CHAPTER 4

The outer limits: learning about edges and orientation

> *The Disc, being flat, has no real horizon. Any adventurous sailors who got funny ideas from staring at eggs and oranges for too long and set out for the antipodes soon learned that the reason why distant ships sometimes looked as though they were disappearing over the edge of the world was that they* were *disappearing over the edge of the world.*
>
> (Pratchett, 1986)

The previous chapter showed how elementary sensitivity to motion is sufficient to gather segmentations of objects in the robot's vicinity, with some support from the robot's behavior to evoke easily processed scenarios. Once this data is coming in, there is a lot that can be learned from it. One reasonable use of the data would be to learn about the appearance of specific objects, and the next chapter (Chapter 5) will address that. But even before that, it is also possible to simply learn something about the appearance of boundaries, since the robot now has a collection of such boundaries side by side with their visual appearance. In particular, this allows an orientation detector to be trained on automatically annotated data. Orientation information is present in images at all scales. It is typically detected using quadrature filters applied at many locations and scales (Freeman, 1992), an approach developed to be independent of contrast polarity and to act equally well on edges and lines. With the data the robot collects, the opportunity arises to take a complementary, empirical approach, where the appearance of edges is learned from experience rather than derived theoretically. The main challenge is whether the appearance of edges can be sampled densely enough to get good coverage on a reasonable timescale. The answer to this is shown to be yes, primarily because orientation information is quite robust to pixel-level transformations. It turns out that a useful orientation filter can be constructed a a simple interpolating look-up table, mapping from a very small window size (4×4 pixels) directly to orientation. This allows for extremely rapid access to orientation information right down at the finest scale visible.

The contribution of this work is to demonstrate that orientation detection is amenable to empirical treatment, and that it can be performed at a very high speed. This work is critical to a real-time implementation of the object recognition method that will be proposed in Chapter 5.

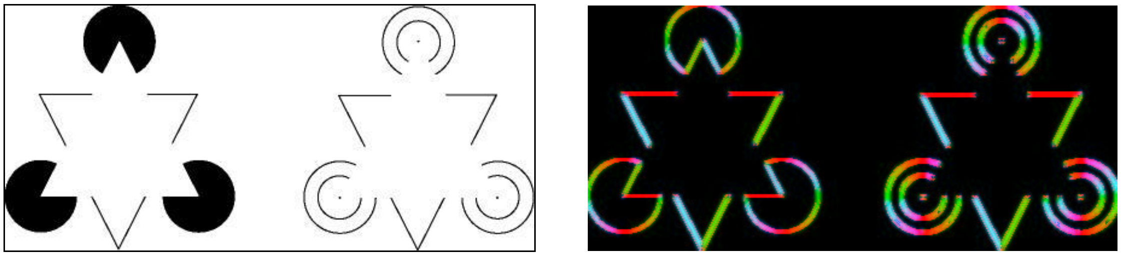

Figure 4-1: The goal of orientation detection is to take an image such as the one shown on the left here, and annotate every point in it with a direction, if there is a well-defined orientation that can be associated with it. For example, on the right is a color-coded orientation map corresponding to the first image, where all horizontal lines and edges are colored red, etc. This map is produced by the methods developed in this chapter. It shows only orientation that is clear from local information – the “illusory contours” present in the Kanizsa triangles are not detected.

4.1 What is orientation?

Natural images are full of discontinuities and local changes. This anisotropy can be used to associate directions with regions of the image. These directions are potentially more robust to image-wide transformations than the individual pixels upon which they are based. The most obvious example is a luminance edge, where there is a discontinuity between a dark and light region. The direction associated with this edge remains unchanged even if overall illumination on the regions change their appearance dramatically. Contours of constant luminance on a shaded surface behave somewhat like edges also, with luminance change being minimal parallel to the contour and maximal when measured perpendicular to them. For such directional changes in luminance, or any other property, it is natural to associate a direction or *orientation* in which change is minimal. In this chapter, we will be concerned with the orientation associated with edges in luminance at the finest scale available. This is certainly not all that is to be said about orientation (see, for example, Figure 4-1). But it is a useful case, particularly for object localization and recognition. Orientation detection will prove key to achieving orientation and scale invariance in these tasks.

Orientation is associated with neighborhoods rather than individual points in an image, and so is inherently scale dependent. At very fine scales, relatively few pixels are available from which to judge orientation. Lines and edges at such scales are extremely pixelated and rough. Orientation filters derived from analytic considerations, with parameters chosen assuming smooth, ideal straight lines or edges (for example, Chen et al. (2000)) are more suited to larger neighborhoods with more redundant information. For fine scales, an empirical approach seems more promising, particularly given that when the number of pixels involved is low, it is practical to sample the space of all possible appearances of these pixels quite densely. At very fine scales, the interpretation of an image patch could hinge on a relatively small number of pixels. Noise sensitivity becomes a critical issue. But even beyond that, it seems that the assignment of labels to image patches is likely to be quite a non-linear process.

4.2 Approaches to orientation detection

Most methods for detecting local orientation fall into one of two categories. Gradient-based approaches such as that of Kass and Witkin (1987) are relatively direct, and operate by applying spa-

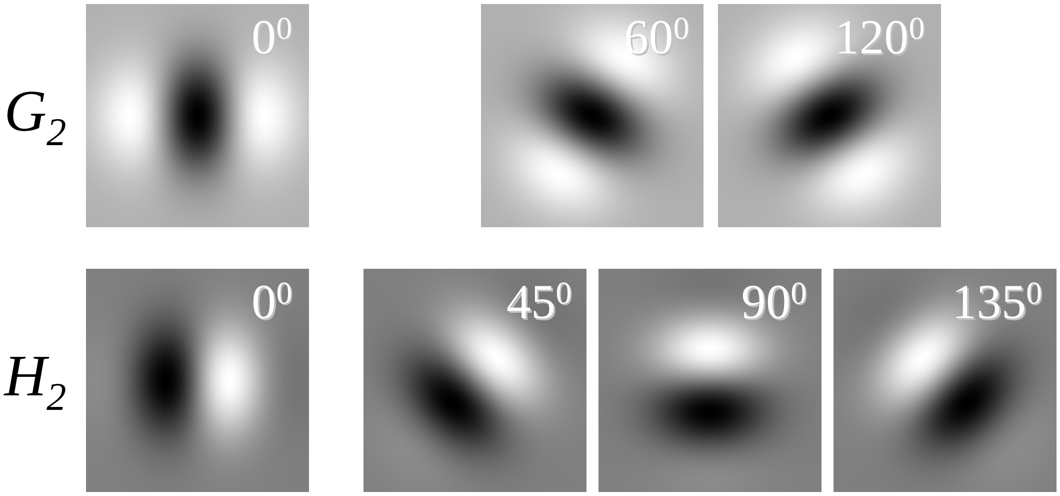

Figure 4-2: An example of a steerable filter, following Freeman and Adelson (1991). G_2 is the second derivative of a Gaussian, and H_2 is an approximation to its Hilbert transform. These two filters are said to be in quadrature. From its even form, G_2 will respond well to vertical lines. H_2 is odd, and will respond well to vertically oriented step edges. The theory associated with steerable filters shows that the response of an image with a small set of basis filters at discrete angles, as shown here, can be used to compute the response of one of the filter rotated to any angle. Orientation detection then becomes applying these filters and computing the angles that would give maximum response.

tial derivatives to the output of an isotropic edge-detecting filter (such as a Laplacian or difference of Gaussians). A different approach often used is to examine the response of each neighborhood in the image to a set of oriented filters, chosen so that some of them respond to edges ('cosine-phase'), and some respond to bars ('sine-phase'), analogous to the receptive fields found by Hubel and Wiesel in the visual cortex of cats (Hubel and Wiesel, 1962). The filter set may be overcomplete and non-orthogonal since image reconstruction is not the goal. Figure 4-2 shows an example of a possible filter set. If the filter is chosen carefully, then it need only be replicated at a discrete number of orientations, and the response of the image to any other orientation computed from the response to those few. Such filters are said to be steerable (Freeman and Adelson, 1991). Orientation is computed by finding the orientation that maximizes the response of the image to the filter (here the cosine-phase and sine-phase filters can be thought of as the real and imaginary components of a single quadrature filter).

