Learning-based versus model-based log-polar feature extraction operators: a comparative study

Herman Martins Gomes*
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
R. Aprígio Veloso 882
58109-970 Campina Grande, PB, Brazil
hmg@dsc.ufcg.edu.br

Robert B. Fisher
School of Informatics
Edinburgh University
5 Forrest Hill
EH1 2QL Edinburgh, Scotland, UK
rbf@inf.ed.ac.uk

Abstract

In this paper, we compare two distinct primal sketch feature extraction operators: one based on neural network feature learning and the other based on mathematical models of the features. We tested both kinds of operator with a set of known, but previously untrained, synthetic features and, while varying their classification thresholds, measured the operator's false acceptance and false rejection errors. Results have shown that the model-based approach is more unstable and unreliable than the learning-based approach, which presented better results with respect to the number of correctly classified features.

1 Introduction

Primal sketch features, like *Edges*, *Bars*, *Blobs* and *Ends* are part of Marr's [7] hypothesis for the human visual system, and are believed to take part in early visual processes providing a representation that captures invariant primitives from the object's surface as well as providing cues for an attention mechanism. Marr's original idea was to use differential operators, such as the Laplacian of Gaussian operator, to detect intensity changes (or zero-crossings) at different scales, which would later be used to form these features.

Inspired by Marr's ideas, Grove and Fisher [4] conceived a vision system based on primal sketch features. Instead of traditional Cartesian sensor geometry, they decided to use a biologically inspired representation, a log-polar image, which brought a number of advantages into their system but had a unusual geometry and computation. They proposed an alternative method to detect the features which was based on operators defined as small windows containing weights associated with each of the feature classes.

Later, Gomes and Fisher [2, 3] identified some problems with the way feature extraction was made in this system. Since the operators were heuristically defined, there is no guarantee that they will work correctly with all the possible cases and that they will allow graceful degradation. Moreover, whenever one needs a different window size or window shape it will be necessary to design new logical expressions for the operators which can lead to mistakes as the operators are defined by hand. In order to improve the previous feature extraction method, they proposed a learning-based approach, which is described, later in the paper, alongside the previous approach.

Although they performed a subjective evaluation of both approaches based on the visual output of the operators on a set of real images, which showed some improvement when using the learning-based approach, to date, there was no objective or numerical comparison between both approaches. Thus, the main purpose of this paper is to objectively compare these two distinct primal sketch feature extractors: one based on neural network feature learning and another one based on descriptive models of the features. A secondary objective is to improve on the subjective analysis by means of applying the operators to a set of simpler (synthetic) images, which made easier a subjective evaluation of the operators' visual output.

Section 2 presents some details as well as the motivations for a log-polar image representation, followed by a discussion of the model-based operators (Section 3) and the learning-based ones (Section 4). In Section 5, we show and analyse the results of an objective comparison (based on classification errors from synthetic features). Section 6 contains a more meaningful way of subjectively comparing both approaches.

*This author received financial support from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil)

2 Log-polar representation

Traditionally, the photometric information of images is acquired from sensors that have a uniformly distributed rectangular array of sampling units. As a consequence of the sensor architecture, most machine vision applications tend to use Cartesian representations to manipulate images.

However this is not the way images are acquired in the mammalian visual system. A biologically inspired approach to vision is to transform the original 2-D image into a retina-like representation and then to use this representation as the main data for the vision processes, like feature extraction, matching and attention. There is already a reasonable number of related work previously published on this kind of image representation [8, 9, 4, 6, 5].

The retina is responsible for the reception and transmission of the input light signals to neuron layers located on the photoreceptive surface. Each of these neurons receives the outputs of a group of photoreceptors on an approximately circular area of the retina called the receptive field. Typically, the input image is re-sampled through the use of a mask consisting of concentric rings of overlapping circular receptive fields whose centres are geometrically spaced from the centre of the mask. The central region of the retina (fovea) is formed by receptive fields with approximately constant size and hexagonally organised, while in the most external region, the receptive fields are distributed circularly with an area exponentially increasing as a function of the distance to the retina centre. There is a certain degree of overlap (typically 60%) between a receptive field and its neighbors, which prevents some Cartesian areas from not being covered by the retinal transformation (due to the circular geometry of the receptive fields).

