Learning Optimal Linear Filters for Early Vision

Xiaofang Sun and David G. Lowe
Department of Computer Science
University of British Columbia
Vancouver, B. C., Canada V6T 1W5

Abstract

Edge detection is important both for its practical applications to computer vision as well as its relationship to early processing in the visual cortex. We describe experiments in which the *rescaling backpropagation* learning algorithm was used to learn sets of linear filters for the task of determining the orientation and location of edges to sub-pixel accuracy. A model of edge formation was used to generate novel input-output pairs for each iteration of the training process. The desired output included determining the interpolated location and orientation of the edge. The linear filters that result from this optimization process bear a close resemblance to oriented Gabor or derivative-of-Gaussian filters that have been found in primary visual cortex. In addition, the edge detection results appear to be superior to the existing standard edge detectors and may prove to be of considerable practical value in computer vision.

1 Introduction

As edge detection plays an important role in many aspects of computer vision, a great deal of research has been devoted to the design of optimal edge detectors. Some of the most widely used existing edge detectors are those by Marr and Hildreth[1] and Canny[2]. Although the designs of these edge detection methods were motivated in part by knowledge of biological vision, their sensitivity and accuracy is much less than that exhibited by biological systems. In particular, these detectors take account of only a narrow range of spatial frequencies in the image and fail to make use of oriented filters. Although many attempts have been made to overcome these limitations, none of them have been successful enough to receive widespread use.

It is known that early vision in mammals is performed in part by the simple cells of primary visual cortex. Each of these neurons responds to edges at a particular range of orientations and spatial frequencies at a single location on the retina, with a response that is linear over a substantial range of inputs[3]. In order to study the role of these oriented linear filters in edge detection and to develop practical methods for combining filters at multiple scales and orientations, we have used rescaling backpropagation learning to develop filters for performing an edge detection task. Plaut and Hinton[4] have described the use of backpropagation for learning filters that are suitable for early processing of data in both speech and vision. As the hidden units in the most common form of the backpropagation algorithm respond to a linear weighted sum of their inputs, they can be viewed as forming linear spatial filters through the modification of their input weights. These filters can then be combined in a non-linear fashion to produce the final edge detection output.

The initial results of these experiments show that the receptive fields that are learned are similar in form to those that are known to exist in primary visual cortex. In addition, the performance of the network as an edge detector appears to be superior to existing edge detectors.

International Joint Conference on Neural Networks, Singapore (November 1991)

2 Network Structure and Rescaling Backpropagation Algorithm

We use a 3-layer feed-forward network, including a 7×7 input layer, a hidden layer with 70 neurons, and a 17-neuron output layer.

Rescaling backpropagation[5] is implemented to train the network. In the backpropagation[6] algorithm, the chain rule of differentiation introduces the factor $o * (1 - o)$, o is the output of each node. The factor $o*(1-o)$ occurs once in the output layer, twice on the first hidden layer, ..., and so on, until the input layer is reached. Also, it should be noted that $0 \leq o*(1-o) \leq 1/4$ when o is in the interval $[0.1, 0.9]$. So a major cause of the ill-conditioned nature of BP is the gradient element at the different layers involve a fraction which can not exceed $1/4, 1/16, \ldots$ at the various layers, causing the gradient vector to differ radically in magnitude. The compensatory rescaling is need for the partial derivative. The rescaling factor $4, 16, \ldots$ is applied to the sequential layers in our implementation. Comparing with BP, RBP is much faster, the training time plot is shown in figure 1.

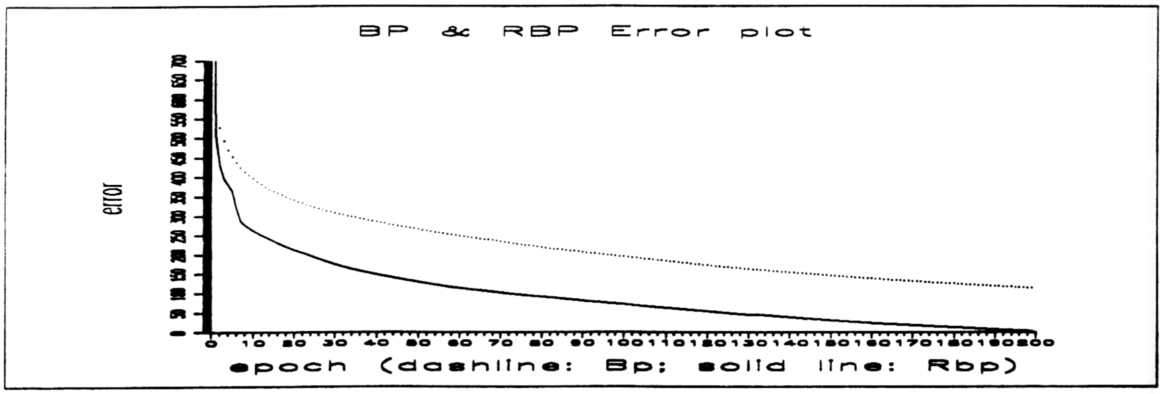


Figure 1: The epoch error: Backpropagation vs. Rescaling Backpropagation.

3 Training Data Generation and Features Representation

A 7×7 window is used to scan the computer-generated overlapped-polygon images. The image window is fed to the input layer of the network. Various kinds of images are designed for the training pairs, such as sharp images, low contrast images, blurred images, strips, blobs, one-pixel-wide thin lines, heavily-noise-polluted images, short bars, etc, some examples are illustrated in figure 2.

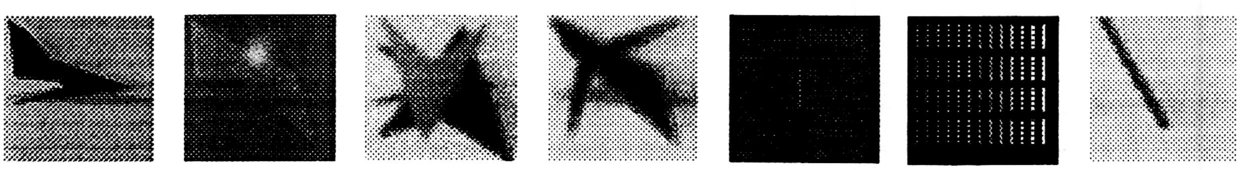


Figure 2: Some examples of training images.

For desired outputs, 17 visual primatives need to be learned by the network. 16 output units are for 8 directions, with 2 units representing one direction to distinguish the intensity change from dark to white or from white to dark (this is a good way to represent the two edges of one-pixel-wide thin line). The angle increment is 22.5 degrees, with 8 directions to cover the whole plane. The value of the desired output is in the interval $[0.1, 0.9]$, which increases linearly as the center pixel gets closer to an edge, and decreases linearly as the center pixel gets farther away from an edge. The 17th output is a general edge marker, which takes the sum of the 16 direction neurons outputs. Similarly, the value of each pixel linearly interpolates between neighboring orientation outputs. An example of an overlapped polygon and its 17 desired outputs is given in figure 3.

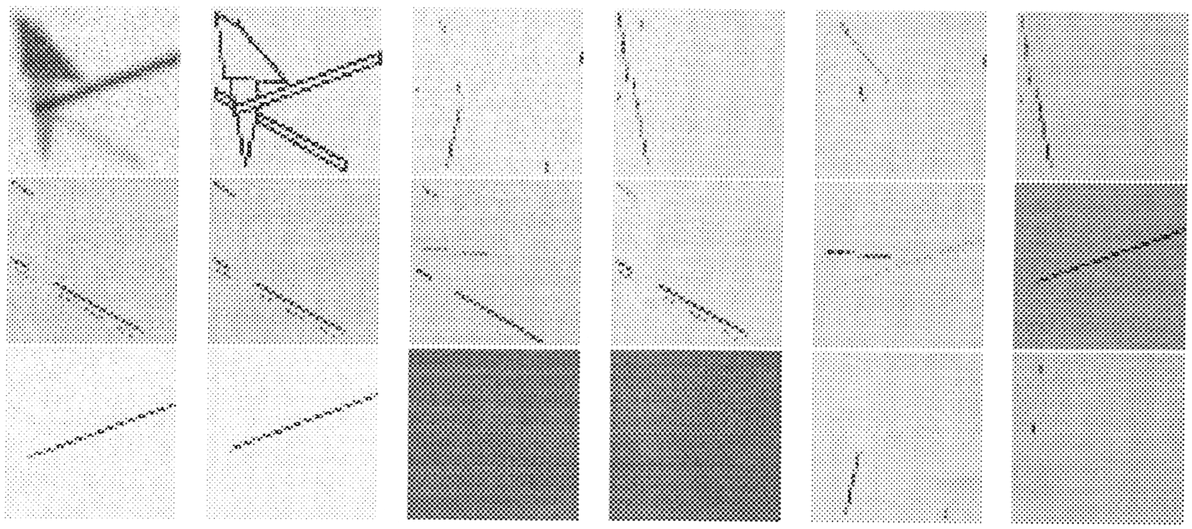


Figure 3: An example of an overlapped polygon and its 17 desired outputs.

4 Visual Receptive Fields and Hidden Units Analysis

Early vision in mammals is performed in part by the simple cells of primary visual cortex. Orientation-selective properties of these neurons make it possible to encode orientations of image elements and to compress initial information.

When the network finishes training, the weights are found to have converged into recognizable patterns. The weight patterns differ from each other in orientation and phase (displacement from the center), see figure 4.

A close qualitative correspondence has been found between the weight pattern and the visual neuron receptive fields. The visual image received by the receptors is filtered by the receptive fields to get the maximum response. From figure 4, we can see the weight functions are close to Gaussian derivative filters, including non-oriented (Laplacian of Gaussian) and oriented Gaussian derivatives. The resemblance of retinal receptive fields to non-oriented Laplacian of Gaussian filters, and the resemblance of simple cortical receptive fields to oriented Gaussian derivative-like filters can be noted.