4.3 Empirical orientation detection

Poking allows the robot to build up a reference "catalog" of the manifold appearances real edges can take on. At fine scales, with relatively few pixels, we can hope to explore the space of possible appearances of such a neighborhood quite exhaustively, and collect empirical data on how appearance relates to orientation. This chapter is basically an exploration of how edges in "natural" images appear when viewed through an extremely small window (4 by 4 pixels). This window size is chosen to be large enough to actually allow orientation to be well-defined, but small enough for the complete range of possible appearances to be easily characterized and visualized. Even at this scale, manual data collection and labelling would be extremely tedious, so it is very advantageous

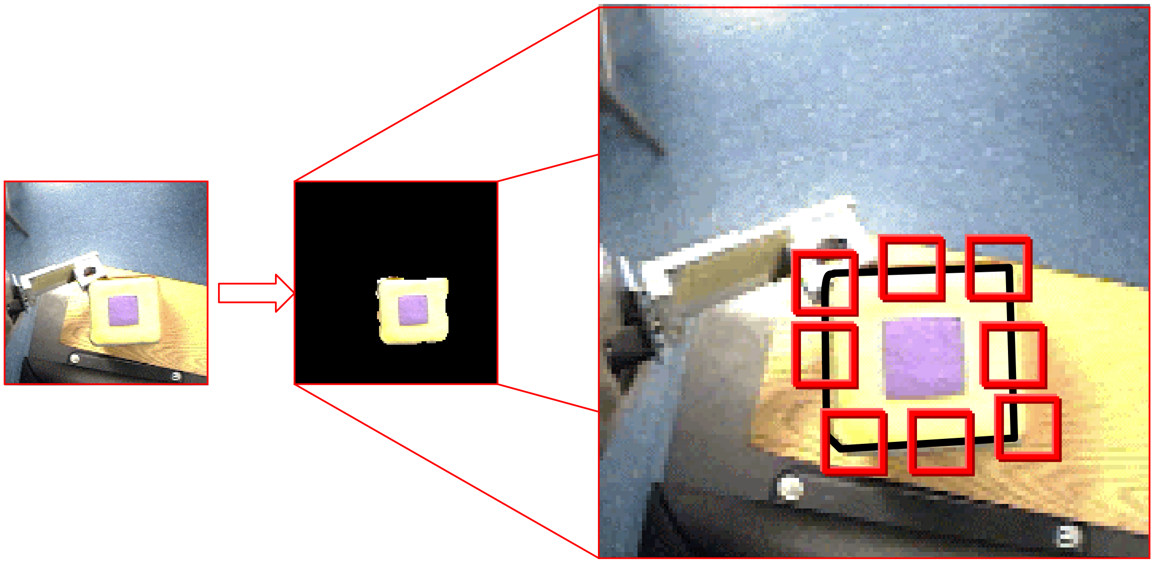

Figure 4-3: Sampling the appearance of edges at an object boundary. The object is detected and segmented as described in Chapter 3. Its boundary is sampled, and quantized window appearance is stored along with the actual angle of the boundary at that point.

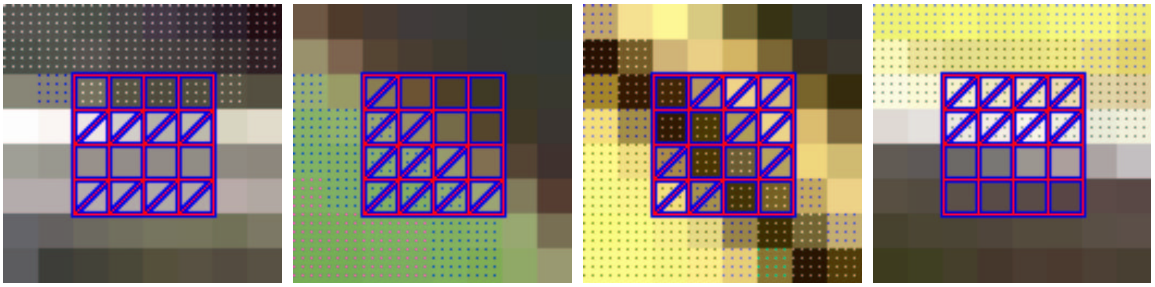

Figure 4-4: Some examples of boundary samples. Dotted pixels belong to a segmented object. The four-by-four grid overlaid on the boundary shows the result of thresholding.

to have a robot to take care of this. The robot automatically compiles a database of the appearance of oriented features using the poking behavior.

Oriented features were extracted by sampling image patches along object boundaries, which were in turn determined using active segmentation. The resulting "catalog" of edge appearances proved remarkably diverse, although the most frequent appearances were indeed the "ideal" straight, noise-free edge (Section 4.3). Finally, it is a simple matter to take this catalog of appearances and use it as a fast memory-based image processing filter (Section 4.3).

The details of the robot's behavior are as described in Chapter 3, and are briefly reviewed here. A robot equipped with an arm and an active vision head was given a simple "poking" behavior, whereby it selected objects in its environment, and tapped them lightly while fixating them. As described in Chapter 3, the motion signature generated by the impact of the arm with a rigid object greatly simplifies segmenting that object from its background, and obtaining a reasonable estimate of its boundary. Once this boundary is known, the appearance of the visual edge between the object and the background can be sampled along it (see Figure 4-3). These samples are labelled with the orientation of the boundary in their neighborhood (estimated using a simple discrete deriva-

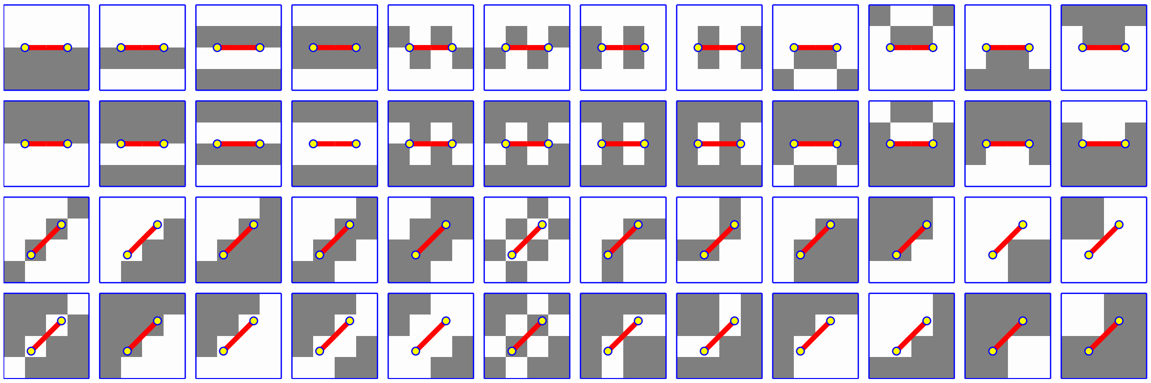

Figure 4-5: Edges have diverse appearances. This figure shows the orientations assigned to a test suite prepared by hand. Each 4×4 grid is a single test edge patch, and the dark line centered in the grid is the orientation that patch was observed to have in the training data. The oriented features represented include edges, thin lines, thick lines, zig-zags, corners etc.

tive of position along the boundary). The samples are assumed to contain two components that are distinguished by their luminance. The pixels of each sample are quantized into binary values corresponding to above average and below average luminance. Quantization is necessary to keep the space of possible appearances from exploding in size. The binary quantization gives a very manageable 65536 possible appearances. About 500 object boundaries were recorded and sampled. 49616 of the possible appearances (76%) were in fact observed; the remaining 24% were all within a Hamming distance of one of an observed appearance. The orientation of these unobserved appearances were interpolated from their immediate neighbors in Hamming space. If the same appearance was observed multiple times, the orientations associated with these observations are averaged using a double-angle representation (Granlund, 1978).

It is a straightforward matter to use the data we have collected to filter an image for fine scale orientation features. A 4×4 window is moved across the image, sampling it as described earlier in Section 4.3. Each sample is used as an index into a table mapping appearance to orientation.

4.4 Results

Figure 4-5 shows that although the data collection procedure operates on views of simple physical edges, the appearance of these edges can be quite complex. Nevertheless, the most common appearances observed are ideal, noise-free edges, as Figure 4-6 shows. The first four appearances shown (top row, left) make up 7.6% of all observed appearances by themselves. Line-like edges are less common, but do occur, which means that it is perfectly possible for the surfaces on either side of an edge to be more like each other than they are like the edge itself. This was completely serendipitous – it was anticipated that obtaining and automatically labelling such examples would be very difficult.