If an image is accessed by using the rings (logarithm of the distance of the rings to the retina centre) and sectors of this type of mask then this is called a log-polar image representation. This essentially simulates the mapping between the retina and neurons in the visual cortex [10, 12]. Each receptive field value is computed through the integration of a function over a region of the input image. This non-uniform sampling can be realised either by software or by a space variant sensor. Figure 1 illustrates the mapping of a hypothetical 5 ring retina containing a 3 ring fovea. In our experiments, we used a retinal image containing 48 rings and 60 sectors (outside the fovea).

An important reason for the log and polar elements of our representation is the property that scaling and rotating an object located at about the centre of the retinal mask corresponds to translating the object in the log-polar image. This has been used to design systems that are scale and rotation invariant [9]. In another work [1], we specially took advantage of this property when creating visual models containing relative scale and orientation measurements for all

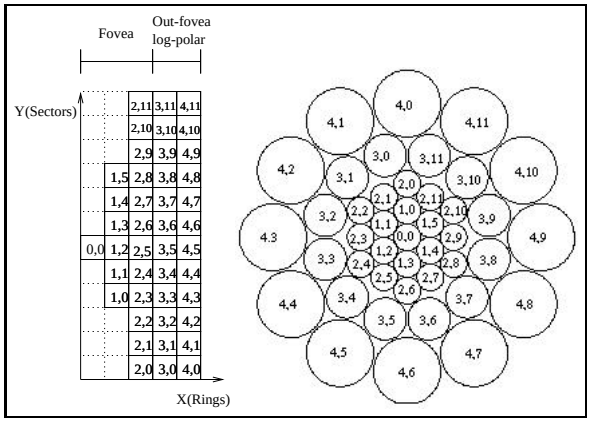

Figure 1. Log-polar mapping of a 5 ring retina containing a 3 ring fovea. To simplify the figure, no overlapping was used. Only the out of fovea area is log-polar.

pairs of the models' internal objects.

Some factors motivated the choice for a biologically inspired representation in the log-polar form. This does not necessarily mean that we want a representation that precisely mimics in size and complexity all the (sometimes not yet understood) structures found in a biological system. We are mainly looking at structural and functional similarities. One factor is the inherent reduced spatial complexity and log-polar property of the retinal image, which favours the implementation of faster matching algorithms. When used in conjunction with an attention mechanism, the space-variant nature of this representation may lead to a more robust matching process, when compared to conventional uniformly sampled images, since objects or parts of objects are more likely to be found in the high resolution centre. The low resolution periphery, which occupies only a small number of pixels, but covers a large area, will usually include parts of the background that are not so important for matching. There is much research on visual attention, including recent work by our group [11], but the operators discussed in this paper are concerned only with the features extracted after the saccade has occurred.

Although the image representation discussed has some interesting properties, more has to be done if one needs a system capable of performing high level vision tasks, such as locating, recognising or learning models of visual objects. More specifically, raw colour and intensity information are not sufficient for vision as their sole use would imply in a large conceptual gap between data and models and, therefore, this would complicate the design of algorithms. Both Machine Vision and Biology have proved that extracting properties and features from images, by means of build-

ing intermediate representations, is a useful way of reducing this conceptual gap. The following sections describe two distinct methods for extracting primal sketch features from log-polar images: model-based and learning-based.

3 Model-based operators

In Grove and Fisher's system [4], feature extraction operators were manually modelled as logical expressions involving the pixels of a 3x3 window which is applied through the log-polar image. Figure 2 illustrates the original 3x3 mask used in a rectangular retinal tessellation (i.e., the receptive fields are radially aligned). Some operators defining *-Corner*, *+Blob* and *+Bar* features are exemplified in the following Equations:

$$\begin{aligned} -Corner &= min(b-x, d-x, e-x) - |x-g|/2 \\ +Blob &= min(x-a, x-b, x-c, x-d, x-e, x-f, \\ &\quad x-g, x-h) \\ +Bar_{@90} &= min(b-a, x-d, g-f, b-c, x-p4, g-h) \\ &\quad -max(|x-b|, |x-g|) \end{aligned}$$

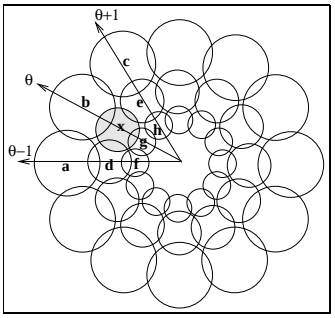

Figure 2. The 3x3 mask used by the Grove and Fisher system [4] to detect features. Each pixel in the mask corresponds to a particular receptive field output in the polar coordinate system. A rectangular retinal tessellation was used in this example.

