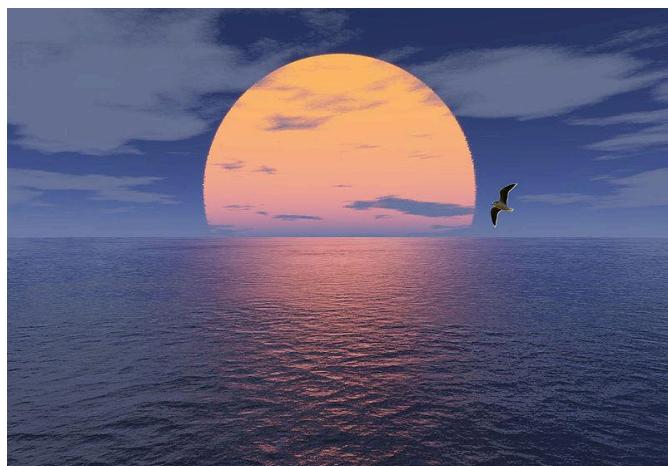


final blended result:



- At least one more good result (10 pts)

background image:



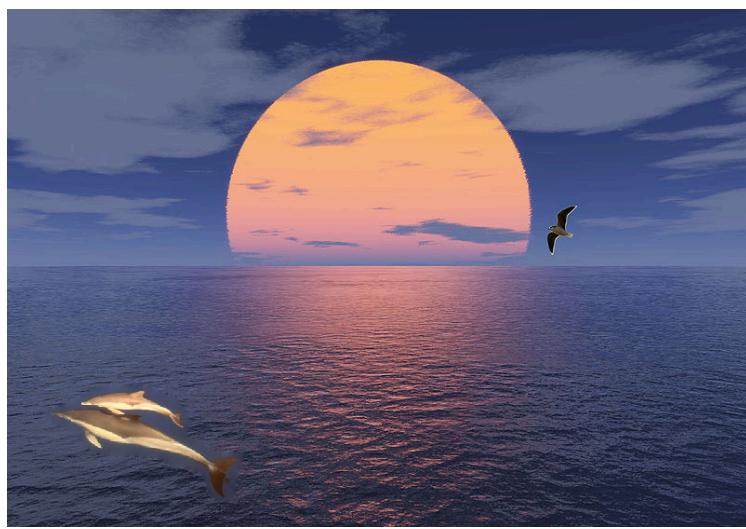
object image:



pasted image:

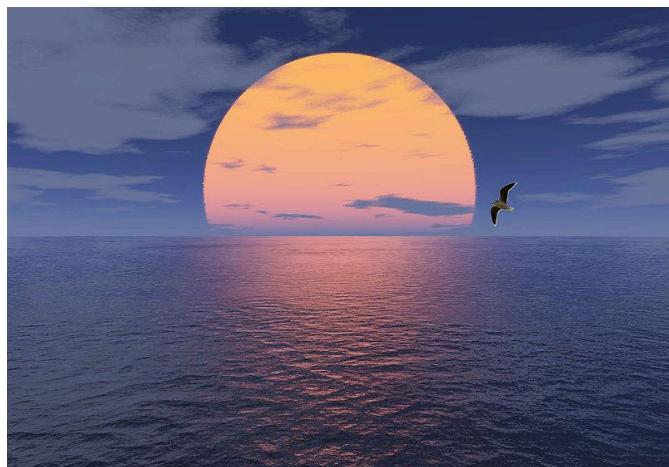


final blended image:



- At least one failure case, where the result is bad. Explain why it doesn't work. (10 pts)

background image:



object image:



pasted image:



final blended image:



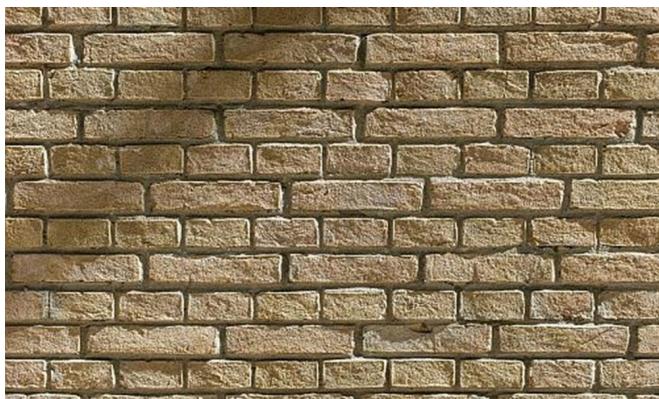
Poisson blending fails in this case because the lighting condition differs too much between the source and target image, which makes poisson blending produce odd results because it attempts to impose the gradient of the source onto the target, which might not be appropriate under different lighting conditions.

3. Mixed gradients

Using your own images (not sample images), include:

- At least one result with: (1) background and object images; (2) pasted image with source pixels directly copied onto target background region; (3) final blend result.

background image:



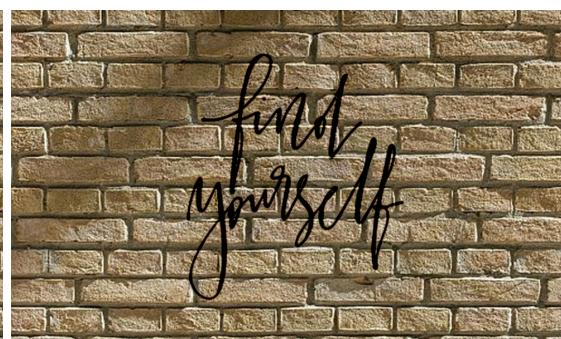
object image:



pasted image:



final blend result:



4. Quality of results / report

Nothing extra to include (scoring: 0=poor 5=average 10=great).

5. Color2Gray (B&W)

Include

- Color and grayscale of colorBlind8.png and colorBlind4.png, where grayscale is created by your code
- Color and grayscale of one natural image, where grayscale is created by your code
- Explain your method/constraints

6. Laplacian Pyramid Blending (B&W)

Include

- For at least one example, compare copy-paste, poisson, and laplacian pyramid blending. Include the object and background source images, and the blended results for each method. You can use different masks for different methods.

7. More gradient domain processing applications (B&W)

Include

- Show at least one example for each method that you implement. Explain the constraints used for each method. Something relatively complex like colorization is worth full points. Simpler applications like non-photorealistic rendering can also be worth full points if multiple variations are shown or clever methods used.

Acknowledgments / Attribution

List any sources for code or images from outside sources