USE OF PREDICTIVE CODING DISTRIBUTION FOR EDGE DETECTION

Bogdan Cramariuc, Ioan Tăbuş and Moncef Gabbouj

Signal Processing Laboratory
Tampere University of Technology
P.O. Box 553, SF-33101 Tampere, Finland
e-mail: {crama,tabus,moncef}@cs.tut.fi

ABSTRACT

The purpose of the present paper is to analyze how the predictive distribution estimated using a set of context dependent nonlinear adaptive predictors can be used to localize edges in graylevel images. Since the adaptive predictors have the potential of learning repetitive structure, as those characteristic to certain textures, our predictive edge detection scheme may be a practical way to conceal the relative high contrast of certain texture regions.

1. INTRODUCTION

Predictive coding methods are in one form or another the most efficient and widely used techniques for lossless image compression, which is certainly due to the dramatic reduction in the entropy of the prediction error image compared to the original image. What have been perceived as the main weakness of predictive modelling is the failure to adequately predict the image near and at edges. Recent developments of predictive methods cope with this difficulty by interleaving the adaptive linear or nonlinear prediction with context modelling[1][2]. One such technique,[2], uses the estimates of directional gradients in a causal neighborhood of the current pixel, as selectors between several predictors, designed to be optimal for certain edge orientations. In this sense, a rudimentary edge detection was shown to improve prediction and coding performances.

The purpose of the present paper is to analyze the reciprocal problem: how the predictive distribution estimated using a set of context dependent nonlinear adaptive predictors can be used to localize edges in graylevel images. The motivation of this study is multifolded.

First, we may assume the following realistic scenario: for some high level vision tasks it is required to perform edge detection of images which are received in a lossless encoded format; when decoding the image, the predictive distribution becomes available at no cost for the edge detection task.

Second, the adaptive predictors have the potential of learning repetitive structure[3, 4], as those characteristic to certain textures. A two-stage processing edge detection scheme may be a practical way to conceal the relative high contrast of texture regions, provided that textures belong to what we call the kernel of the nonlinear predictor based edge detector. We are not able yet to analytically characterize such classes of textures, but our experiments have shown that surprisingly many natural textures belong to the kernel of our new edge detectors.

The cost of the procedure we propose is extremely low as processing time (especially in the above presented scenario) and therefore it may be seen as an ideal initialization step for more elaborated iterative edge detectors based on stochastic models[5]. The technique can also be extended to the detection of the *line process*[6] which can be used as an initial step in conjunction with Markov random field model based image restoration.

Finally, the use of predictive coding distribution offers the information needed to progress from edge decision based on simply thresholding the error image (where the threshold selection process is highly dependent on the local contrast) to a probabilistic setting based on confidence intervals.

While conceptually the structure we introduce is intuitive and simple, the selection of the proper features of the adaptive prediction procedure and of context modelling are essential in obtaining the desired edge detection properties. Our goal is therefore to present simple and effective solutions for the problem of *local edge detection.*

Subsequently processing the local edge image using more elaborate tools such as boundary detection, segmentation, region growing based on this local edges will certainly be able to improve the edge detection as required by the high level vision tasks.

2. PREDICTIVE EDGE DETECTION

We present a simple edge detection method, which runs an adaptive linear prediction algorithm in one pass through the image, where the switching between different linear predictors is controlled by a context selection algorithm similar to the one described in [2] (but with a major difference, namely we are using adaptive filters for prediction instead of fixed filters). A rough edge image is first obtained by estimating the directional gradients (in a causal neighborhood) and thresholding them against predefined fixed values. A pixel declared edge candidate at this stage will be further checked if it may be well predicted by one of few well adapted predictors. Only if the prediction error is large enough, the pixel will be declared edge pixel. The thresholds corresponding to each context can be maintained at fixed values, obtained in an initial training stage, or can be allowed to vary during the edge detection process, according to some heuristically established rules.