They also proposed a set of operators for use in a triangular tessellation (i.e., when the receptive fields are shifted by half a sector every 2 rings in order to minimise the gaps in the mask. Instead of a 3x3 window, this version of the operators had a hexagonal layout of a central receptive field surrounded by its 6 neighbours (see Figure 3). This was the kind of tessellation and operators used in this paper. Examples of detectors for *Blobs* and *Edges* in triangular tessellation are shown in the following Equations:

$$\begin{aligned} +Blob &= min(x-a, x-b, x-c, x-d.x-e.x-f) \\ -Blob &= min(a-x, b-x, c-x, d-x, e-x, f-x) \\ Edge_{@0} &= abs(f+a+b-c-d-e)/3 \\ Edge_{@60} &= abs(a+b+c-d-e-f)/3 \\ &\cdots\cdots\cdots \\ Edge_{@300} &= abs(e+f+a-b-c-d)/3 \\ Edge &= min(Edge_{@0}, Edge_{@60}, \ldots, Edge_{@300}) \end{aligned}$$

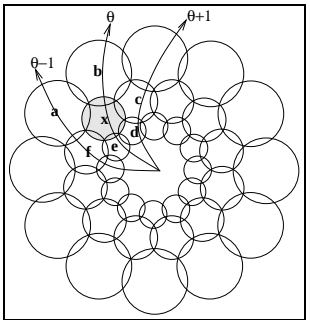

Figure 3. Triangular retinal tessellation.

Since the operators' outputs are continuous, any output below a given empirical positive threshold was set to zero to indicate the absence of a feature pattern. Originally, a fixed threshold equal to 8 was used for all operators, provided that image intensities were within the range $[0...255]$. During the comparison carried out in this paper, we varied this threshold in order to plot a classification error graph of the operator's output.

4 Learning-based operators

Figure 4 illustrates the architecture of the learning-based operators, which is based on a MLP-backpropagation neural network minimising a least square error metric.

The architecture receives receptive field windows already orientation normalised. This normalisation was achieved via a set of very simple symmetrical operators. At the top of the architecture is a PCA pre-processing stage which consists in projecting a normalised receptive field window onto a subset of the training set's principal components. This stage was needed in order to spread out the features, thus increasing the inter-class variability, which originally was too small due to the existence of similar feature shapes and due to the low dimensional input space.

At the bottom of the architecture is a neural network module with 17 inputs, 9 hidden neurons and 2 output neurons (N and $\bar{N}$) used to discriminate between feature and non-feature classes. Seven neural network modules like

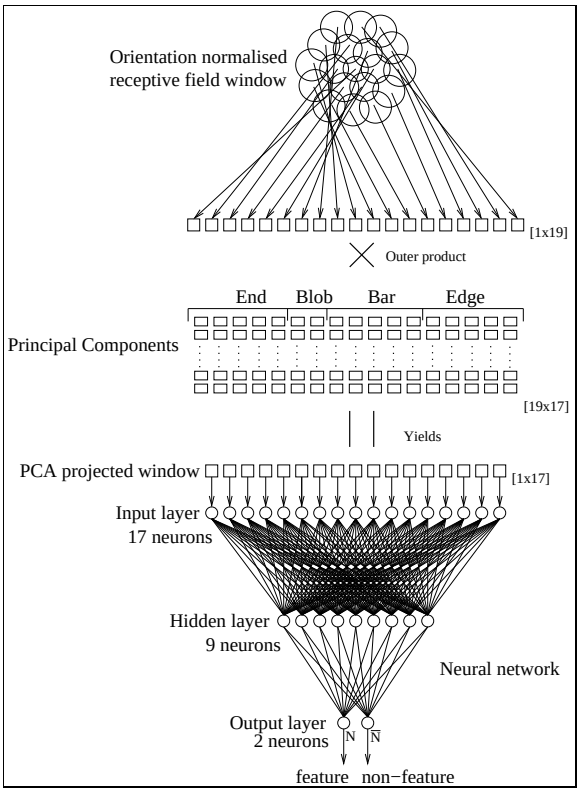

Figure 4. Architecture of the learning-based operators.

this were trained, one for each of the seven feature classes: *Edge*, *+Bar*, *-Bar*, *+Blob*, *-Blob*, *+End*, *-End*.