Figure 4-5 shows the most frequently occurring image appearances with a particular orientation. Here it is clearer that the most frequent patches are generally “ideal” forms of the edges, followed by very many variations on those themes with distracting noise. Amidst the edge-like patterns are examples of a line with single-pixel thickness, and a pair of such lines running parallel. It is encouraging that examples of such appearances can be collected without difficulty and united with more classical edge patches of the same orientation.

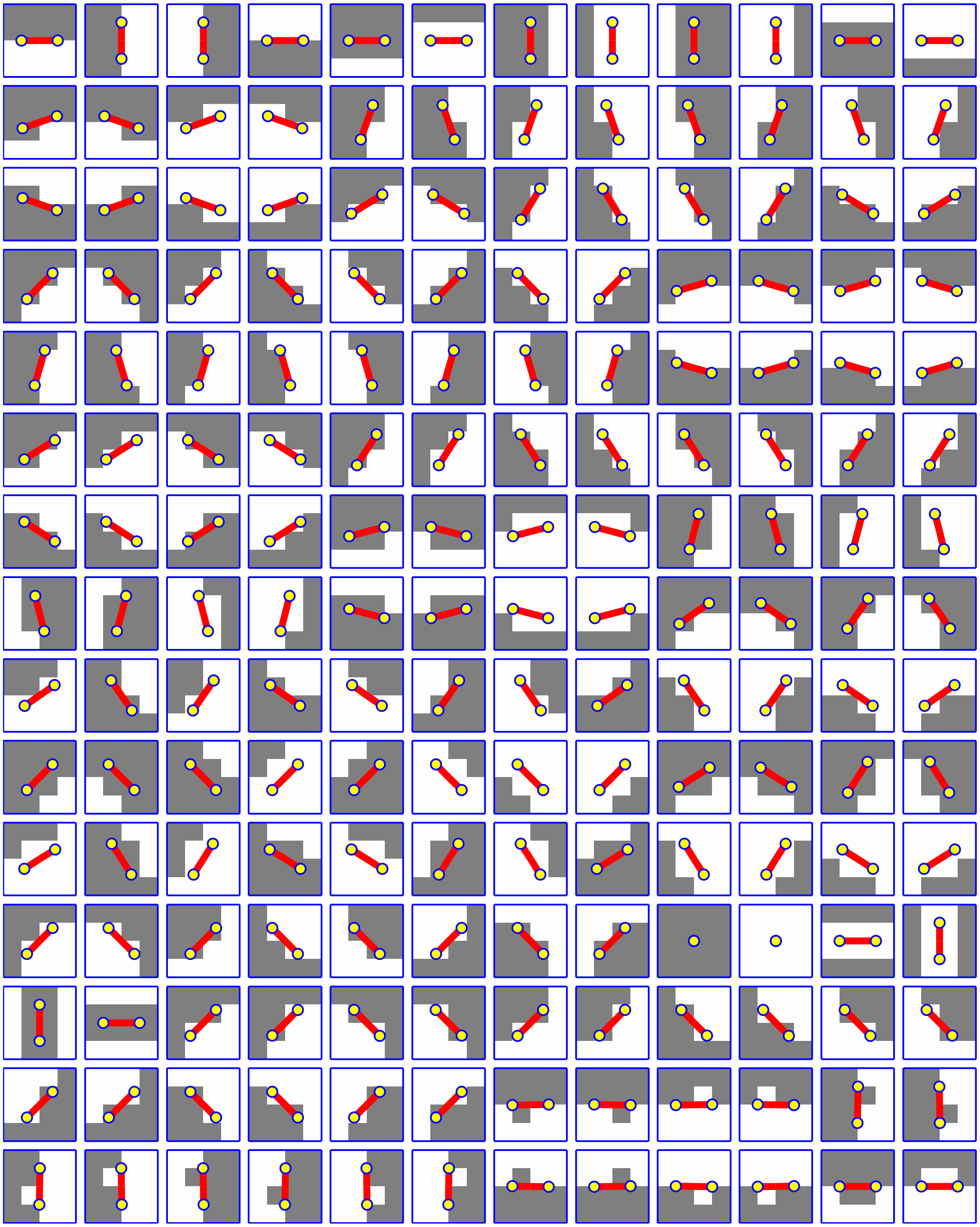

Figure 4-6: The most frequently observed edge appearances. All patches observed are replicated for all 90° rotations, mirror flips, and inversion of foreground/background. The most frequent (top) are simple straight edges. The line in the center of each patch shows the orientation associated with that patch. After the straight edges, the completely empty patch is common (produced in saturated regions), followed by a tube-like feature (third-last row) where the boundary is visually distinct to either side of the edge. This is followed corner-like features and many thousands of variations on the themes already seen.

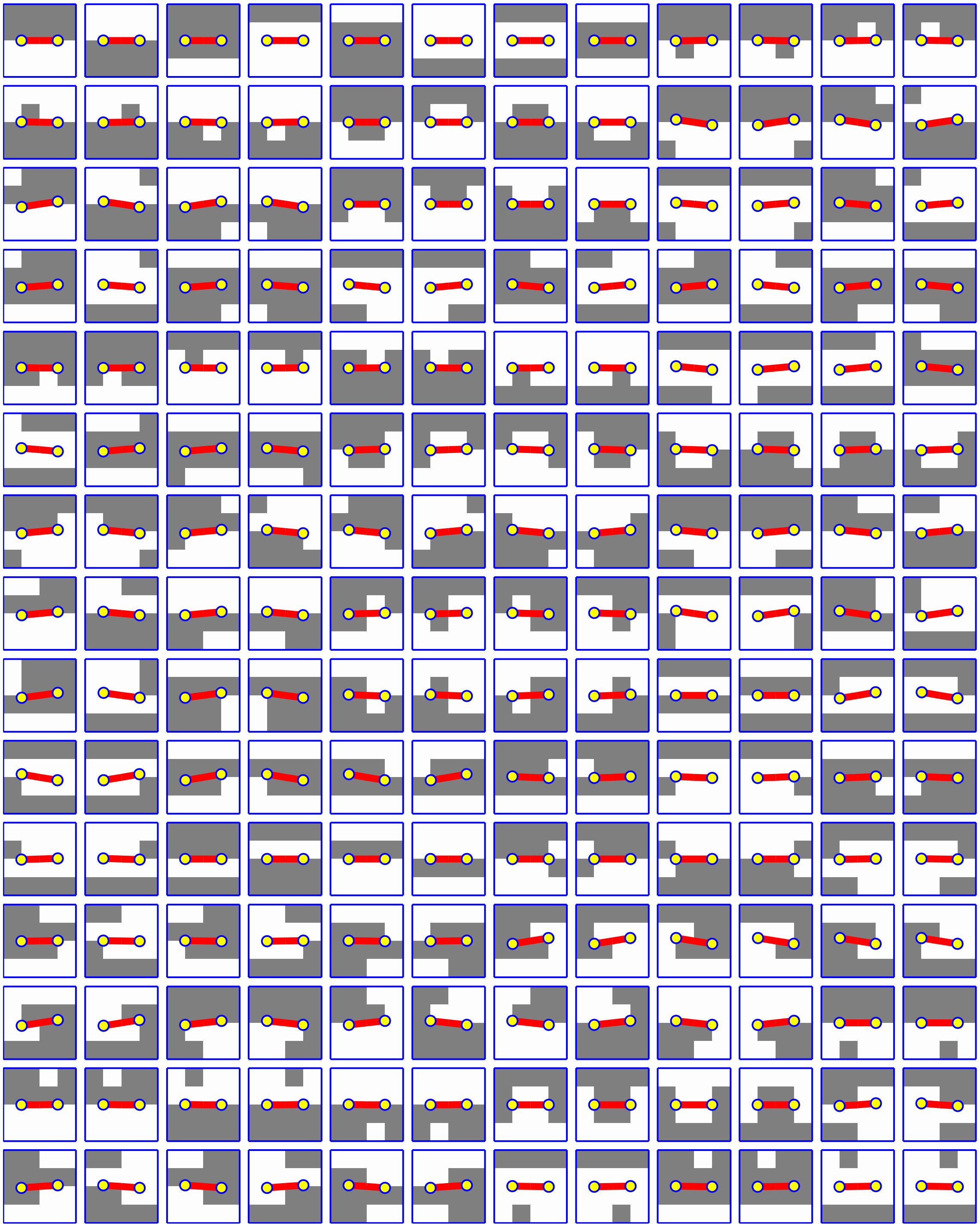

Figure 4-7: The most frequently observed appearances whose orientation is within 5° of the horizontal. There is a clear orientation assigned to many patches that deviate a great deal from “ideal” edges/lines, showing a robustness that is examined systematically in Figure 4-5.

