A method for using marginal statistics for image denoising

Ruth Bergman, Yacov Hel-Or, Hila Nachlieli, Gitit Ruckenstein
HP Laboratories Israel
HPL-2004-201(R.1)
August 11, 2006*

image denoising, co-occurrence matrix, salt and pepper noise, image statistics, universal denoiser

This invention is concerned with a method for removing noise (denoising) from gray level images using statistics that model the image textures. The denoising method presented here uses the framework of the general denoising algorithm published in [1]. The algorithm in [1], known as the Discrete Universal DEnoising algorithm (DUDE), has been proved to be an optimal algorithm for data denoising in certain settings. However, when the input is a noisy gray-level image of a finite size, the sufficient conditions for optimality are not satisfied. A major difficulty in implementing DUDE for gray-level images, as opposed to binary images, lays in the high space complexity required for collecting block statistics. Furthermore, since these statistics are very sparse they do not estimate the corresponding probabilities very well.

We suggest here a method to approximate the image block statistics required for DUDE by using a gray level co-occurrence matrix. Such a matrix has been used, for example in [2], to model the statistics of textured images.

* Internal Accession Date Only
Approved for External Publication

A method for using marginal statistics for image denoising

Ruth Bergman, Yacov Hel-Or, Hila Nachlieli, Gitit Ruckenstein *
HP Laboratories Israel, Haifa, Israel.

1 Introduction

This invention is concerned with a method for removing noise (denoising) from gray level images using statistics that model the image textures. The denoising method presented here uses the framework of the general denoising algorithm published in [1]. The algorithm in [1], known as the Discrete Universal DEnoising algorithm (DUDE), has been proved to be an optimal algorithm for data denoising in certain settings. However, when the input is a noisy gray-level image of a finite size, the sufficient conditions for optimality are not satisfied. A major difficulty in implementing DUDE for gray-level images, as opposed to binary images, lays in the high space complexity required for collecting block statistics. Furthermore, since these statistics are very sparse they do not estimate the corresponding probabilities very well.

We suggest here a method to approximate the image block statistics required for DUDE by using a gray level co-occurrence matrix. Such a matrix has been used, for example in [2], to model the statistics of textured images.

2 Model

The model in which [1] is applied is illustrated in Figure 1. The model consists of the following components.

- The signals coming out of the source are represented by random variables X_i that take their values x_i from a discrete alphabet $\mathcal{A}$, i.e., a finite set of symbols. The signals that come out of the noisy channel are represented by random variables Z_i that take their values z_i from the same discrete alphabet. The alphabet in our case consists of the gray scale values, typically $\mathcal{A} = \{0, 1, \ldots, 255\}$.

*E-mail: {ruth.bergman, yacov.hel-or, hila.nachlieli, gitit.ruckenstein}@hp.com

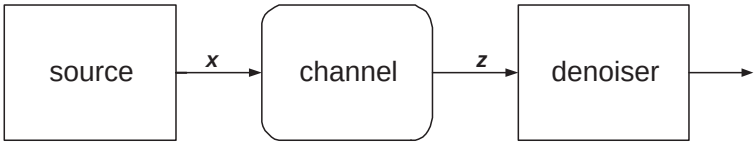

Figure 1: Model

- No assumptions are made on the source distribution. This is why the denoiser is considered universal.
- On the other hand, the error probability $\Pr(\mathbf{Z} = \mathbf{z} \,|\, \mathbf{X} = \mathbf{x})$, namely the channel distribution, is assumed to be known.
- The channel is memoryless, namely

$$\Pr(Z_i = z_i \,|\, X_1 X_2 \cdots X_i = x_1 x_2 \cdots x_i) \;=\; \Pr(Z_i = z_i \,|\, X_i = x_i)\,.$$

  The channel can therefore be compactly described by an $M \times M$ transition probability matrix Π, where $\Pi_{i,j} = \Pr(Z = \alpha_j \,|\, X = \alpha_i)$, in which $\alpha_i, \alpha_j \in \mathcal{A} = \{\alpha_1 \cdots \alpha_M\}$. The channel probability matrix is assumed to be non-singular, hence Π^{-1} exists.
- A cost function $\Lambda : \mathcal{A} \times \mathcal{A} \to \Re$ is defined to represent the error measure of estimating one alphabet symbol by another one.