The prediction window contains nine pixels: $\underline{D} = [D_n, D_w, D_{nw}, D_{ww}, D_{nn}, D_{ne}, D_{nen}, D_{nnw}, D_{nww}]$, but for the computation of the directional gradients we use D_{nnww}, D_{nnee} as well. The directional gradients (horizontal, vertical, diagonal at 45°, diagonal at 135°) are estimated as follows:

$$\begin{aligned}
d_h &= |D_{ww} - D_w| + |D_{nw} - D_n| + |D_n - D_{ne}| \\
d_v &= |D_{nw} - D_w| + |D_{nn} - D_n| + |D_{nen} - D_{ne}| \\
d_{45^\circ} &= |D_w - D_{wwn}| + |D_{nw} - D_{nnww}| + \\
&\quad + |D_n - D_{nnw}| \qquad (1) \\
d_{135^\circ} &= |D_{ne} - D_{nnee}| + |D_n - D_{nen}| + |D_n - D_w|
\end{aligned}$$

The contexts we have used are defined using conditional logic statements (similar to those used in the prediction stage of [2])

$$\begin{aligned}
&\text{IF}(d_v + d_h > 32)\{\text{sharp edge}\} \\
&\quad \{con = 1;\} \\
&\text{ELSE IF}(d_v - d_h) > 12)\{\text{horizontal edge}\} \\
&\quad \{con = 2;\} \\
&\text{ELSE IF}(d_h - d_v) > 12)\{\text{vertical edge}\} \\
&\quad \{con = 3;\} \\
&\text{ELSE}\{\text{smooth area}\} \\
&\quad \{con = 0;\} \\
&\text{IF}(d_{45^\circ} - d_{135^\circ} > 32)\{\text{sharp } 135^\circ \text{ edge}\} \\
&\quad \{con = con + 4;\} \\
&\text{IF}(d_{135^\circ} - d_{45^\circ} > 32)\{\text{sharp } 45^\circ \text{ edge}\} \\
&\quad \{con = con + 8;\} \qquad (2)
\end{aligned}$$

The context $con = 0$ corresponds to smooth areas in the images, while all other contexts, $con \in \{1, \ldots, 11\}$, are specific for different edge orientations. We use 12 adaptive predictors, one for each context. At time t the context index, con, is determined with (2) and then the prediction $\underline{\hat{D}_t}$ and the prediction error ε_t are computed using the parameter vector $\underline{w}_{con}$:

$$\begin{aligned}
\hat{D}_t &= \underline{w}_{con}^T \underline{D}_t \\
\varepsilon_t &= D_t - \hat{D}_t
\end{aligned}$$

The parameter vector corresponding to the context con is then updated using the recursive Least Squares algorithm with a forgetting factor[7]. The procedure for edge detection is summarized in Table 1.

3. EXPERIMENTAL RESULTS

We focus on illustrating the following feature of our new edge detectors: concealing the contrast changes inside the natural texture, while remaining active for nonpredictable intensity changes.

First are shown the original images, where some regular geometrical shapes are placed in a textured background. The edge images obtained when using Canny edge detector[8] are shown to capture spurious contours from the textured background in the first example, while in the second example the edges of some objects are not found. Simple edge detectors belonging to our class of methods are shown to successfully detect the boundary of the objects, while concealing the contours from the textured region.

4. REFERENCES

[1] M. Weinberger, J. Rissanen, and R. Arps. Applications of universal context modeling to lossless compression of gray-scale images. *IEEE Transactions on Image Processing*, IP-5:575–586, Apr. 1996.

[2] X. Wu, N.D. Memon, and K. Sayood. A context-based, adaptive, lossless/nearly-lossless coding scheme for continuous-tone images. working document ISO/IEC/SC29/WG1/N256, ISO, 1995.

