3.9 Exercises 175

1. Compute the Wiener ﬁltered version of all the noised images and compare them against your hand-tuned Gaussian-smoothed images.
2. (Optional) Do your image spectra have a lot of energy concentrated along the horizontal and vertical axes (*fx* = 0 and *fy* = 0)? Can you think of an explanation for this? Does rotating your image samples by 45*◦* move this energy to the diagonals? If not, could it be due to edge effects in the Fourier transform? Can you suggest some techniques for reducing such effects?

**Ex 3.17: Deblurring using Wiener ﬁltering** Use Wiener ﬁltering to deblur some images.

1. Modify the Wiener ﬁlter derivation (3.66–3.74) to incorporate blur (3.75).
2. Discuss the resulting Wiener ﬁlter in terms of its noise suppression and frequency boosting characteristics.
3. Assuming that the blur kernel is Gaussian and the image spectrum follows an inverse frequency law, compute the frequency response of the Wiener ﬁlter, and compare it to the unsharp mask.
4. Synthetically blur two of your sample images with Gaussian blur kernels of different radii, add noise, and then perform Wiener ﬁltering.
5. Repeat the above experiment with a “pillbox” (disc) blurring kernel, which is charac- teristic of a ﬁnite aperture lens (Section 2.2.3). Compare these results to Gaussian blur kernels (be sure to inspect your frequency plots).
6. It has been suggested that regular apertures are anathema to de-blurring because they introduce zeros in the sensed frequency spectrum (Veeraraghavan, Raskar, Agrawal *et al.* 2007). Show that this is indeed an issue if no prior model is assumed for the signal, i.e., *Ps−*1***l***1. If a reasonable power spectrum is assumed, is this still a problem (do we still get banding or ringing artifacts)?

**Ex 3.18: High-quality image resampling** Implement several of the low-pass ﬁlters pre- sented in Section 3.5.2 and also the discussion of the windowed sinc shown in Table 3.2 and Figure 3.29. Feel free to implement other ﬁlters (Wolberg 1990; Unser 1999).

Apply your ﬁlters to continuously resize an image, both magnifying (interpolating) and minifying (decimating) it; compare the resulting animations for several ﬁlters. Use both a synthetic chirp image (Figure 3.65a) and natural images with lots of high-frequency detail (Figure 3.65b-c).[27](#_bookmark0)

You may ﬁnd it helpful to write a simple visualization program that continuously plays the animations for two or more ﬁlters at once and that let you “blink” between different results.

Discuss the merits and deﬁciencies of each ﬁlter, as well as its tradeoff between speed and quality.

**Ex 3.19: Pyramids** Construct an image pyramid. The inputs should be a grayscale or color image, a separable ﬁlter kernel, and the number of desired levels. Implement at least the following kernels:

27 These particular images are available on the book’s Web site.