

# Probabilities and Decision Theory

We now describe a principled approach for combining the response of many features/filters to perform tasks like stereo or motion estimation. This approach is based on decision theory. The section also illustrates the importance of knowing whether filter responses, hence visual cues for the task, are dependent or independent.

We introduce the probabilities of filter responses by describing a classical experimental finding about natural image statistics. Intuitively the intensities of neighboring pixels tend to be similar. This intuition can be captured by taking derivative filters of the image, i.e.,  $\frac{dI}{dx}$  or  $\frac{d^2I}{dx^2}$ , and plotting their probability distribution, or histogram. Surprisingly these probability distributions are very similar from image to image [155]. This can be verified from interactive demo (3a) on natural image statistics. Interactive demo (3b) explores the statistics of edge detection and illustrates decision theory.

## Edge Detectors/ Texture Detectors and Decisions

Consider the tasks of deciding whether an *image patch* at position  $x$  contains an *edge* by which we mean the boundary of an object or a strong texture boundary (e.g., like the writing on a tea shirt). The previous section showed that some Gabor filters are tuned (i.e. respond strongly) to edges at specific orientations. But such filters will also respond to other stimuli such as texture patterns, so how can we decide if their response is due to an edge? The simplest way is to *threshold* the response so that an edge, at a specific orientation, is signalled if the filter response is larger than a certain threshold value. But what should that threshold be? How do we do a trade-off to balance *false negative* errors, where we fail to detect a true edge in the image, with *false positive* errors where we incorrectly label a pixel as an edge? Also each filter in a filterbank contains some evidence about the presence of an edge, so how can we combine their evidence in an optimal manner? How can we formulate the intuition that the evidence from some filters give *independent* evidence while others do not.

# Decision Theory

Decision theory gives a way to address these issues. The theory was developed as a way to make decisions in the presence of uncertainty. In this section we develop the key ideas of decision theory by addressing the specific task of edge detection. In the next section we give a more general treatment. We will only treat the case when we are detecting edges based on local evidence in the image. Later in this chapter we will extend to when we can use non-local, or contextual information. Interactive demo (3b) on decision theory and edge detection illustrates most of the ideas in these two sections.

# Filters

To start with, we consider the evidence for the presence of an edge using a single filter  $f(\cdot)$  only. We assume we have a benchmarked dataset so that at each pixel we have intensity  $I(x)$  and a variable  $y(x) \in \{\pm 1\}$  (where  $y = 1$  indicates an edge, and  $y = -1$  does the opposite). We apply the filter to the image to get a set of filter responses  $f(I(x))$ . If the filter is tuned to edges, then the response  $f(I(x))$  is likely to be higher if an edge is present than if not. This requires selecting a filter  $f(x)$ , such as the modulus of the gradient of the intensity  $|\vec{\nabla} I(x)| = \sqrt{\frac{dI}{dx}^2 + \frac{dI}{dy}^2}$  (since  $|\vec{\nabla} I(x)|$  is likely to be large on edges and small off edges).

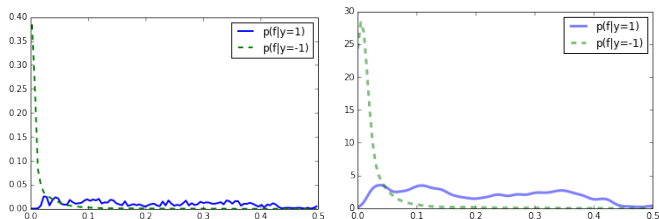
## Conditional Probability Distributions

To quantify this, we use the benchmarked dataset to learn *conditional probability distributions* for the filter response  $f(I)$  conditioned on whether there is an edge or not:

$$P(f(I)|y = 1), P(f(I)|y = -1).$$

Each distribution is estimated by computing the *histogram* of the filter response by counting the numbers of times the response occurs within one of  $N$  equally spaced bins and normalizing by dividing by the total number of responses. The histograms for  $P(f(I)|y = 1)$  and  $P(f(I)|y = -1)$  are computed from the filter responses on the points labeled as edges  $\{f(I(x)) : y(x) = 1\}$  and not-edges  $\{f(I(x)) : y(x) = -1\}$  respectively. Typical conditional distributions are shown in figure (21).

## Figure for Conditional Distributions



**Figure 21:** The probability of filter responses conditioned on whether the filter is *on* or *off* an edge –  $P(f|y = 1)$ ,  $P(f|y = -1)$ , where  $f(x) = |\vec{\nabla} I(x)|$ . Left panel: the probability distributions learnt from a dataset of images. Right panel: the smoothed distributions after fitting the data to a parametric model.

## Statistical Edge Detection

We can now perform edge detection on an image, see figure (22). At each pixel  $x$  we compute  $f(I(x))$  and calculate the conditional distributions  $P(f(I(x))|y = 1)$  and  $P(f(I(x))|y = -1)$ . These distributions give local evidence for the presence of edges at each pixel. Note, however, that local evidence for edges is often highly ambiguous, see figure (23). Spatial context can supply additional information to help improve edge detection, as discussed in a later section, and so can high-level knowledge (e.g., by recognizing the objects in the image).

## Log-likelihood ratio

The log-likelihood ratio  $\log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)}$  gives evidence for the presence of an edge in image  $I$  at position  $x$ . This ratio takes large positive values if  $P(f(I(x))|y=1) > P(f(I(x))|y=-1)$  (i.e. if the probability of the filter response is higher given an edge is present) and large negative values if  $P(f(I(x))|y=-1) > P(f(I(x))|y=1)$ . So a natural decision criterion is decide that an edge is present if the log-likelihood ratio is greater than zero and that otherwise there is no edge. This can be formulated as a *decision rule*  $\alpha(x)$ :

$$\alpha(x) = 1, \text{ if } \log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)} > 0, \quad \alpha(x) = -1, \text{ if } \log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)} < 0.$$

This can be expressed, more compactly, as

$$\alpha(x) = \arg \max_{y \in \{\pm 1\}} y \log \frac{P(f(I(x))|y=1)}{P(f(I(x))|y=-1)}.$$



# Statistical Edge Detection Figure



**Figure 22:** The input image and its groundtruth edges (far left and left). The derivative  $dI/dx$  of the image in the  $x$  direction (center). The probabilities of the local filter responses  $P(\vec{f}(I(x))|y = 1)$  (right) and  $P(\vec{f}(I(x))|y = -1)$  (far right) have their biggest responses on the boundaries and off the boundaries respectively, hence the log-likelihood ratio  $\log \frac{P(\vec{f}(I(x))|y=1)}{P(\vec{f}(I(x))|y=-1)}$  gives evidence for the presence of edges.

## Ambiguities in Edge Detection

Note that this rule gives perfect results (i.e. is one hundred percent correct) if the two distributions do not overlap, i.e. if

$P(f(I(x))|y = 1)P(f(I(x))|y = -1) = 0$  for all  $I$ . In this case it is impossible to confuse the filter responses to the different types of stimuli. But this situation is very unlikely to happen. Now consider a more general *log-likelihood ratio test* which depends on a threshold  $T$ , this gives a rule:

$$\alpha_T(x) = \arg \max_{y \in \{\pm 1\}} y \left\{ \log \frac{P(f(I(x))|y = 1)}{P(f(I(x))|y = -1)} - T \right\}.$$

By varying  $T$  we get different types of mistakes. We can distinguish between the *false positives* which are non-edge stimuli which the decision rule mistakenly decides to be an edge, and *false negatives* which are edge stimuli which are mistakenly classified as not being edges. Increasing the threshold  $T$  reduces the number of false positives but at the cost of increasing the number of false negatives, while decreasing  $T$  has the opposite effect.

## Ambiguity of Edges Figure



**Figure 23:** The local ambiguity of edges. An observer has no difficulty in detecting all the boundary of the horse if the full image is available (left). But it is much more difficult to detect edges locally (other panels).

## Decision Theory and Trade-Offs

Making a decision requires a trade-off between these two types of errors. Bayes decision theory says this tradeoff should depend on two issues. Firstly, the *prior* probability that the image patch is an edge. Statistically most image patches do not contain edges, so we would get a small number of total errors (false positives and false negatives) by simply deciding that every image patch is non-edge. This would encourage us to increase the threshold  $T$  (to  $-\infty$  so that every image patch would be classified as non-edge). Secondly, we need to consider the *loss* if we make a mistake. If our goal is to detect edges, then we may be willing to tolerate many false positives provided we keep the number of false negatives small. This means we choose a decision rule, by reducing the threshold  $T$ , so that we detect all the real edges but also output “false edges”, which we hope to remove later by using contextual cues (see the next section). Later we show how this approach can be justified using the framework of decision theory.

## Combining Multiple Cues for Edge Detection

Now we consider combining several different filters  $\{f_i(.)|i = 1, \dots, M\}$  to detect an edge by estimating the *joint* response of all the filters

$P(f_1, f_2, \dots | y) = P(\{f_i(I(x))\} | y)$  *conditioned* on whether the image patch  $I$  at  $x$  is an edge  $y = 1$  or not an edge  $y = -1$ . This leads to a decision rule:

$$\alpha_T(I(x)) = \arg \max_{y \in \{\pm 1\}} y \left\{ \log \frac{P(\{f_i(I(x))\} | y = 1)}{P(\{f_i(I(x))\} | y = -1)} - T \right\}.$$

This approach has two related drawbacks. Firstly, the joint distributions require a large amount of data to learn particularly if we represent the distributions by histograms. Secondly, the joint distributions are “black-boxes” and give no insight into how the decision is made. So it is better to try to get a deeper understanding of how the different filters contribute to making this decision, by studying whether they are *statistically independent*.

## Combining Cues with statistical independence

The response of the filters is statistically independent if:

$$P(\{f_i(I(x))\}|y) = \prod_i P(f_i(I(x))|y) \text{ for each } y$$

This implies that the distributions  $P(f_i(I(x))|y)$  can be learnt separately (which decreases the amount of data) and also implies that the log-likelihood test can be expressed in the following form:

$$\alpha_T(x) = \arg \max_{y \in \{\pm 1\}} y \left\{ \sum_i \log \frac{P(f_i(I(x))|y = 1)}{P(f_i(I(x))|y = -1)} - T \right\}$$

Hence the decision rule corresponds to summing the evidence (the log-likelihood ratio) for all of the filters to determine whether it is above or below the threshold  $T$ . This means that each filter gives a “vote”, which can be positive or negative, and the decision is based on the sum of these votes. This process is very simple so it is easy to see which filters are responsible for the decision.

## Combining cues with conditional independence

But very few filters are statistically independent. For example, the response of each filter will depend on the total brightness of the image patch and so all of them will response more to a “strong” edge than to a “weak” edge. This suggests a weaker independence condition known as *conditional independence*. Suppose we add an additional filter  $f_0(I(x))$  which, for example, measures the overall brightness. Then it is possible that the other filters are statistically independent conditioned on the value of  $f_0(I(x))$ :

$$P(\{f_i(I(x))\}, f_0(I(x))|y) = P(f_0(I(x))|y) \prod_i P(f_i(I(x))|f_0(I(x)), y)$$

This requires only representing(learning) the distributions  $P(f_i(I(x))|f_0(I(x)), y)$  and  $P(f_0(I(x))|y)$ .

## Combining cues with conditional independence

It also leads to a simply decision rule:

$$\alpha_T(x) = \arg \max_{y \in \{\pm 1\}} y \left\{ \log \frac{P(f_0(I(x))|y=1)}{P(f_0(I(x))|y=-1)} + \sum_i \log \frac{P(f_i(I(x))|f_0(I(x)), y=1)}{P(f_i(I(x))|f_0(I(x)), y=-1)} \right\} - T$$

It has been argued [137] that methods of this type can be implemented by neurons and may be responsible for edge detection. Note that the arguments here are general and do not depend on the type of filters  $f_i(\cdot)$  or whether they are linear or non-linear. It has, for example, been suggested that edge detection is performed using the energy model of complex cells [118].



## Classification for other visual tasks

The same approach can be applied to other visual tasks. For example, consider using local filter responses to classify whether the local image patch at  $x$  is "sky", "vegetation", "water", "road", or "other", see figure (24). We denote these by a variable  $y \in \mathcal{Y}$  (e.g., where  $\mathcal{Y} = \{\text{"sky"}, \text{"vegetation"}, \text{"water"}, \text{"road"}, \text{or "other"}\}$ ). We choose a set of filters  $\{f_i(I(x))\}$  which are sensitive to texture and color properties of image patches. Then, as before, we learn distributions  $P(\{f_i(I(x))\}|y)$  for  $y \in \mathcal{Y}$ . We select a decision rule of form:

$$\alpha(I(x)) = \arg \max_{y \in \mathcal{Y}} P(\{f_i(I(x))\}|y) T_y,$$

where  $T_y$  is a set of thresholds (which can be derived from decision theory). Experiments on images show that this method can locally estimate the local image class with reasonable error rates for these types of classes [88] and computer vision researchers have improved these types of results using more sophisticated filters.

# Classifying other image classes



**Figure 24:** Classifying local image patches. The image shows the groundtruth, see [119]. Certain classes – sky, grass, water – can be classified approximately from small image patches.

We stress that the theories described in this section model edge detection *without context*. There are two types of context we will consider in this chapter. The first uses spatial context and is low- or mid-level since it depends only on *generic* properties of images and surfaces. It exploits the idea that edges in natural images are often geometrically regular and co-linear. The second type of context, is high-level and is object specific. For example, if we detect a face in an image then our knowledge about faces enables us to detect the boundaries of a face better than if we rely only on local edge cues. This second type of context is out of scope of this chapter but is briefly discussed at the end of these lectures.