EC522 Computational Optical Imaging
Homework No. 6

Shuyue Jia
BUID: U62343813

April 2024

1 Depth from defocus: a proof-of-concept demonstration

This problem will provide a simple illustration of "depth from defocus" based on a single measurement. Let us consider a two-layer "3D" scene, consisting of f_1 at depth 1 and f_2 at depth 2. Both objects are transparent so no occlusion effects need to be considered. A single 2D image g is taken, which is modeled as

$$g = f_1 * h_1 + f_2 * h_2 \tag{1}$$

where h_1 and h_2 are the PSFs at the two depths, respectively, and * denotes the 2D convolution.

The following Matlab files are provided.

a) The two-layer object, f_1 and f_2 are in the mat-file o1.mat and o2.mat, respectively.

b) The two PSFs h_1 and h_2 are provided in psf1.mat and psf2.mat, respectively.

Write Matlab scripts to complete the following questions. Submit both your scripts as well as the output results.

1.1 Simulate the output image, assuming no noise is present.

Although the exact Tikhonov-regularized solution for this problem is difficult to derive in a closed form, one can show that the following approximate solution is effective in estimating the object at each depth:

$$
\begin{aligned}
\mathbf{f}_{1,\mu_1} &= \mathbf{W}_{2\mathrm{D}}^{*} \operatorname{diag}\left(\frac{\widehat{\mathbf{h}_1}^{*}}{\left|\widehat{\mathbf{h}_1}\right|^{2} + \mu_1} \right) \mathbf{W}_{2\mathrm{D}} \mathbf{g}, \\
\mathbf{f}_{2,\mu_2} &= \mathbf{W}_{2\mathrm{D}}^{*} \operatorname{diag}\left(\frac{\widehat{\mathbf{h}_2}^{*}}{\left|\widehat{\mathbf{h}_2}\right|^{2} + \mu_2} \right) \mathbf{W}_{2\mathrm{D}} \mathbf{g},
\end{aligned}
\tag{2}
$$

where $\mathbf{W}_{2\mathrm{D}}$ is the DFT matrix, $\widehat{\mathbf{h}_1}$ and $\widehat{\mathbf{h}_2}$ are the corresponding transfer functions, and μ_1 and μ_2 are the regularization parameters.

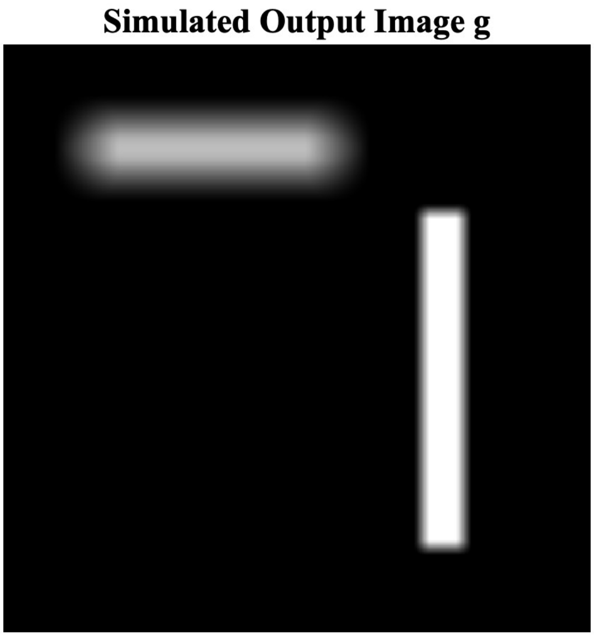

Figure 1: The simulated output image.

Matlab Codes:

```
% Load the provided data
load ('o1.mat', 'o1'); % Load object at depth 1
load ('o2.mat', 'o2'); % Load object at depth 2
load ('psf1.mat', 'psf1'); % Load PSF at depth 1
load ('psf2.mat', 'psf2'); % Load PSF at depth 2

```

```
% Simulate the output image
g = conv2(o1, psf1, 'same') + conv2(o2, psf2, 'same');

% Display the simulated output image
figure('Units', 'normalized', 'Position', [0.1, 0.1, 0.8, 1.0]);
imshow (g, []);
title('Simulated Output Image g', 'FontName', 'Times New Roman',
    'FontSize', 24, 'FontWeight', 'bold');
```

1.2 Choose the appropriate regularization parameters μ_1 and μ_2 and show the reconstructions of the two-layer object. Describe your observation.