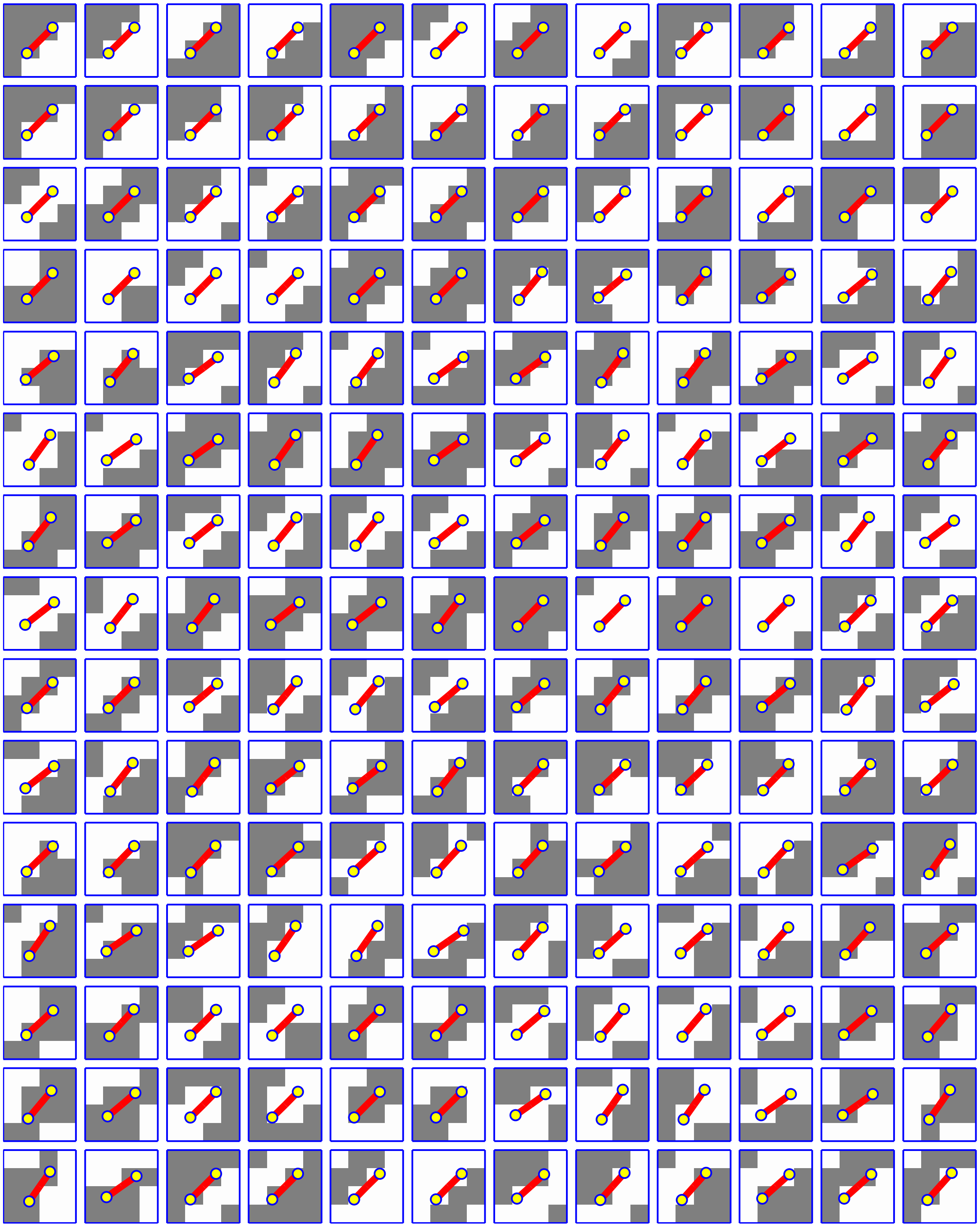

Figure 4-8: The most frequently observed appearances whose orientation is in the range $40 - 50°$. Again, there is a clear orientation assigned to many patches that deviate a great deal from "ideal" edges/lines.

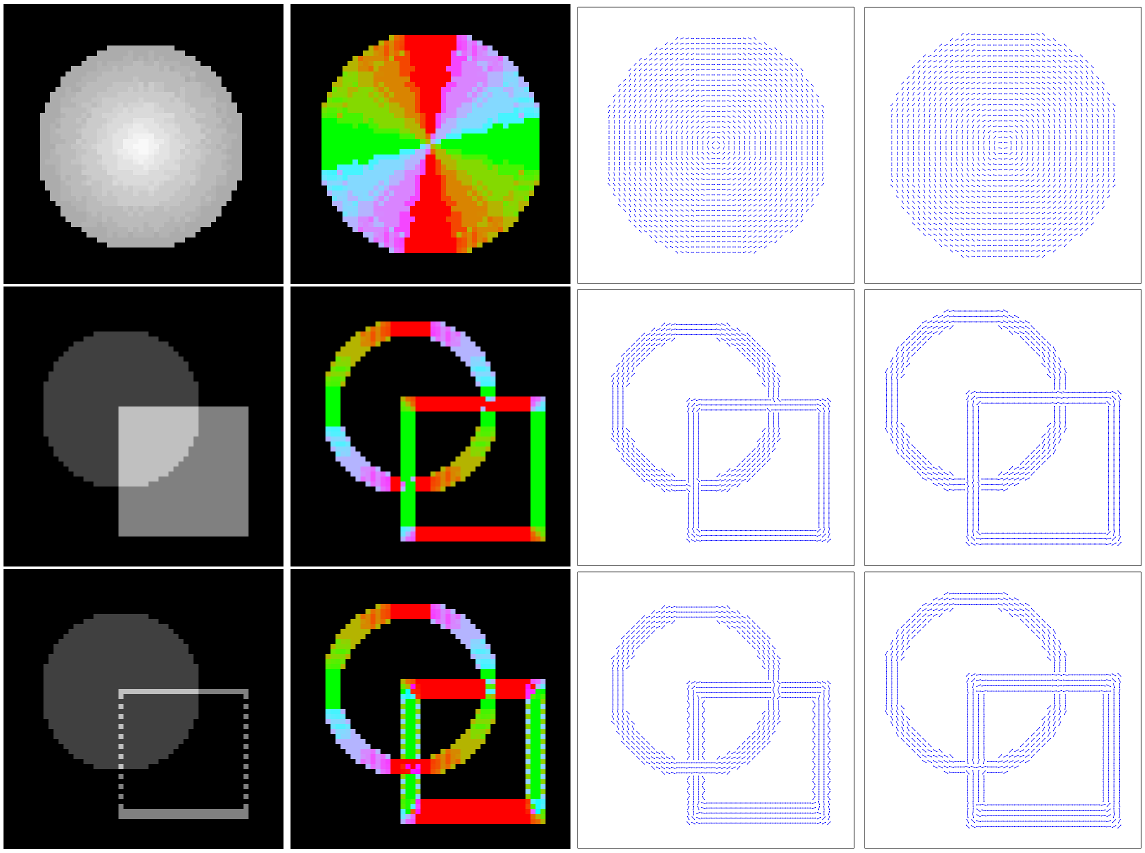

Figure 4-9: The orientation filter applied to some synthetic test images (on left), modeled after an example in (Freeman and Adelson, 1991). The second column shows the output of the orientation filter, color coded by angle (if viewed in color). The third column shows the same information in vector form. The fourth column shows the orientation determined using steerable quadrature filters Folsom and Pinter (1998) applied on the same scale. The results are remarkably similar, but the quadrature filters are much more computationally expensive to apply.

