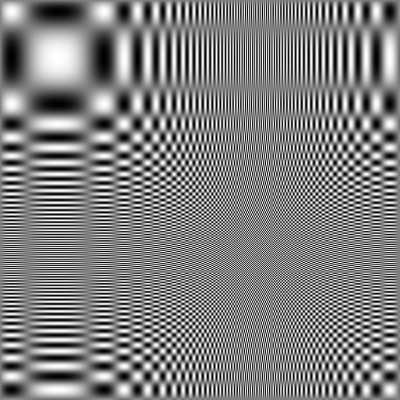
176 3 Image processing



1. (b) (c)

**Figure 3.65** Sample images for testing the quality of resampling algorithms: (a) a synthetic chirp; (b) and (c) some high-frequency images from the image compression community.

* + 2 *×* 2 block ﬁltering;
  + Burt and Adelson’s binomial kernel 1*/*16(1*,* 4*,* 6*,* 4*,* 1) (Burt and Adelson 1983a);
  + a high-quality seven- or nine-tap ﬁlter.

Compare the visual quality of the various decimation ﬁlters. Also, shift your input image by

1 to 4 pixels and compare the resulting decimated (quarter size) image sequence.

**Ex 3.20: Pyramid blending** Write a program that takes as input two color images and a binary mask image and produces the Laplacian pyramid blend of the two images.

1. Construct the Laplacian pyramid for each image.
2. Construct the Gaussian pyramid for the two mask images (the input image and its complement).
3. Multiply each Laplacian image by its corresponding mask and sum the images (see Figure 3.43).
4. Reconstruct the ﬁnal image from the blended Laplacian pyramid.

Generalize your algorithm to input *n* images and a label image with values 1 *...n* (the value 0 can be reserved for “no input”). Discuss whether the weighted summation stage (step 3) needs to keep track of the total weight for renormalization, or whether the math just works out. Use your algorithm either to blend two differently exposed image (to avoid under- and over-exposed regions) or to make a creative blend of two different scenes.

**Ex 3.21: Wavelet construction and applications** Implement one of the wavelet families described in Section 3.5.4 or by Simoncelli and Adelson (1990b), as well as the basic Lapla- cian pyramid (Exercise 3.19). Apply the resulting representations to one of the following two tasks: