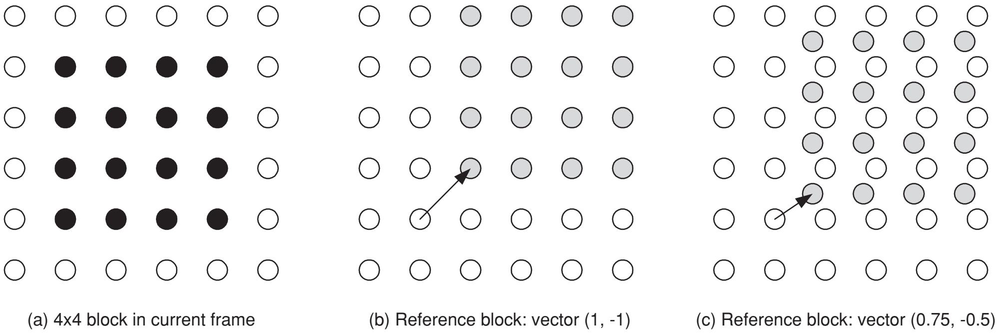

Figure 6.18 Example of integer and sub-pixel prediction

6.4.2 Interpolating reference pictures

Each partition in an inter-coded macroblock is predicted from an area of the same size in a reference picture. The offset between the two areas, the motion vector, has $^1/_4$-pixel resolution for the luma component and 1/8-pixel resolution for the chroma components. The luma and chroma samples at sub-pixel positions do not exist in the reference picture and so it is necessary to create them using interpolation from nearby image samples [ii]. In Figure 6.18, a 4×4 block in the current frame (a) is predicted from a neighbouring region of the reference picture. If the horizontal and vertical components of the motion vector are integers (b), the relevant samples in the reference block actually exist, shown as grey dots. If one or both vector components are fractional values (c), the prediction samples, shown as grey dots, are generated by interpolation between adjacent samples in the reference frame, shown as white dots.

Example

Figure 6.19 shows a small region of the current frame, containing a vertical object on a white background. We want to find a match for the 4×4 block shown in Figure 6.20, marked with small stars. The corresponding region of the reference frame is shown in Figure 6.21. Note that the vertical object is **not** aligned with the sample positions in Figure 6.21; the object has moved by a non-integer number of pixels between frames. Without any interpolation, it is not possible to find a good match in the reference region. The best match will be something like the one shown in Figure 6.22. We may be able to do better by interpolating between the samples of the reference frame to generate half-pixel positions (Figure 6.23). Searching the interpolated reference frame gives a better match (Figure 6.24), best match indicated by stars. The match is not perfect – the luminance levels are not quite the same as those of the original 4×4 block (Figure 6.20) – but the prediction is better than the integer-pixel match. A better prediction gives a smaller residual and hence better compression. In general, 'finer' interpolation, i.e. increasing the number of interpolation stages, reduces the residual, at the expense of increased computation and more bits required to send motion vectors.