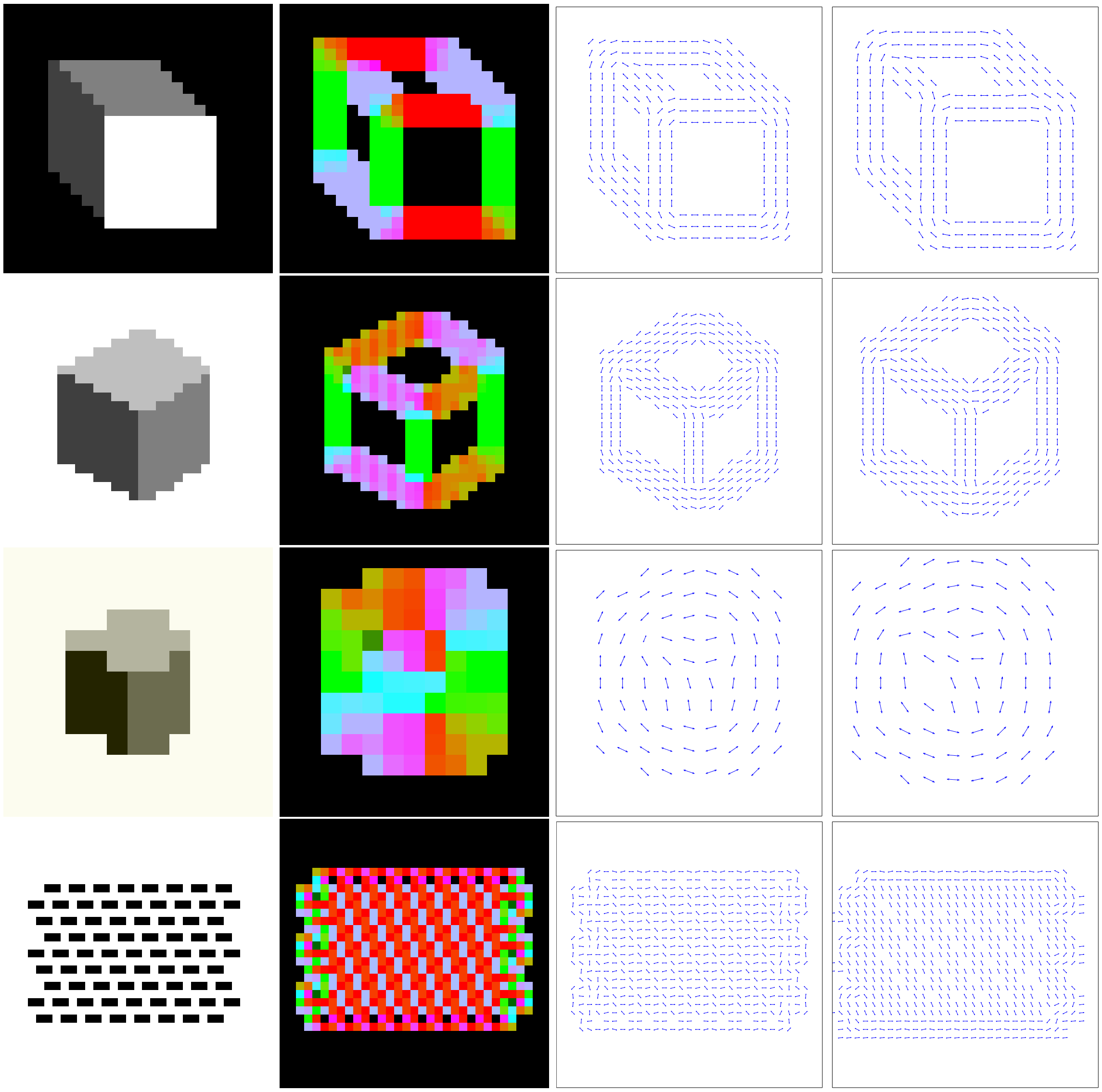

Figure 4-10: Some more test images, but on a much smaller scale – the individual pixels are plainly visible, and no smoothing is applied. These tests are modeled after an example in (Folsom and Pinter, 1998), but significantly scaled down.

Figure 4-9 shows the orientations measured for a 64×64 image consisting of a circle and square. This is based on an example in (Freeman and Adelson, 1991). The detector gives good results for solid edges with arbitrary contrast, and various kinds of lines. The response to edges is diffuse by design – during data collection, samples are taken both along the boundary and slightly to either side of it, and treated identically. If a sharper response is desired, these side-samples could be dropped, or their offset from the boundary could be recorded. Figure 4-10 shows the filter operating on a very small image of a cube. Each visible edge of the cube is clearly and faithfully represented in the output.

Figure 4-13 systematically explores the effect of adding noise to an "ideal" edge. The resilience of the orientation measure is encouraging, although a small number of gaps in coverage are revealed, suggesting that further data should be collected.

4.5 Discussion and Conclusions

The orientation detection scheme presented in this chapter has an unusual combination of properties, some of which are essential to the approach and some which are incidental details :-

- ▷ **Data driven (versus model based).** Detection relies heavily on the existence of training data – it is not achieved directly based on a formal model of edges instantiated in an algorithm.
- ▷ **Uses look-up table (versus neural network, support vector machine, etc.).** The use of training data is simply to populate a look-up table, rather than anything more elaborate.
- ▷ **Autonomous data collection (versus human annotation).** Training data is collected by the robot, and not a human.

Data driven

Work on edge and orientation detection has historically been model based, rather than data driven. To make progress analytically, the nature of edges was grossly simplified – for example, researchers worked with additive Gaussian noise overlaid on a luminance step (see, for example Canny (1986)). Before long it was pointed out that edges can take a diversity of forms beyond steps or lines (Perona and Malik, 1990). With the introduction of diverse cases, an empirical approach becomes more attractive. Jitendra Malik's group are now looking at how to locate boundaries between objects in images using features trained on human-produced segmentations (Martin et al., 2002). Many other parameters of a modern edge detector can also profit from empirical training, and can be optimized per domain (Konishi et al., 2003). So there is clearly considerable scope for a data driven approach to edge detection to improve performance.

Look-up table

The use of look-up tables has an important place in AI and computer science, from the Huge Look-Up Table problem in philosophy (Block, 1978) to the implementation of elementary arithmetic operations in CPU design (Wong and Flynn, 1992). An interpolating look-up table is also just about the simplest possible learning module. When it can be used, the results of learning are much simpler to understand than is the case for neural networks, support vector machines, etc. For example, the work by McDermott (2000) on training a neural network to detect junctions ran into the problem that, even with sensitivity analysis, it can be hard to understand a network's failure modes. With a look-up table, it is trivial. Chapter 4 exploited this fact to provide several visualizations of cross-sections of the look-up table. The index into the look-up table used is quantized pixel values.

This stays closer to the raw image than other work (Konishi et al., 2003; Martin et al., 2002) which focuses on optimizing the combination of existing hand-designed features. In theory, this means the approach could capture unanticipated domain-specific properties that will not show up in the hand-designed cases. This possibility was not explored here since the work was implemented on a robot inhabiting a single fixed space.

Populating a look-up table makes for fast run-time operation. It is possible to make a filter bank approach that runs at comparable speeds – for example, orientation can be detected from the output of two 3×3 filters, if we are willing to put some work into interpreting the different responses to step and line edges (essentially line edges give doubled responses, and appear to be a pair of close step edges). The look-up table approach encapsulates this interpretation step automatically, since it is trained on the end-to-end judgement required (from pixels to angles).

Autonomy

Training examples of edges could be generated in many ways. For example, computer graphics could be used to make images with known ground truth, or human labelled datasets could be used as in Martin et al. (2002) and Konishi et al. (2003). The work of Konishi et al. (2003) has shown that domain-dependent improvements can be made in edge detection, so in that sense an argument can be made for adaptivity. An autonomous, empirical approach holds out the promise of developing a 'wise' low-level perceptual system that makes good context-sensitive guesses, making the job of higher level modules that much simpler. This was beyond the scope this thesis, but it was certainly a motivating consideration and a direction for future work.

While this work has shown that idealized edges are empirically well grounded, in that they occur more frequently than other variants, it also shows that many more exotic forms do occur and can profitably be modeled. At fine scales, where the number of pixels used to compute orientation is low, a practical approach is to sample the appearance of edges empirically and average over noise (see Figure 4-17). With the large cache size of modern processors, this memory-based approach to orientation detection can facilitate extremely rapid orientation detection, which is important for real-time vision systems (Kubota and Alford, 1995).

Rectangular windows are the natural size for real-time machine vision applications. The memory-based approach proposed here has the advantage that it can make use of every pixel in the window a principled way. Filters for orientation detection are typically circular in nature, and so must ignore pixels that lie outside the largest circle that fits inside the window.

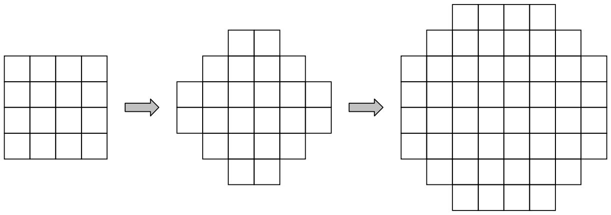

Figure 4-11: Possibilities for expanding the size of the orientation filter.

Can this technique be applied to larger windows? Not easily. In fact the window size used in one of the earliest papers on orientation detection, which had a diameter of eight pixels, seems completely out of reach (Hueckel, 1971). A 5×5 window of binary pixels can take on $2^{5 \times 5}$ possible values – about 33.6 million, or about 2.1 million allowing for symmetries. This would be hard to sample exhaustively, but with some further quantization a look-up table of that size would not be impossible. An intermediate possibility shown in Figure 4-11 involves one fewer pixel, and has a more symmetric shape: a 4×4 window augmented with 8 extra pixels to “round it off”.

Would an empirical approach work for features other than orientation? This isn’t clear, since not all features are as robust to pixel-wise transformation as orientation is – and hence it may not be as easy to explore their space of appearances as exhaustively.

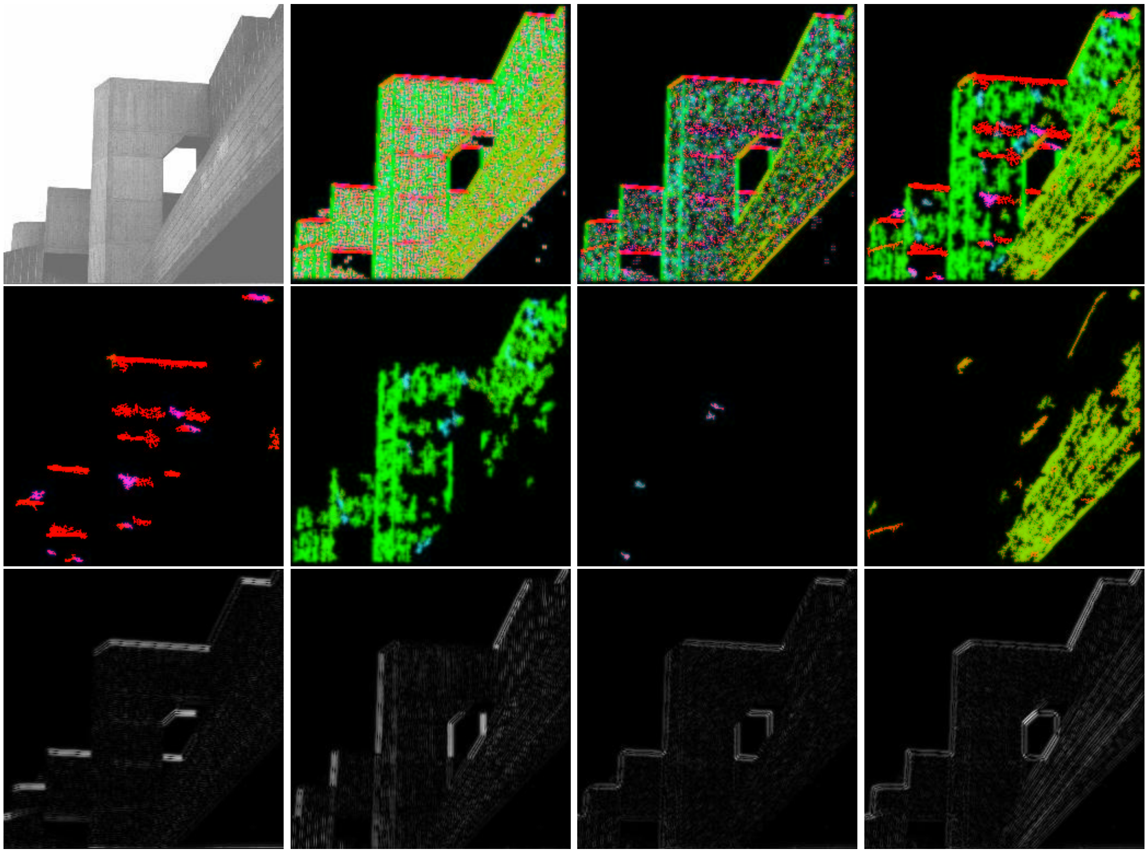

Figure 4-12: Here the orientation detector is applied to an image from Chabat et al. (1999). The top row shows from left to right the original image, output of the Folsom detector, output of the memory-based detector, and a simple enhancement of that output using region growing. The second row filters that output by orientation range (within 22.5° of horizontal, vertical, $+45^\circ$ and -45° respectively). The final row shows the output of a steerable filter by comparison, steered to the same nominal orientations, to reinforce that orientation does not pop out immediately from those filters – note for example the significant response in the third column at -45° even though there is nothing at that orientation in the image; they require a level of additional processing that the memory-based approach bypasses.