If the channel output distribution were known, then the channel input distribution, namely the source distribution, could be computed by inverting the matrix Π. Specifically, let $\mathbf{P}_X$ denote the column vector of input probabilities, where the ith component is $\Pr(X = \alpha_i)$, and let $\mathbf{P}_Z$ denote the column vector of output probabilities. If Π^{-T} is the transpose of Π^{-1}, then

$$\mathbf{P}_X \;=\; \Pi^{-T}\,\mathbf{P}_Z\,. \tag{1}$$

Though the channel output distribution is generally not assumed to be known, it can be empirically estimated from the data. Therefore, the estimation of $\mathbf{x}$ by $\hat{\mathbf{x}}$ is based on the given channel matrix Π, the statistics gathered from the observed sequence $\mathbf{z}$, and the cost function $\Lambda()$.

3 DUDE definitions and notations

The algorithm performs two passes over the data. The required statistics of the observed sequence $\mathbf{z}$ are computed in the first pass, while the denoiser output sequence $\hat{\mathbf{x}}$ is generated in the second pass.

First pass: learning data statistics

In this framework the input data is assumed to be a realization of a stationary Markovian process. Hence, a sliding window is chosen whose set of indexes, corresponding to the symbols in the window, is denoted by K. The window size $|K|$ and the window shape are chosen a-priori according to space/complexity limitations. The random multi-variable representing a window symbols is denoted by $\mathbf{Z}_K$. The values of $\mathbf{Z}_K$ are taken from the alphabet $\mathcal{A}$. One of the indexes in K, say index 0, defines the *center* of the window, and its corresponding random variable is denoted by Z_0. The set of remaining indexes, i.e., $\mathbf{c} = K \setminus 0$, are defined as the window *context*, and their random variable is defined as $\mathbf{Z_c}$. For every possible context $\mathbf{Z_c} = \mathbf{z_c}$ of size $|K|-1$, we count the number of appearances of each alphabet symbol α_i, $i = 1, 2, \ldots, M$, as a central symbol. This number is denoted here by $\mathbf{m}(\mathbf{z_c})[i]$. The corresponding column vector of size M is denoted $\mathbf{m}(\mathbf{z_c})$.

Second pass: denoising

To estimate a symbol in the original sequence $\mathbf{x}$, we fix a window of size $|K|$ around the respective symbol z_0 in $\mathbf{z}$.

The minimum loss is attained if z_0 is replaced by the symbol β which minimizes the expression

$$\sum_{i=1}^{M} \Lambda(\alpha_i, \beta) \cdot \Pr(X_0 = \alpha_i \,|\, \mathbf{Z} = \mathbf{z}) \,, \tag{2}$$

where Λ is a cost function.

Suppose $Z_0 = z_0$ appears with the context $\mathbf{Z_c} = \mathbf{z_c}$. The probabilities in (2) are approximated by

$$\Pr(X_0 = \alpha_i | Z_0 = z_0, \mathbf{Z_c} = \mathbf{z_c}) = \frac{\Pr(Z_0 = z_0 | X_0 = \alpha_i) \cdot \Pr(X_0 = \alpha_i | \mathbf{Z_c} = \mathbf{z_c})}{\Pr(Z_0 = z_0 | \mathbf{Z_c} = \mathbf{z_c})} \tag{3}$$