The networks were trained using a set of synthetic features so that a large number of feature variations could be easily generated. It is important to emphasise that the synthetic patterns are drawn in the Cartesian space and therefore this does not contradict our assumption that to build a model for the features in the retinal space is a difficult task which has justified the use of a learning based approach. Features were drawn using multiple contrasts, orientations, sizes and additive Gaussian noise levels, thus allowing a wide range of variation. Table 1 shows some of the training features generated. A total of seven different training sets was built, one for each neural network module designed to classify the feature classes. Each training set was built from an equal number of feature examples and counter examples. The same generator that was used to draw these features was also used to generate test features. Care has been taken so that the test features were different from the training ones, and thus completely new to the neural network modules.

More details about the learning-based operators and how they were trained can be found in two previous works [2, 3]. It should be pointed out that, in these previous works, the

	Edge	*+Bar*	*-Blob*	*+End*	*Counter Example*
Cartesian inputs					does not apply
Retinal outputs					

Table 1. Some examples of the synthetic training features.

performance of the learning-based operators was not quantitatively compared with the model-based operators, only some real images were presented for subjective evaluation (which proved to be a difficult task, since real images have too many details to examine). Besides contributing with an objective (numerical) comparison between both kinds of operators, the present paper also confronted their outputs when submitted to simpler, synthetic images, making easier the job of performing a subjective evaluation of the visual output of both operators.

5 Objective Comparison

The strategy we adopted in order to have an objective way of comparing the performance of both approaches presented in this paper was to vary their classification thresholds and measure the resulting classification errors when processing synthetic testing features. We tested both kinds of operator with a set of untrained synthetic features. We employed the same feature generator used to build the initial training sets with the difference that here the features were generated at sizes, orientations, contrasting intensities and noise level different from those used during neural network training.

We generated 80 test exemplars per class by varying the contrast in the range 0.21 to 1.0, with a step of 0.01 (-0.21 to -1.0, with a step of -0.01, for negative features). The intensities used to produce the above contrasts were chosen randomly as well as the remaining parameters specifying orientation, noise level and size. The first 80 patterns of the testing set are exemplars of the feature class in order of increasing contrast, and are generated as discussed above. The following 83 patterns are counter examples (the first 40 were randomly generated and the remaining 43 were randomly chosen from other class examples).

In previous works [2, 3] we discussed the ability of the learning-based operators to correctly estimate the feature class identities and to accurately predict their contrasts. The overall results were promising for all feature classes. There were just a few errors, mainly related to low contrasting features, to prediction errors of the orientation detectors (sepa-

rate module), and to counter examples coming from similar feature classes. However, no comparison of these results with related approaches has been made.

In the experiments that follow we do not consider contrast prediction errors, and focus only on the classification errors. The contrast errors are directly linked to the classification thresholds chosen to be varied in the experiments, thus measuring contrast prediction errors would not allow reaching any significant conclusion.

Error type I (false acceptance) occurs whenever the classifier accepts a non-class sample as belonging to the class. Error type II (false rejection) occurs whenever the classifier considers a member of the class as not belonging to the class. The sum of these two errors gives the total error produced by the classifier and, in our specific case, we want to minimise the sum of these two errors.

We varied the threshold of the learning-based operators from 0.0 to 1.0, in steps of 0.01, and measured the classification errors. Figure 5 shows the results of the threshold experiment for *Edges*, *+Bars*, *+Blobs* and *+Ends*. The graphs for the *-Bars*, *-Blobs* and *-Ends* features were removed to save space, since they were virtually identical to the graphs of their corresponding '+' features.

In order to provide a quantitative evaluation of the relative performance of the method shown here and the model-based approach described in the previous section, we used the same testing sets described above to generate classification statistics (likewise the ones presented in Figure 5) for the model-based approach. Since the classification threshold of the model-based approach is a function of the intensities found in the images, we decided to vary their threshold from 0 to 100 in steps of 1. This yielded exactly 101 sequential thresholds which is compatible with the range of variation used before to test the neural network approach (thresholds varying from 0 to 1 in steps of 0.01). We later discovered that this was a good choice, since the error curves converged near the higher thresholds to the same levels obtained before. Figure 6 shows the errors of the *Edge* and *Bar* operators. The results of the remaining operators are presented on Figure 7.

By contrasting Figure 5 with Figures 6 and 7 it is possible to see that all of the learning-based operators had no type I errors for all tested thresholds, whereas, when using the model-based approach, only *+Ends* had no type I errors.