1 *Initialize the RLS procedures:* $\underline{w}_{con}(0) = 0$;, $P_{con}(0) = 0.1I$ *for all* $con = 0, \ldots, 11$.
2 *Iterate for each pixel* D_t *in the image*
  2.1 *Context selection*
    2.1.1 *Compute the estimates (1) of the directional gradients* d_h, d_v, d_{45° *and* d_{45°.
    2.1.2 *Select the context index, con, using (2).*
  2.2 *Compute the prediction* $\hat{D}_t$ *using the adaptive predictor with parameters* $\underline{w}_{con}$,
  $\hat{D}_t = \underline{w}_{con}^T \underline{D}_t$.
  2.3 *Update the parameters* $\underline{w}_{con}$ *of the con-th predictor using the RLS procedure with exponential forgetting parameter* λ.
  2.4 *Threshold the prediction error* $\varepsilon_t = D_t - \hat{D}_t$ *at a context dependent threshold level* θ_{con}.

Table 1: Procedure for edge detection

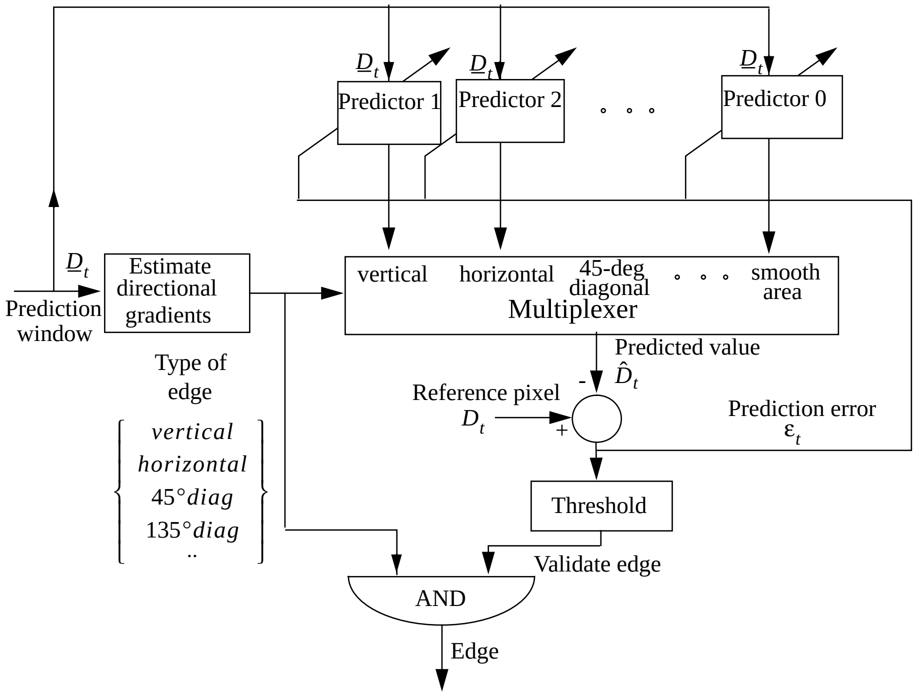

Figure 1: Edge detection based on adaptive context prediction residuals

[3] P. Ffrench and J.R. Zeidler anf W.H. Ku. Enhanced detectability of small objects in correlated clutter using an improved 2-D adaptive lattice algorithm. *IEEE Transactions on Image Processing*, 6:383–397, 1997.

[4] T. Soni and J.R. Zeidler anf W.H. Ku. Performance evaluation of 2-D adaptive prediction filters for detection of small objects in image data. *IEEE Transactions on Image Processing*, 2:327–339, 1993.

[5] M. Petrou, P. Papachristou, and J. Kittler. Error propagation analysis for edge postprocessing. In *Proc. ICIP-96, International Conference on Image Processing*, pages 861–864, Lausanne, Switzwerland, Sept. 1996.

[6] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[7] S. Haykin. *Adaptive filters theory.* Prentice Hall, Upper Saddle River, 1996.

[8] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698, Nov. 1986.

