was illustrated in Figures 5 and 23. When the frequency domain characteristics can be represented in an analytic form, then this can lead to relatively straightforward implementations of $H(\Omega,\Psi)$. Possible candidates include the lowpass filters "Airy" and "Exponential Decay" found in Table 4–T.5 and Table 4–T.8, respectively.

9.4.2 Non-Linear Filters

A variety of smoothing filters have been developed that are not linear. While they cannot, in general, be submitted to Fourier analysis, their properties and domains of application have been studied extensively.

• *Median filter* – The median statistic was described in Section 3.5.2. A median filter is based upon moving a window over an image (as in a convolution) and computing the output pixel as the median value of the brightnesses within the input window. If the window is $J \times K$ in size we can order the $J{\bullet}K$ pixels in brightness value from smallest to largest. If $J{\bullet}K$ is odd then the median will be the $(J{\bullet}K+1)/2$ entry in the list of ordered brightnesses. Note that the value selected will be exactly equal to one of the existing brightnesses so that no roundoff error will be involved if we want to work exclusively with integer brightness values. The algorithm as it is described above has a generic complexity per pixel of $O(J{\bullet}K{\bullet}\log(J{\bullet}K))$. Fortunately, a fast algorithm (due to Huang et al. [23]) exists that reduces the complexity to $O(K)$ assuming $J \geq K$.

A useful variation on the theme of the median filter is the *percentile filter*. Here the center pixel in the window is replaced not by the 50% (median) brightness value but rather by the p% brightness value where p% ranges from 0% (the *minimum filter*) to 100% (the *maximum filter*). Values other then $(p=50)$% do not, in general, correspond to smoothing filters.

• *Kuwahara filter* – Edges play an important role in our perception of images (see Figure 15) as well as in the analysis of images. As such it is important to be able to smooth images without disturbing the sharpness and, if possible, the position of edges. A filter that accomplishes this goal is termed an *edge-preserving filter* and one particular example is the Kuwahara filter [27]. Although this filter can be implemented for a variety of different window shapes, the algorithm will be described for a square window of size $J = K = 4L + 1$ where L is an integer. The window is partitioned into four regions as shown in Figure 29.

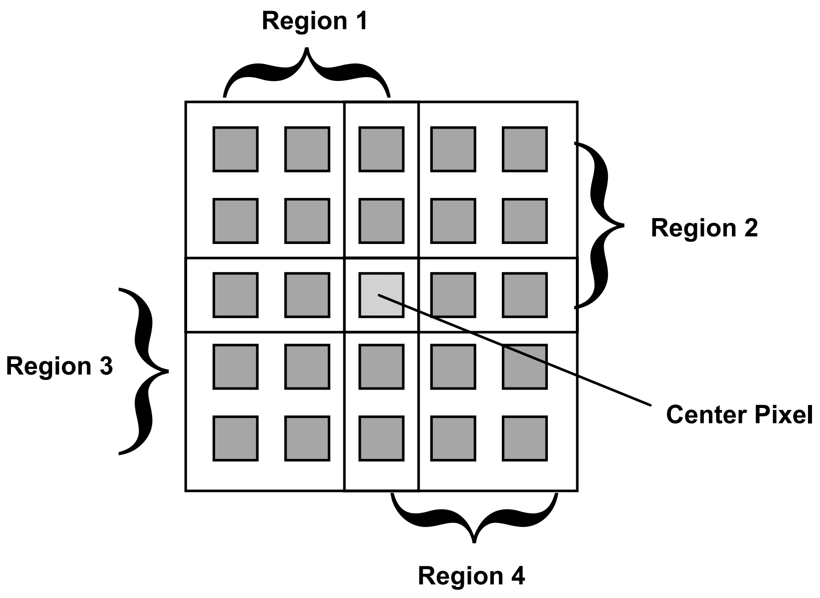

Figure 29: Four, square regions defined for the Kuwahara filter. In this example *L*=1 and thus *J*=*K*=5. Each region is [(*J*+1)/2] × [(*K*+1)/2].

In each of the four regions (*i*=1,2,3,4), the mean brightness, m_i in eq. (34), and the *variance*$_i$, s_i^2 in eq. (36), are measured. The output value of the center pixel in the window is the mean value of that region that has the smallest variance.

9.4.3 Summary of Smoothing Algorithms

The following table summarizes the various properties of the smoothing algorithms presented above. The filter size is assumed to be bounded by a rectangle of $J \times K$ where, without loss of generality, $J \geq K$. The image size is $N \times N$.

Algorithm	*Domain*	*Type*	*Support*	*Separable / Incremental*	*Complexity/pixel*
Uniform	Space	Linear	Square	Y / Y	*O*(*constant*)
Uniform	Space	Linear	Circular	N / Y	*O*(*K*)
Triangle	Space	Linear	Square	Y / N	*O*(*constant*) [a]
Triangle	Space	Linear	Circular	N / N	*O*(*K*) [a]
Gaussian	Space	Linear	∞ [a]	Y / N	*O*(*constant*) [a]
Median	Space	Non-Linear	Square	N / Y	*O*(*K*) [a]
Kuwahara	Space	Non-Linear	Square [a]	N / N	*O*(*J*• *K*)
Other	Frequency	Linear	—	— / —	*O*(log*N*)

Table 13: Characteristics of smoothing filters. [a]See text for additional explanation.

Examples of the effect of various smoothing algorithms are shown in Figure 30.