This excellent type I error performance by the neural networks may be partially explained by the fact that the classification rule decides based on the output of two neurons (one responsible for representing features and the other, non-features) that are mutually exclusive. Moreover, there was a reasonable threshold range (typically between 0 and 0.15) for which the learning-based operators had almost zero errors of type II (optimum performance). No such range could be found in the graphs of the previous approach,

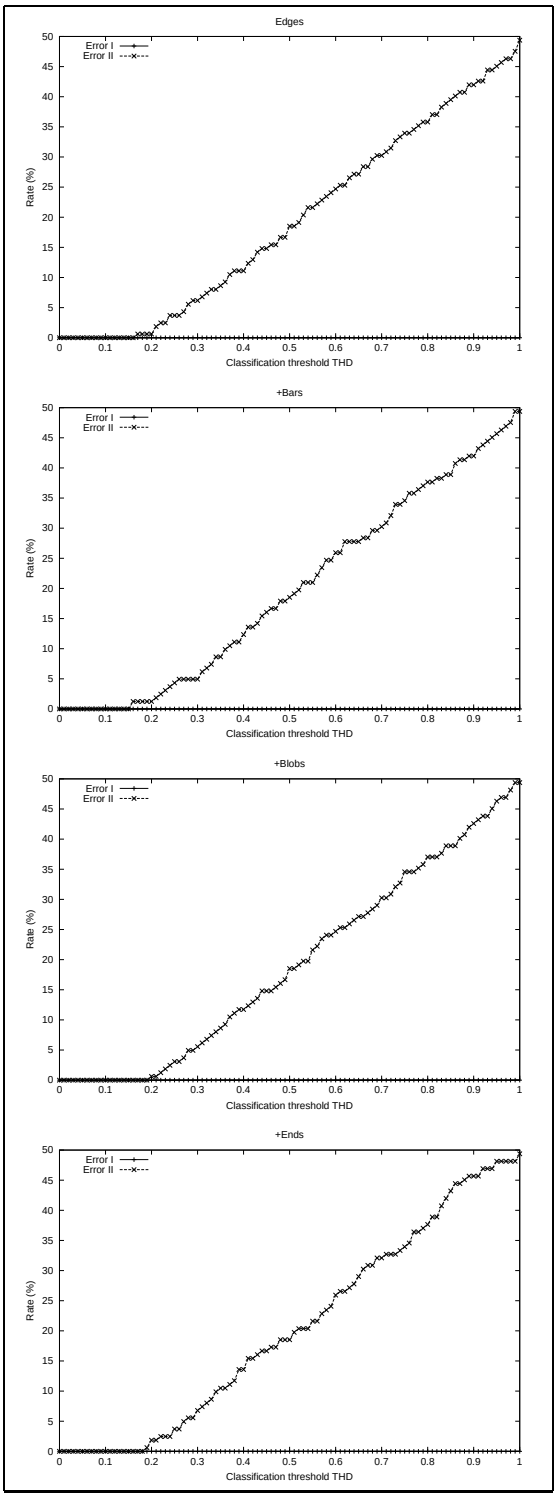

Figure 5. Errors type I and II vs. the classification threshold for *Edges*, *+Bars*, *+Blobs* and *+Ends* for the operators using the new method. The type I error curve allways lies on the horizontal axis.

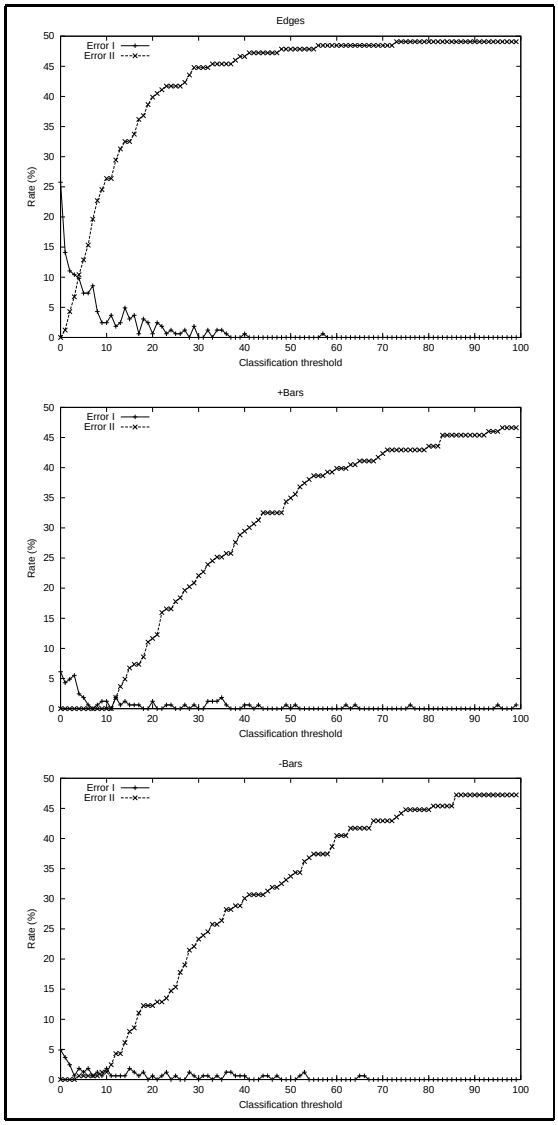

Figure 6. Errors type I and II vs. the classification threshold for the operators using the previous approach of Grove and Fisher [4]. The type I error curve is not allways zero.

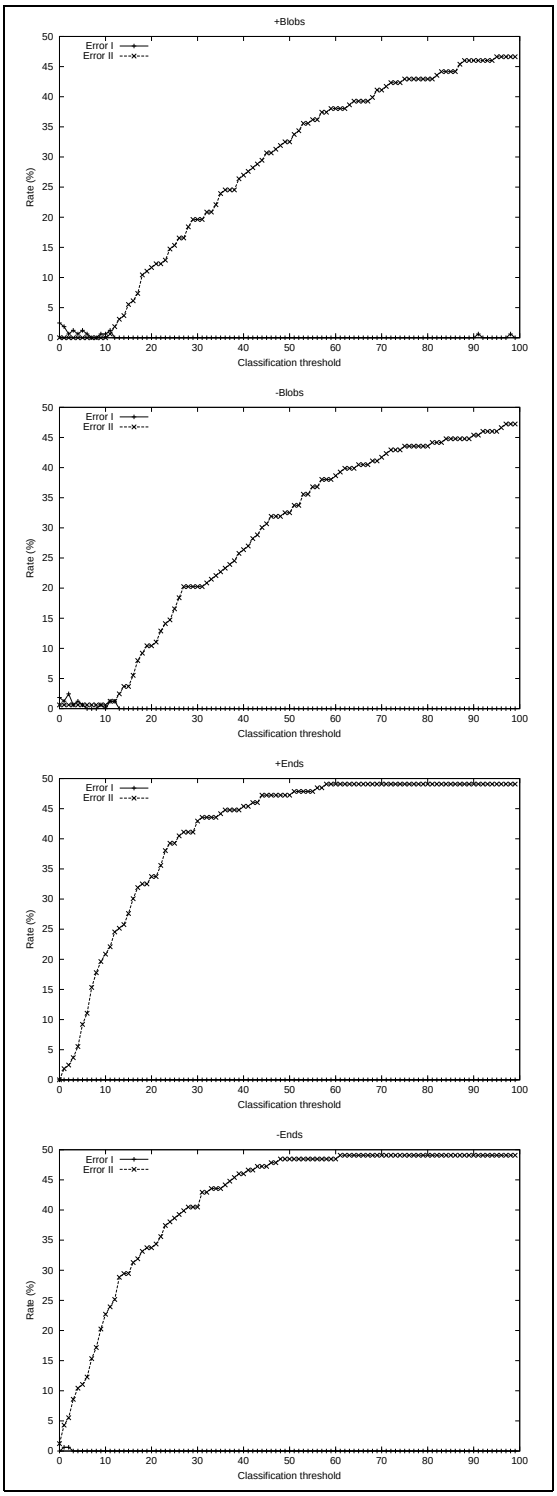

Figure 7. Errors type I and II vs. the classification threshold for the operators using the previous approach of Grove and Fisher [4]. The type I error curve is not allways zero, apart from +*Ends*.

which indicates that the previous approach is more unstable and unreliable.

The above experiments also provided an effective way of determining the best classification thresholds for both approaches. The idea is to choose the highest classification threshold that produces best overall performance. The reason for this is that we want high performance while are not interested in detecting very low contrast features.

From Figure 5 it is easy to see that a 0.15 threshold meets this requirement for the learning-based operators. Moreover, Figures 6 and 7 indicate that a threshold equal to 8, originally (and empirically) determined for the model-based operators, would not be a bad choice since it meets the above requirements for most of the feature types, apart from *Ends*, which would need a nearly 0 threshold, and *Edges*, which would need a threshold equal to 4.

6 Subjective Comparison

In order to help the reader to have a clearer view of the outputs generated by the feature extraction operators described in this paper, and before applying it to any real images, which may contain complicated textures and features not easily spotted by the eye, a number of synthetic images were used. Input images containing horizontal, vertical and diagonal edges at varying contrasts were used to test the outputs of the *Edge* operators of both approaches compared in this paper. The *Bar* operators were tested against images containing vertical, horizontal and diagonal bar patterns. The *Blob* operators were tested against images containing blob-like features at varying sizes and contrasts. Finally, images containing simple polygons at varying contrasts and orientations were used to test the *End* operators (the vertices of these polygons are the features to be detected).

