3.9 Exercises 177

* **Compression:** Compute the entropy in each band for the different wavelet implemen- tations, assuming a given quantization level (say, 1/4 gray level, to keep the rounding error acceptable). Quantize the wavelet coefﬁcients and reconstruct the original im- ages. Which technique performs better? (See (Simoncelli and Adelson 1990b) or any of the multitude of wavelet compression papers for some typical results.)
* **Denoising.** After computing the wavelets, suppress small values using *coring*, i.e., set small values to zero using a piecewise linear or other *C*0 function. Compare the results of your denoising using different wavelet and pyramid representations.

**Ex 3.22: Parametric image warping** Write the code to do afﬁne and perspective image warps (optionally bilinear as well). Try a variety of interpolants and report on their visual quality. In particular, discuss the following:

* In a MIP-map, selecting only the coarser level adjacent to the computed fractional level will produce a blurrier image, while selecting the ﬁner level will lead to aliasing. Explain why this is so and discuss whether blending an aliased and a blurred image (tri-linear MIP-mapping) is a good idea.
* When the ratio of the horizontal and vertical resampling rates becomes very different (anisotropic), the MIP-map performs even worse. Suggest some approaches to reduce such problems.

**Ex 3.23: Local image warping** Open an image and deform its appearance in one of the following ways:

1. Click on a number of pixels and move (drag) them to new locations. Interpolate the resulting sparse displacement ﬁeld to obtain a dense motion ﬁeld (Sections 3.6.2 and 3.5.1).
2. Draw a number of lines in the image. Move the endpoints of the lines to specify their new positions and use the Beier–Neely interpolation algorithm (Beier and Neely 1992), discussed in Section 3.6.2, to get a dense motion ﬁeld.
3. Overlay a spline control grid and move one grid point at a time (optionally select the level of the deformation).
4. Have a dense per-pixel ﬂow ﬁeld and use a soft “paintbrush” to design a horizontal and vertical velocity ﬁeld.
5. (Optional): Prove whether the Beier–Neely warp does or does not reduce to a sparse point-based deformation as the line segments become shorter (reduce to points).

**Ex 3.24: Forward warping** Given a displacement ﬁeld from the previous exercise, write a forward warping algorithm:

1. Write a forward warper using splatting, either nearest neighbor or soft accumulation (Section 3.6.1).
2. Write a two-pass algorithm, which forward warps the displacement ﬁeld, ﬁlls in small holes, and then uses inverse warping (Shade, Gortler, He *et al.* 1998).