Observations:

Observing the reconstructed objects based on different depths reveals several visual effects:

1. **Blur and Sharpness**: Objects reconstructed at different depths may exhibit varying degrees of blur or sharpness. This difference is particularly noticeable when comparing objects at different depths in the same scene. Objects at shallower depths may appear sharper with clearer edges, while objects at deeper depths may appear more blurred or less defined.

2. **Contrast and Intensity**: The contrast and intensity of the reconstructed objects can vary based on their depths and the chosen regularization parameters. Objects at shallower depths may exhibit higher contrast and intensity compared to objects at deeper depths, especially if stronger regularization is applied to the latter.

3. **Artifact and Noise Suppression**: Stronger regularization parameters tend to suppress noise and artifacts in the reconstructed objects, resulting in smoother surfaces and cleaner edges. However, excessive regularization may lead to loss of fine details and texture in the objects, particularly noticeable in regions with intricate patterns or high-frequency components.

4. **Depth Perception**: The reconstructed objects provide visual cues for depth perception in the scene. Objects at shallower depths may appear closer to the viewer and exhibit more pronounced features, while objects at deeper depths may appear farther away and exhibit more subdued features.

Choosing appropriate regularization parameters μ_1 and μ_2 involves finding a balance between reducing noise and preserving details in the reconstructed objects. Generally, smaller values of μ result in stronger regularization, which can suppress noise but may also blur the reconstructed objects. Conversely, larger values of μ result in weaker regularization, preserving more details but potentially amplifying noise.

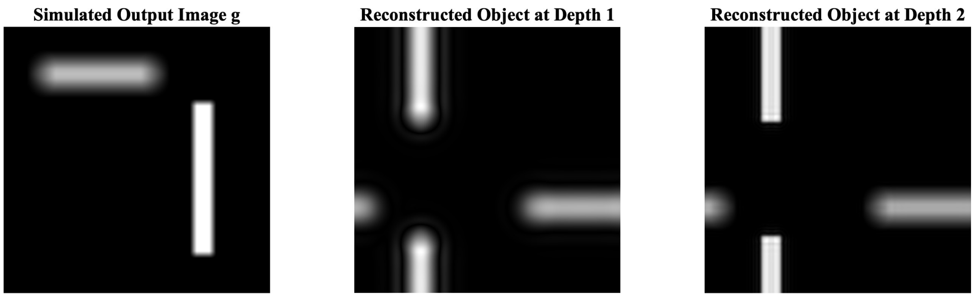

Figure 2: The simulated output image, the reconstructed image at depth 1, and the reconstructed image at depth 2.

Matlab Codes:

```
% Load the provided data
load ('o1.mat', 'o1'); % Load object at depth 1
load ('o2.mat', 'o2'); % Load object at depth 2
load ('psf1.mat', 'psf1'); % Load PSF at depth 1
load ('psf2.mat', 'psf2'); % Load PSF at depth 2

% Simulate the output image
g = conv2(o1, psf1, 'same') + conv2(o2, psf2, 'same');

% Display the simulated output image
figure('Units', 'normalized', 'Position', [0.1, 0.1, 0.8, 0.4]);
subplot(1, 3, 1);
imshow (g, []);
title('Simulated Output Image g', 'FontName', 'Times New Roman',
    'FontSize', 24, 'FontWeight', 'bold');

% Set regularization parameters
mu1 = 0.5;
mu2 = 0.05;

% Compute 2D FFT of PSFs
psf1 = fft2(psf1, size(g, 1), size(g, 2));
psf2 = fft2(psf2, size(g, 1), size(g, 2));

% Compute 2D FFT of the simulated output image
G = fft2(g);
```

```
% Compute reconstructed objects at each depth using regularization
o1_mu1 = conj(psf1) ./ (abs(psf1).^2 + mu1) .* G;
o2_mu2 = conj(psf2) ./ (abs(psf2).^2 + mu2) .* G;

% Inverse FFT to obtain reconstructed objects
o1_reconstructed = ifft2(o1_mu1);
o2_reconstructed = ifft2(o2_mu2);

% Display reconstructed objects
subplot(1, 3, 2);
imshow(abs(o1_reconstructed), []);
title('Reconstructed Object at Depth 1', 'FontName', 'Times New Roman', 'FontSize', 24, 'FontWeight', 'bold');

subplot(1, 3, 3);
imshow(abs(o2_reconstructed), []);
title('Reconstructed Object at Depth 2', 'FontName', 'Times New Roman', 'FontSize', 24, 'FontWeight', 'bold');
```