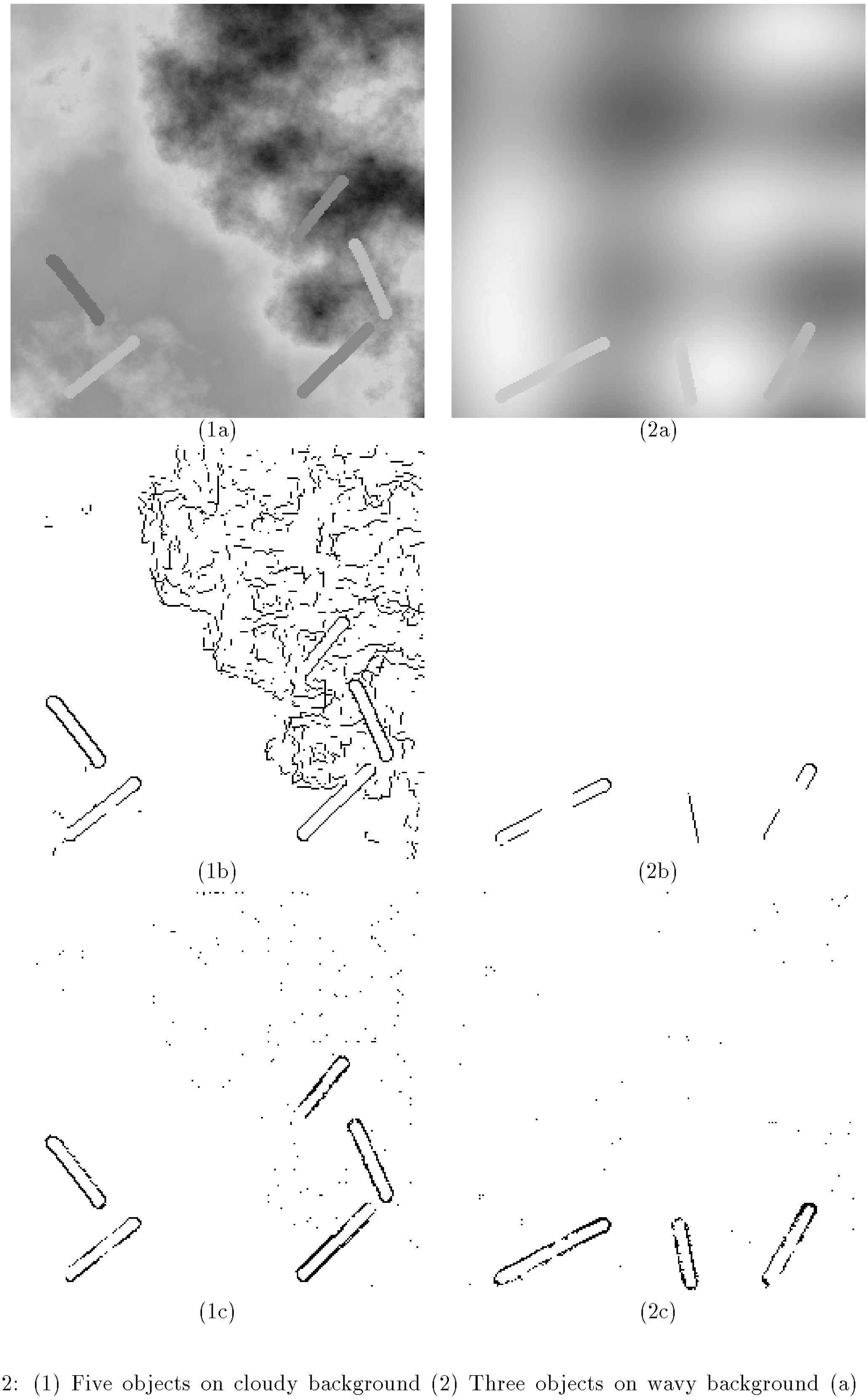

Figure 2: (1) Five objects on cloudy background (2) Three objects on wavy background (a) Original image (b) Result of Canny edge detector (c) Result of the proposed method