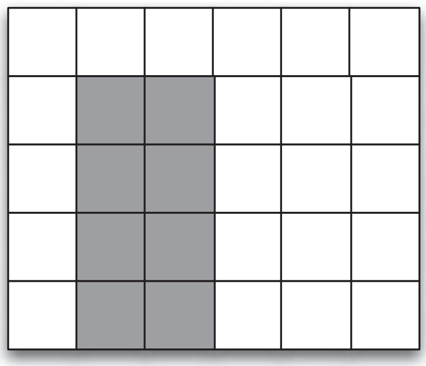

Figure 6.19 Current region

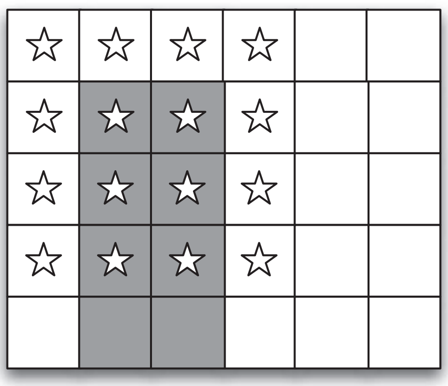

Figure 6.20 4×4 block to be predicted

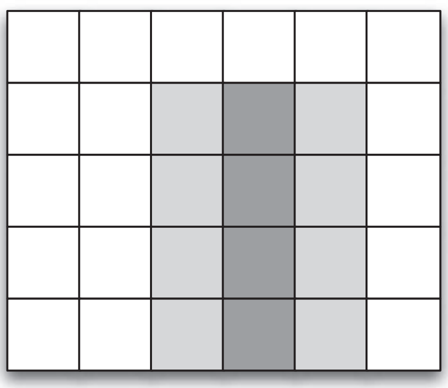

Figure 6.21 Reference region

Figure 6.22 Prediction from integer samples

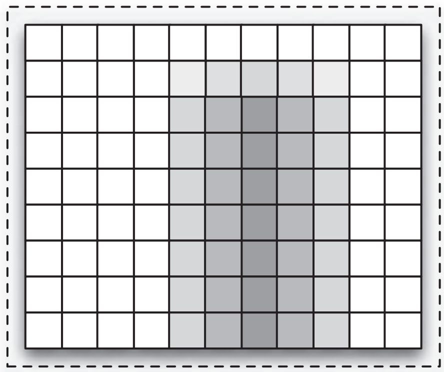

Figure 6.23 Reference region, half-pixel interpolated

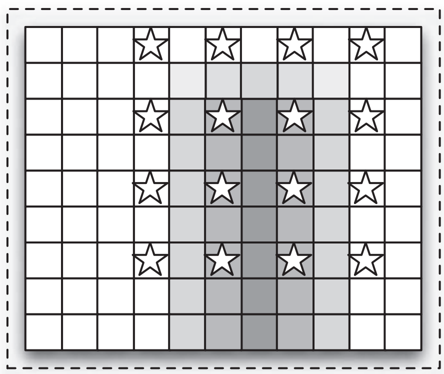

Figure 6.24 Prediction from interpolated samples

6.4.2.1 Generating interpolated sub-pixel samples

Luma component

The half-pel samples in the luma component of the reference picture are generated first, Figure 6.25, grey markers. Each half-pel sample that is adjacent to two integer samples, e.g. b, h, m, s in Figure 6.25, is interpolated from integer-pel samples using a 6 tap Finite Impulse Response (FIR) filter with weights (1/32, −5/32, 5/8, 5/8, −5/32, 1/32). For example, half-pel sample **b** is calculated from the 6 horizontal integer samples E, F, G, H, I and J using a process equivalent to:

$$\mathbf{b} = \text{round}((E - 5F + 20G + 20H - 5I + J)/32)$$

Similarly, **h** is interpolated by filtering A, C, G, M, R and T. Once all of the samples adjacent to integer samples have been calculated, the remaining half-pel positions are calculated by interpolating between six horizontal or vertical half-pel samples from the first set of operations. For example, **j** is generated by filtering cc, dd, h, m, ee and ff. Note that the result is the same whether j is interpolated horizontally or vertically. The 6-tap interpolation filter is relatively complex but produces an accurate fit to the integer-sample data and hence good motion compensation performance.

Once all the half-pixel samples are available, the quarter-pixel positions are produced by linear interpolation Figure 6.26. Quarter-pixel positions with two horizontally or vertically adjacent half- or integer-pixel samples, e.g. **a**, **c**, **i**, **k** and **d**, **f**, **n**, **q** in Figure 6.26, are linearly interpolated between these adjacent samples, for example:

$$\mathbf{a} = \text{round}((G + b)/2)$$

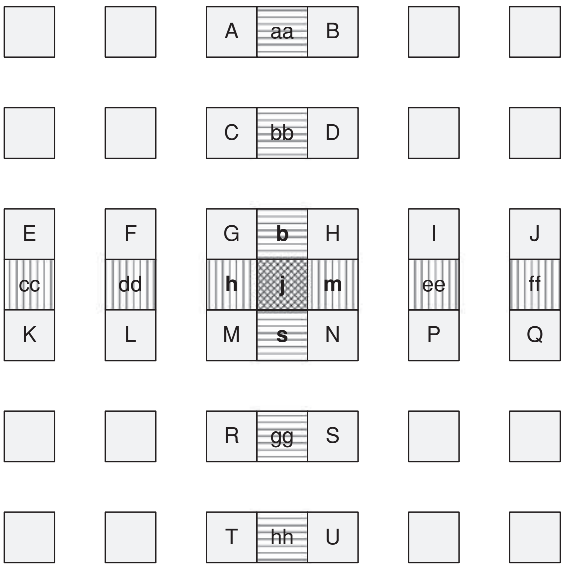

Figure 6.25 Interpolation of luma half-pel positions

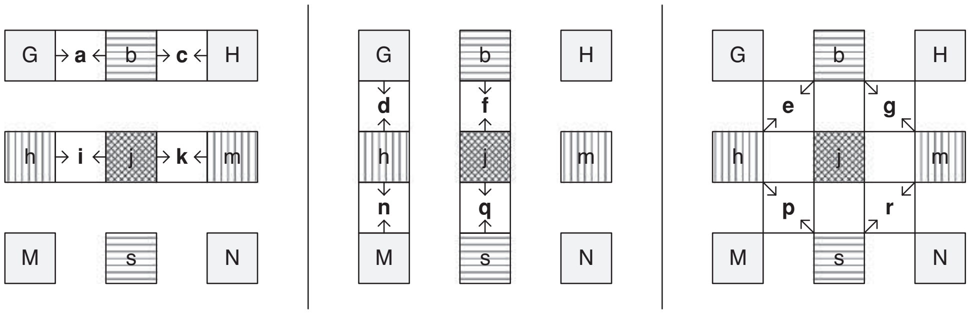

Figure 6.26 Interpolation of luma quarter-pel positions

The remaining quarter-pixel positions, **e**, **g**, **p** and **r** in the figure, are linearly interpolated between a pair of diagonally opposite half-pixel samples. For example, **e** is interpolated between b and h.

Figure 6.27 shows the result of interpolating the reference region shown in Figure 3.16 with quarter-pixel resolution.

Chroma components

Quarter-pel resolution motion vectors in the luma component require eighth-pel resolution vectors in the chroma components, assuming 4:2:0 sampling. Interpolated samples are

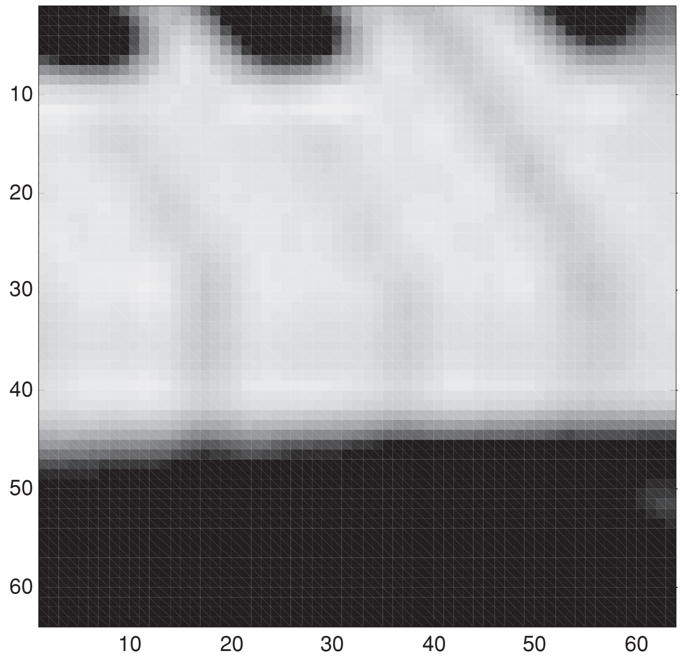

Figure 6.27 Luma region interpolated to quarter-pel positions