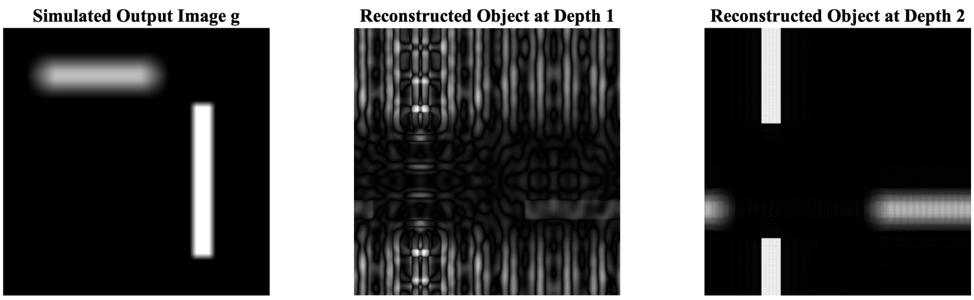

Figure 3: The simulated output image, the reconstructed image at depth 1 (with $\mu = 0.0001$), and the reconstructed image at depth 2 (with $\mu = 0.0001$).

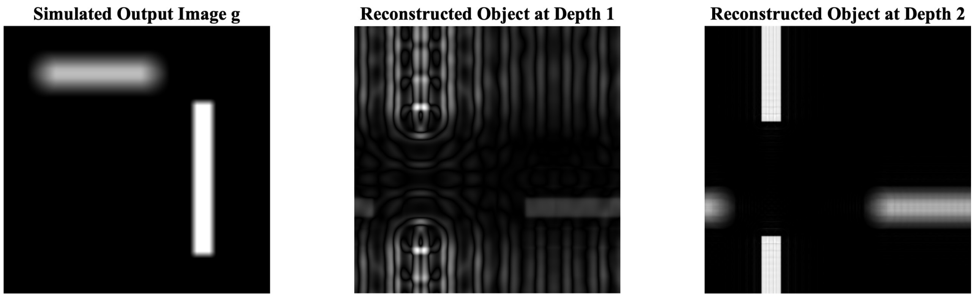

Figure 4: The simulated output image, the reconstructed image at depth 1 (with $\mu = 0.001$), and the reconstructed image at depth 2 (with $\mu = 0.001$).

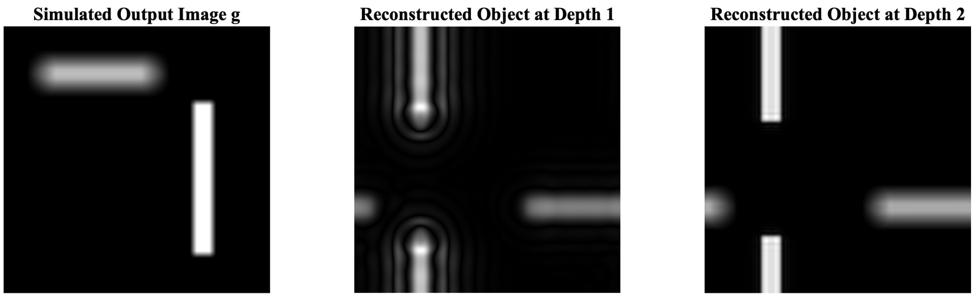

Figure 5: The simulated output image, the reconstructed image at depth 1 (with $\mu = 0.05$), and the reconstructed image at depth 2 (with $\mu = 0.05$).

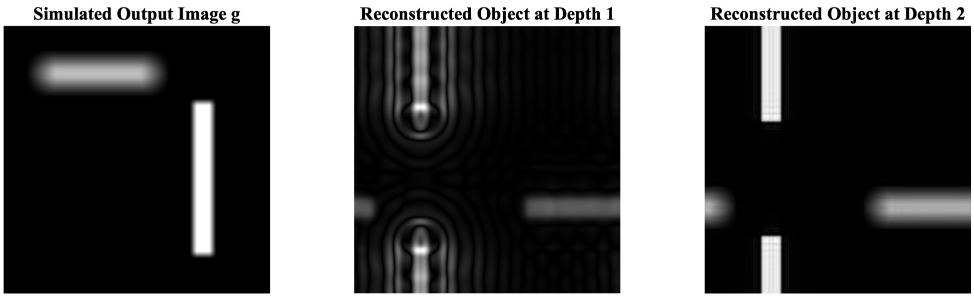

Figure 6: The simulated output image, the reconstructed image at depth 1 (with $\mu = 0.01$), and the reconstructed image at depth 2 (with $\mu = 0.01$).

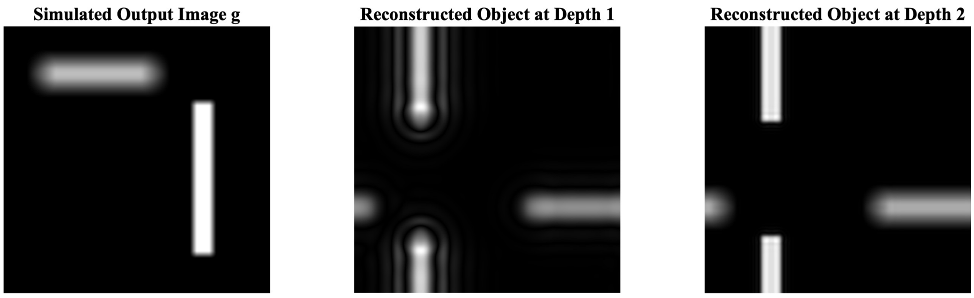

Figure 7: The simulated output image, the reconstructed image at depth 1 (with $\mu = 0.1$), and the reconstructed image at depth 2 (with $\mu = 0.1$).

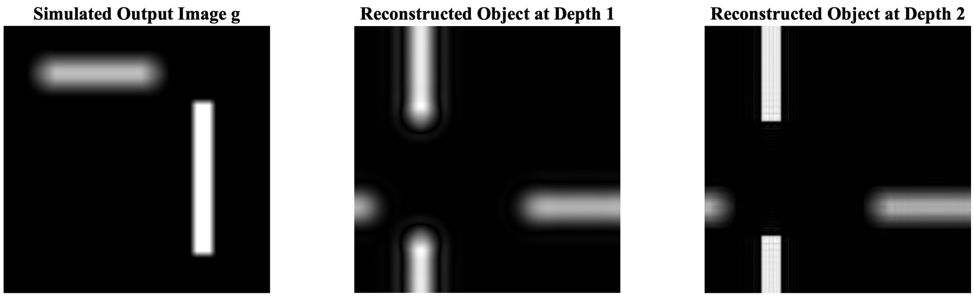

Figure 8: The simulated output image, the reconstructed image at depth 1 (with $\mu = 0.5$), and the reconstructed image at depth 2 (with $\mu = 0.05$).

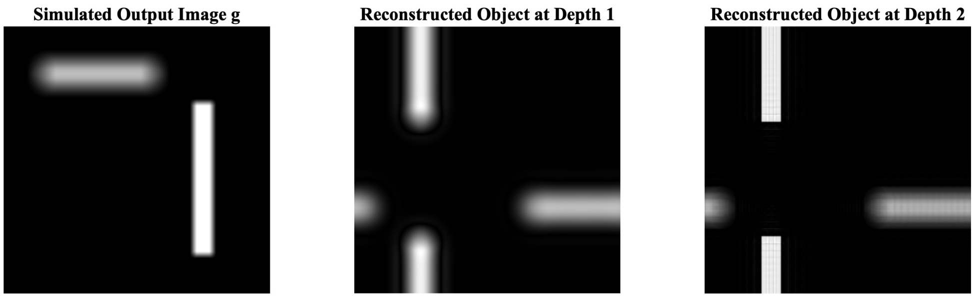

Figure 9: The simulated output image, the reconstructed image at depth 1 (with $\mu = 0.999999$), and the reconstructed image at depth 2 (with $\mu = 0.001$).