These examples are shown on Figures 8 and 9. From a subjective evaluation of these figures, apart from some small uncertainty with regards to the feature location, it is possible to conclude that the neural classifiers are doing a reasonable job at detecting the features they have been designed for.

Note that, in some of the pictures, the *Edge* and *Bar* operators eventually fail near the centre or near the periphery of the images, which is expected since the scale of an image feature does not always match the scale of the receptive field window. The last columns of Figures 8 and 9 illustrate the results of the previous approach when applied to the same set of synthetic input scenes. Overall, the previous approach had poor contrast sensitivity and a greater number of features not being detected (discontinuities) when compared to the learning-based approach.

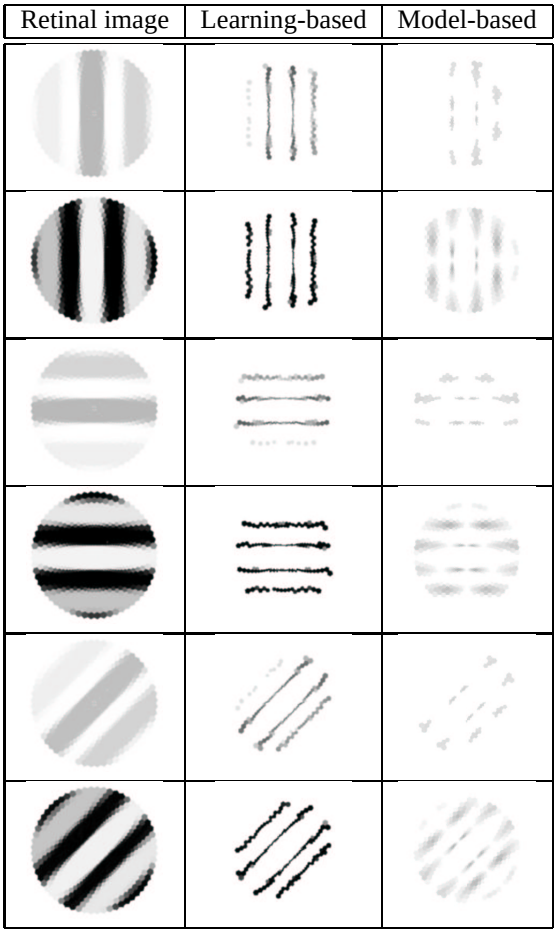

Figure 8. *Edge* **operators on synthetic images.**

7 Conclusions

The main contribution of this paper was to present an objective comparison of two distinct approaches to primal sketch feature extraction from log-polar images. In the first approach [4], features were detected using a number of manually defined logical operators within a fixed retinal window. A second approach [2, 3] took advantage of the inherent learning (from examples) and generalisation properties of neural networks. Instead of designing feature extraction operators from scratch, the basic idea was to use a set of exemplars of features and non-features to train a MLP-backpropagation neural network. If the operators were applied to a Cartesian image, then probably analytic methods would have been possible, as in the Canny edge detector design. However, here the unusual receptive field geometry and sampling computation were sufficiently complicated that we chose the learning-based approach.

Performance evaluation experiments (error type I and II measured under varying classification thresholds) applied