5 Experimental Result and Subpixel Calculation

After training with 100 50×50 images, the weights are usually converged to stable values. Figure 6 is a testing result of a real image taken with a TV camera. Subpixel location is calculated by linearly interpolating the maximum response and the nearby pixel. Each time a maximum response direction is chosen, and the edge is traced in this direction, a non-maximum suppression operation is done, the accurating procedure is plot in figure 5.

The testing result is compared with Canny's edge detector, shown in figure 6. In general, the results from the neural network are significantly more detailed and more precise in locating edges and

Figure 4: The weights connected to 70 hidden units: square regions show the 7×7 set of weights from input units to hidden units. Contrast has been normalized for the display of each unit.

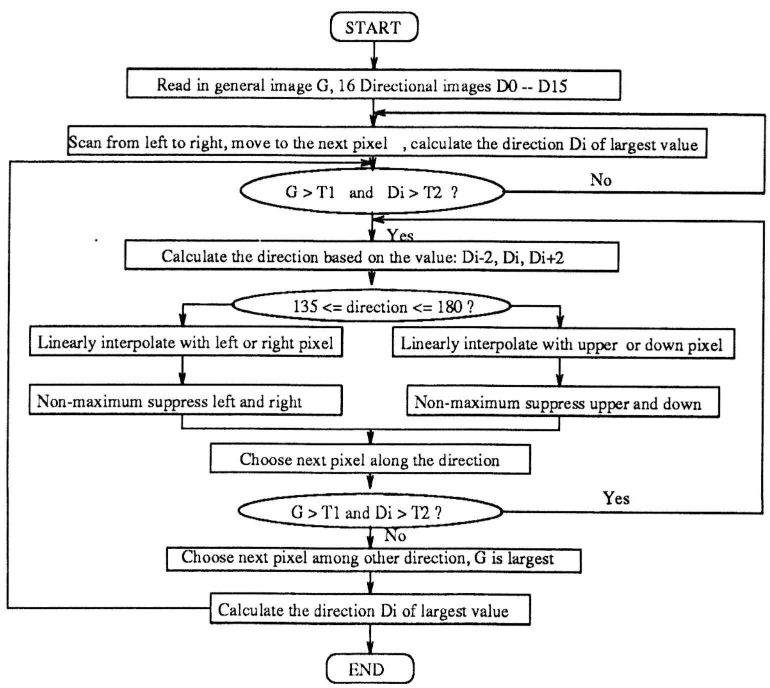
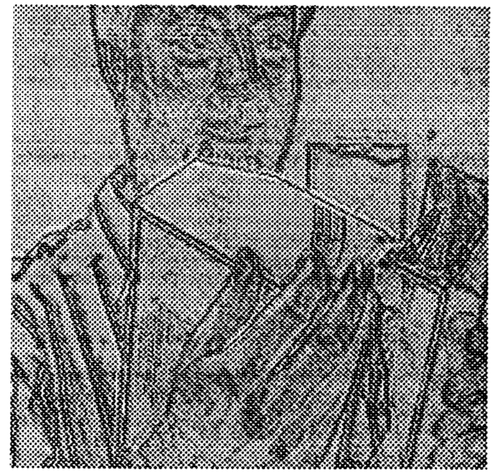


Figure 5: The subpixel calculation procedure.

less sensitive to noise than Canny's edge detector. The subpixel calculation can still be improved; however, the current result is considered to be very encouraging.

(a) Original image



(b) The testing result of using neuralnet

(c) Canny's detecting result, $\sigma = 0.7$

(d) Canny's detecting result, $\sigma = 1.5$

(e) The result after thinning

Figure 6: Shows a 250×250 pixel region of an image taken by a TV camera of a person holding a box.

6 Conclusions and Significance

We have used RBP to learn a set of linear filters and subsequent non-linear operations to perform edge detection. This system has developed a range of oriented and non-oriented receptive fields that bear a close similarity to the response of simple cells in primary visual cortex. In addition, this network shows considerable promise as a practical edge detector for use in computer vision.

References

[1] D. Marr and E. C. Hildreth, *Theory of Edge Detection*, Proceedings of the Royal Society of London, B. 207, pp. 187-217, 1980.

[2] Canny, *A Computational Approach to Edge Detection*, IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. 8, pp. 184-203, 1986. and J.A. Movshon, eds, MIT Press, 1991.

[3] Heeger, David J., *Nonlinear Model of Neural Responses in Cat Visual Cortex*, to appear in Computation Models of Visual Processing. M. Landy and J.A. Movshon, eds, MIT Press, 1991.

[4] David C. Plaut and Geoffrey E. Hinton, *Learning Sets of Filters Using Backpropagation*, Computer Speech and language, Vol. 2, pp. 35-61, 1987.

[5] A. K. Rigler, J. M. Irvine, and T. P. Vogl, *Rescaling of Variables in Back Propagation Learning*, Neural Networks, Vol 4, pp. 225-229, 1991.

[6] D. E. Rummelhart, G. E. Hinton, R. J. Williams, *Learning Internal Representation by Error Propagation*, Parallel Distributed Processing, Vol. 1, Ch. 8, pp. 318-362, 1986.