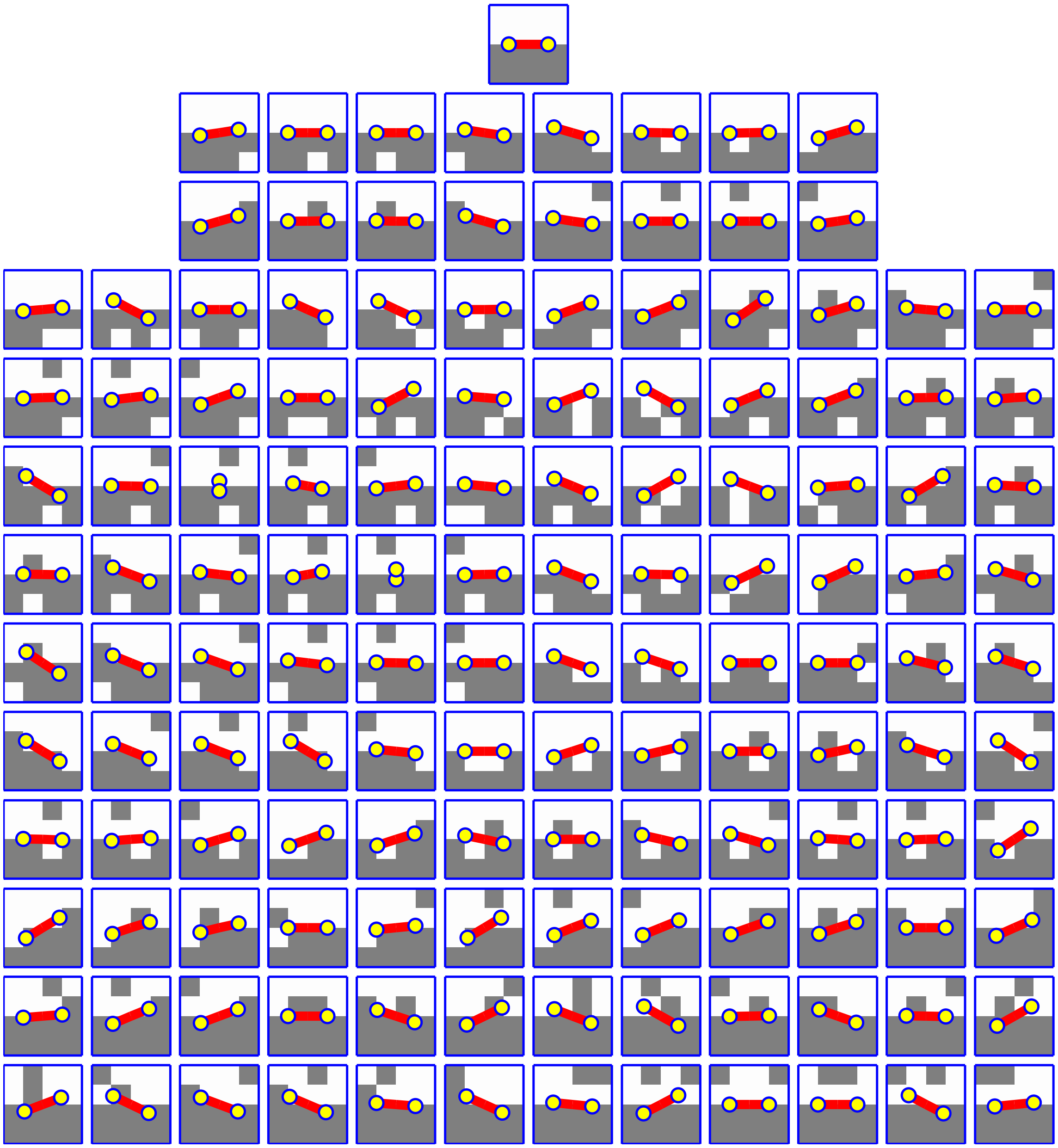

Figure 4-13: The top row shows an “ideal” view of a horizontal edge. The next two rows show patches with a single pixel perturbation from this ideal. The estimated angle remains close to horizontal. The rows that follow show 2-pixel perturbations (25% of the pixel weight of the edge). The behavior of the orientation estimate is generally reasonable. In two cases there is insufficient data for a good estimate (just one sample is available).

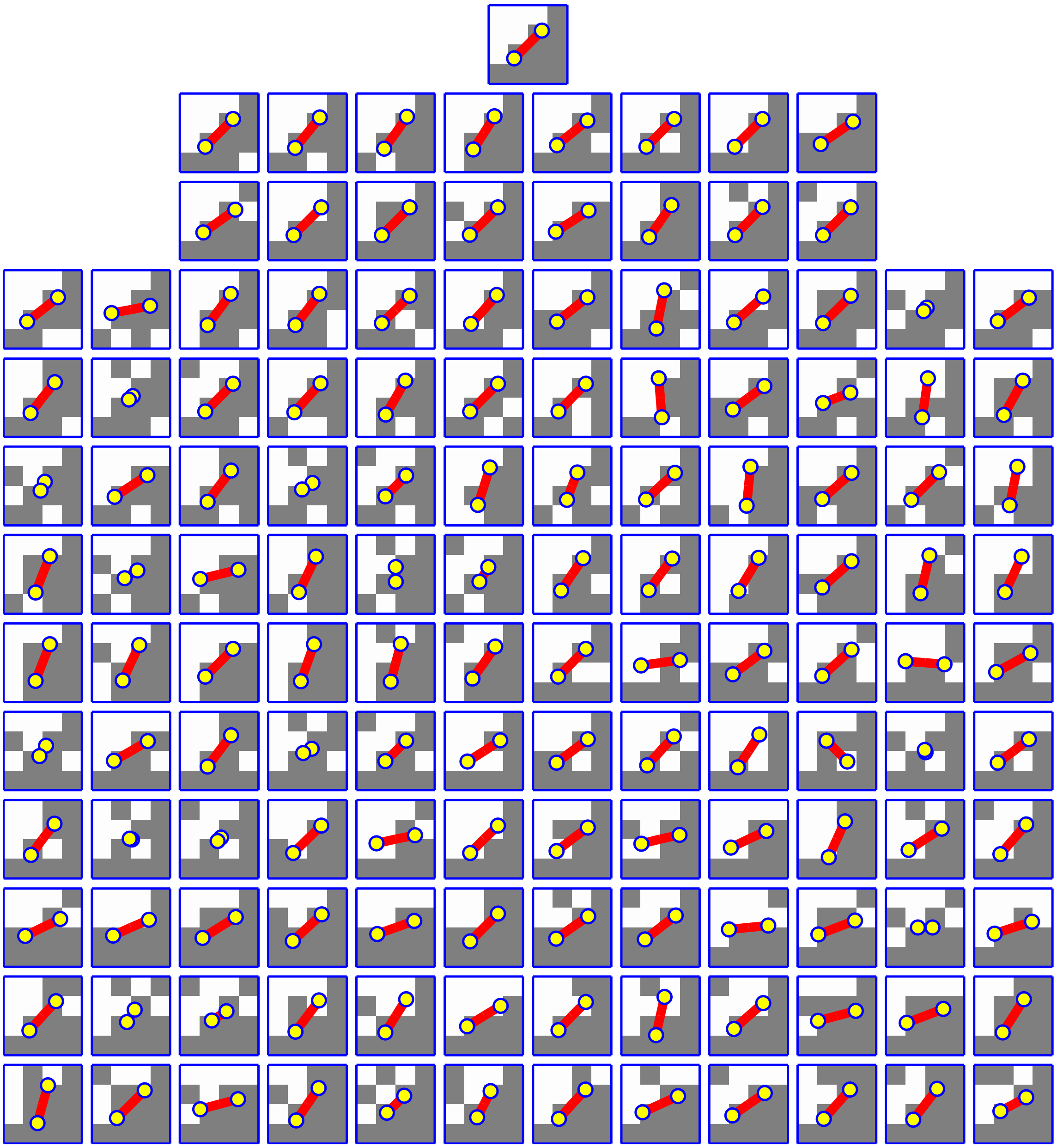

Figure 4-14: Perturbations from an “ideal” view of a diagonal edge. There are now several cases in which little training data is available, and one example of a divergent direction estimate (fourth row from bottom, third column from right).