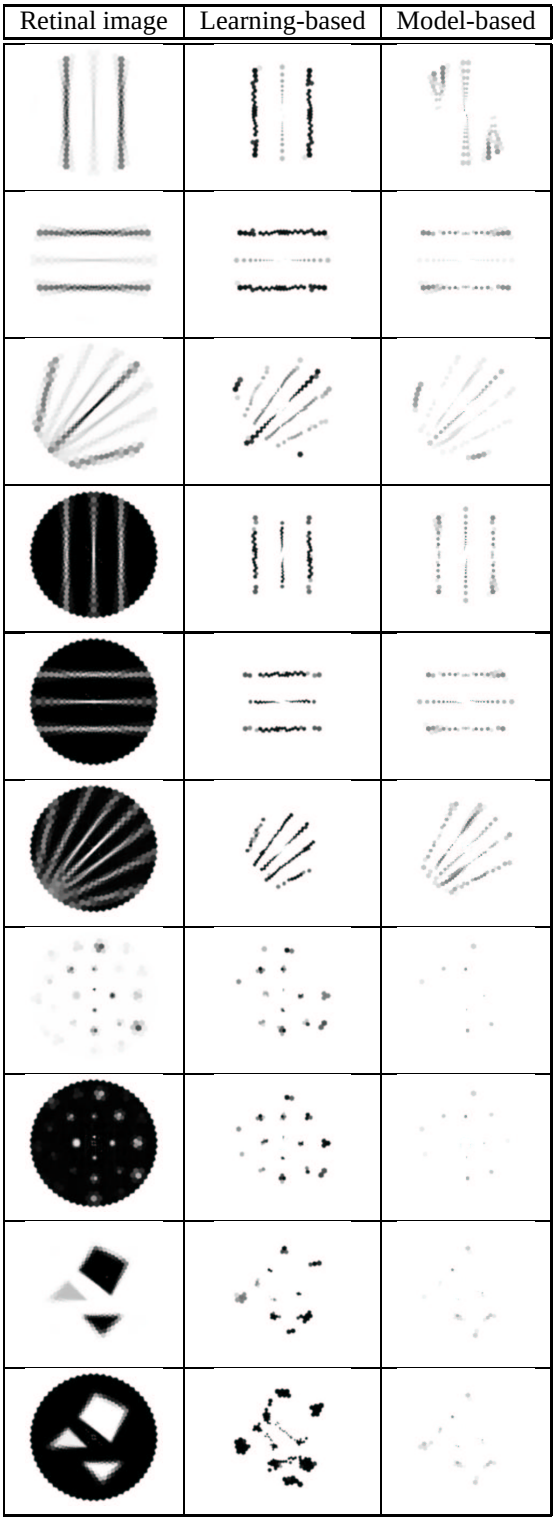

Figure 9. $\pm$*Bar* operators (lines 1-6), $\pm$*Blob* operators (lines 7-8) and $\pm$*End* operators (last two lines), on synthetic images.

both kinds of operators to a set of synthetic features. These experiments were complemented by a subjective evaluation of the operator's visual output on a set of simple scenes, containing mostly features of interest.

As a result of the experiments, it was observed that the new approach (learning-based) presented better results with respect to the number of correctly classified features, provided a richer description for the image data with the addition of an estimate for the feature's contrast, and became a more flexible solution to the problem in the sense that whenever a new feature class is required, only its training set needs to be provided.

Moreover, there was a reasonable threshold range (typically between 0 and 0.15) for which the learning-based operators had practically no errors of types I and II (optimum performance). No such range could be found in the graphs of the previous approach, which indicates that the previous approach is more unstable and unreliable.

References

[1] H. Gomes and R. Fisher. Structural learning from iconic representations. *Lecture Notes in Artificial Intelligence*, 1952:399–408, 2000. Springer.

[2] H. Gomes and R. Fisher. Learning and extracting primal-sketch features in a log-polar image representation. In *Proc. of Brazilian Symposium on Computer Graphics and Image Processing*, pages 338–345. IEEE Computer Society, 2001.

[3] H. Gomes and R. Fisher. Primal sketch feature extraction from a log-polar image. *Pattern Recognition Letters*, 24:983–992, 2003.

[4] T. Grove and R. Fisher. Attention in iconic object matching. In *Proc. of British Machine Vision Conference*, pages 293–302, 1996.

[5] F. Jurie. A new log-polar mapping for space variant imaging: application to face detection and tracking. *Pattern Recognition*, 32:865–875, 1999.

[6] F. Lim, G. West, and S. Venkatesh. Use of log polar space for foveation and feature recognition. *IEE Proc. of Vision, Image and Signal Processing*, 144(6):323–331, Dec 1997.

[7] D. Marr. *Vision*. W. H. Freeman and Co., 1982.

[8] A. Rojer and E. Schwartz. Design considerations for a space-variant visual sensor with complex-logarithmic geometry. In *Proc. of Int. Conference on Pattern Recognition*, pages 278–285, 1990.

[9] G. Sandini and M. Tristarelli. Vision and space-variant sensing. In H. Wechsler, editor, *Neural Networks for Perception*, volume 1, chapter II.11, pages 398–425. Academic Press, 1992.

[10] E. Schwartz. Spatial mapping in primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.

[11] Y. Sun and R. Fisher. Object-based visual attention for computer vision. *Artificial Intelligence Journal*, 146(1):77–123, May 2003.

[12] S. Wilson. On the retino-cortical mapping. *Int. Journal on Man-Machine Studies*, 18:361–389, 1983.