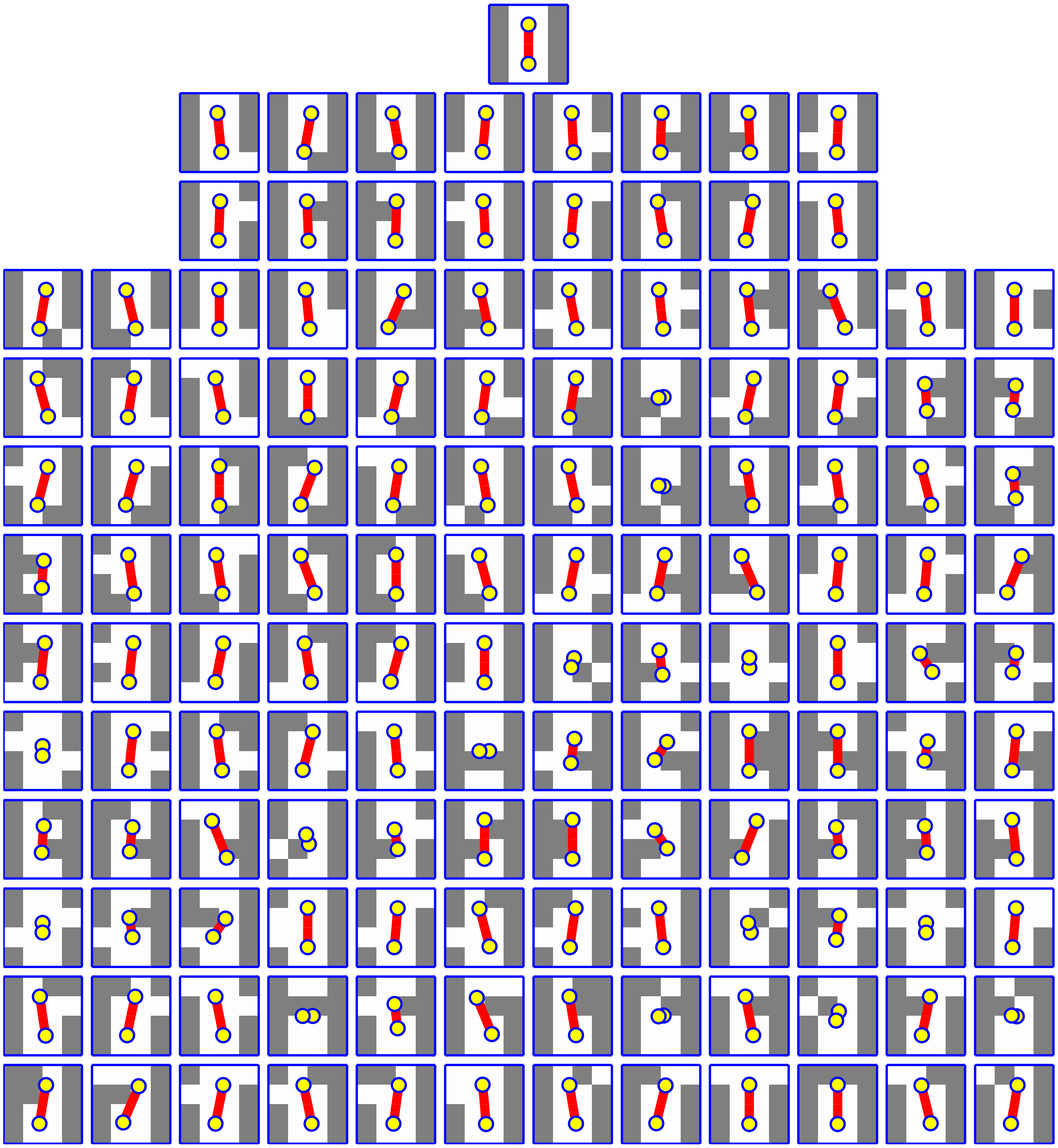

Figure 4-15: Perturbations from an “ideal” view of a tube.

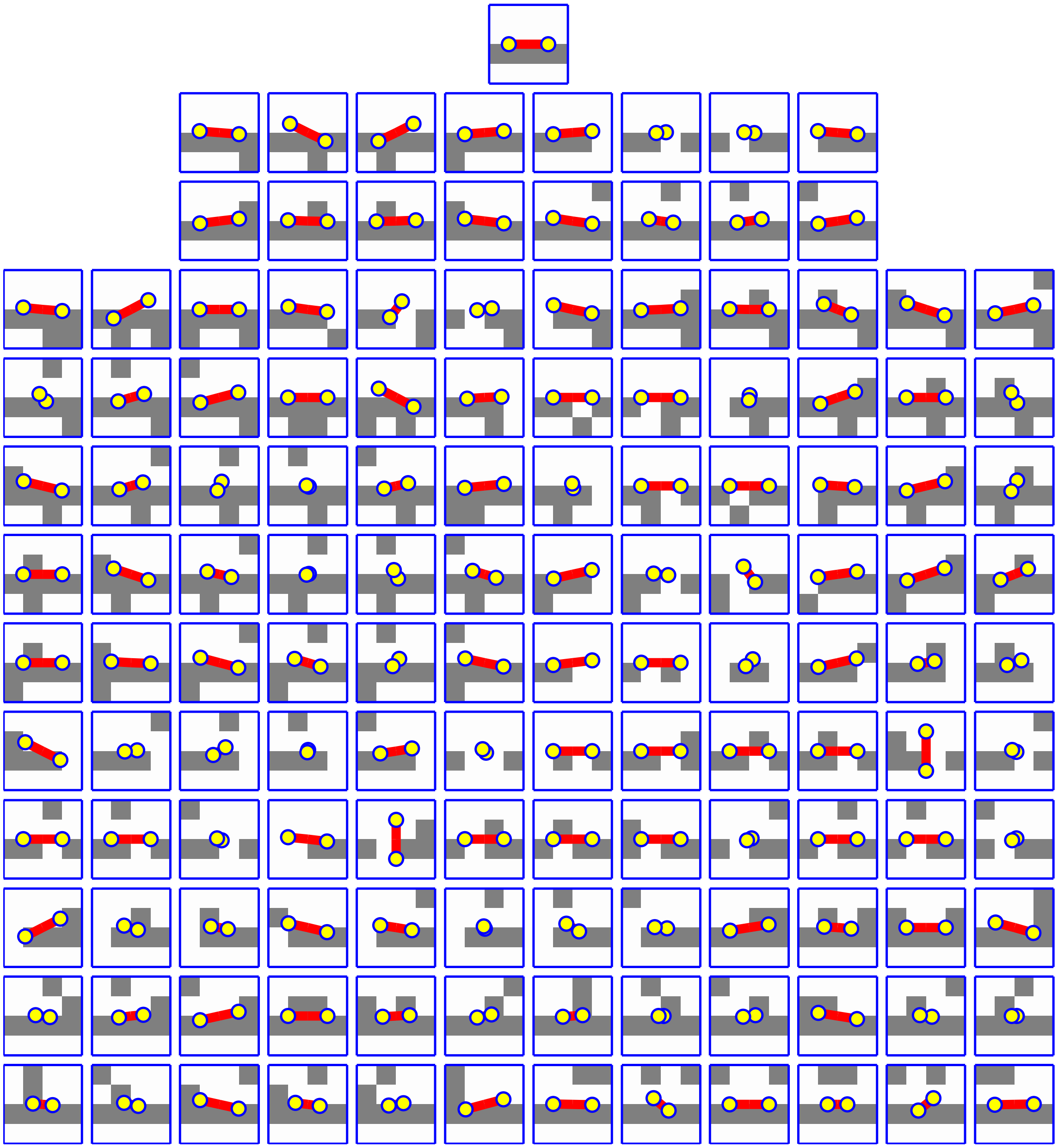

Figure 4-16: Perturbations from an "idea" view of a line. Here the pixel weight of the line is low, so the perturbations have a correspondingly more drastic effect. Lines are also seen less often in the training data, since they require special conditions at the object boundaries sampled (the boundary must be unlike the surface on either side of it). There is considerable room for improvement here if other sources of ground truth could be acquired. For example, orientation information could be propagated across time or space from neighboring patches of known orientation to less frequently encountered patches.

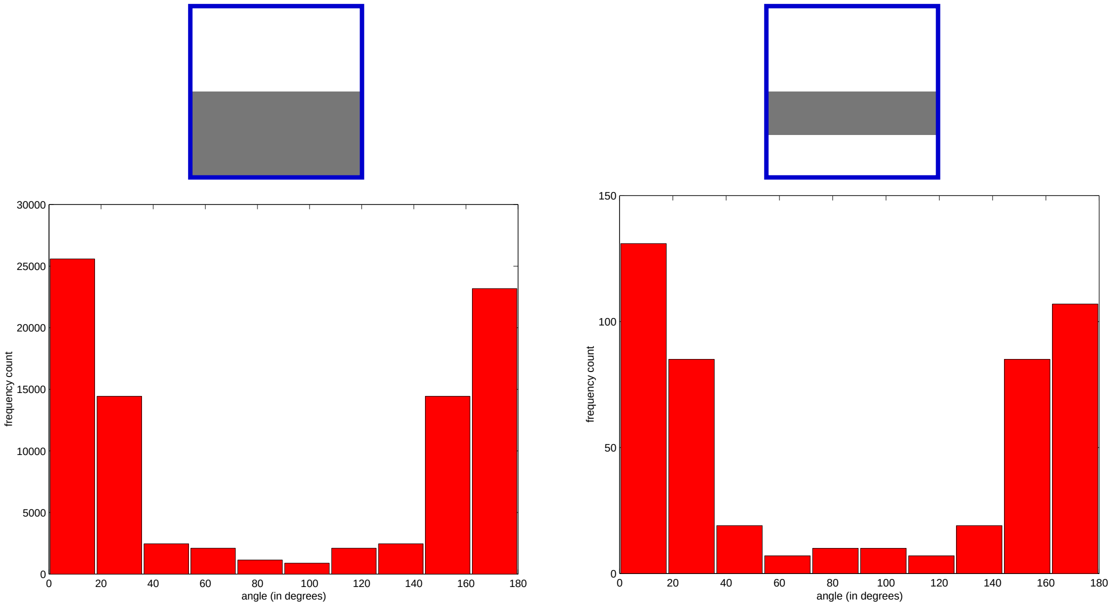

Figure 4-17: The plot on the left shows the frequency with which a thick step edge (as shown in Figure 4-13) is labelled with each possible angle. The distribution has a peak at 0°/180° as is appropriate, but other values do occur. About 7% of the samples lie at close to right angles to the nominally correct value. The plot on the right shows the same results for a thin line (as shown in Figure 4-16). The basic shape is the same, but the pattern occurs much less frequently overall – hundreds of times versus tens of thousands.