Let $\mathbf{P}_{Z_0 \,|\, \mathbf{z_c}}$ denote the column vector whose ith component is $\Pr(Z_0 = \alpha_i \,|\, \mathbf{Z_c} = \mathbf{z_c})$. In a similar manner, we define $\mathbf{P}_{X_0 \,|\, \mathbf{z_c}}$. Due to the memorylessness of the channel,

$$\mathbf{P}_{X_0 \,|\, \mathbf{z_c}} \;=\; \Pi^{-T} \, \mathbf{P}_{Z_0 \,|\, \mathbf{z_c}} \;.$$

Assuming that $z_0 = \alpha_j$, Equation (3) can be re-written as

$$\frac{\Pi_{i,j} \, [\Pi^{-T} \, \mathbf{P}_{Z_0 \,|\, \mathbf{z_c}}]_i}{[\mathbf{P}_{Z_0 \,|\, \mathbf{z_c}}]_j} \;. \tag{4}$$

The channel matrix Π is assumed to be known. The statistics collected in the vector $\mathbf{m}(\mathbf{z_c})$ are used to estimate the probability vector $\mathbf{P}_{Z_0,\mathbf{z_c}}$. This is equivalent here to estimating $\mathbf{P}_{Z_0\,|\,\mathbf{z_c}}$, since the minimization is taken over the various values β that replace z_0 while the context $\mathbf{c}$ is fixed.

4 Estimation of data prior using marginal statistics

An estimation of the post channel distribution $\mathbf{P}_{Z_0,\mathbf{z_c}}$ is required in the second phase of DUDE. The original version of DUDE uses the statistics $\mathbf{m}(\mathbf{z_c})$ in order to estimate $\mathbf{P}_{Z_0,\mathbf{c}}$. Although collecting the post-channel statistics of a binary image is a feasible task, in the case of gray-level images this task is problematic; Due to the finite size of the output image, and the high dimensionality of $\mathbf{m}(\mathbf{z_c})$ the collected statistics are expected to be very sparse and thus unreliable.

In this report we suggest to overcome this problem by estimating the post channel distribution $\mathbf{P}_{Z_0,\mathbf{z_c}}$ using the statistics of its 2D marginals. This approach has been used successfully in the past for texture analysis and synthesis, using the so called *co-occurrence* statistics [2, 4].

4.1 Gray level co-occurrence matrix

Let I be an image of size $L_x \times L_y$ in which pixel values are taken from a set G of gray levels. The image I can be written as a function $I : \{1,\ldots,L_x\} \times \{1,\ldots,L_x\} \to G$. Let Δ be the set $\Delta = \{-\delta,\ldots,0,\ldots,\delta\}$ for some fixed positive integer δ smaller than $\min(L_x, L_y)$. We define a $|G| \times |G| \times |\Delta| \times |\Delta|$ four-dimensional matrix $C = C(I)$ known as the gray-level co-occurrence (GLC) matrix of I. The element $C_{\alpha,\beta,\delta_x,\delta_y}$, where $\alpha,\beta \in G$ and $\delta_x,\delta_y \in \Delta$, is the number of different coordinates (x,y) for which

$$\begin{aligned} I(x,y) &= \alpha\,, \\ I(x+\delta_x, y+\delta_y) &= \beta\,. \end{aligned}$$

The matrix C may be normalized so that

$$\sum_{\alpha\in G}\sum_{\beta\in G} C_{\alpha,\beta,\delta_x,\delta_y} = 1\,, \quad \forall \delta_x,\delta_y$$

defining a 2D probability distribution for every (δ_x,δ_y). By taking I to be a large image and Δ to be a small set of indices, representing a small fixed-size window, we reduce the sparsity of the statistics gathered in C. The information contained in a GLC matrix includes 2D marginals of the joint distribution of the image windows whose size is $|\Delta| \times |\Delta|$. These marginals have been shown to capture the essence of the statistical characterizations of a textured image. GLC matrices have been therefore used in the past to perform texture analysis and texture synthesis, as in [2, 4].

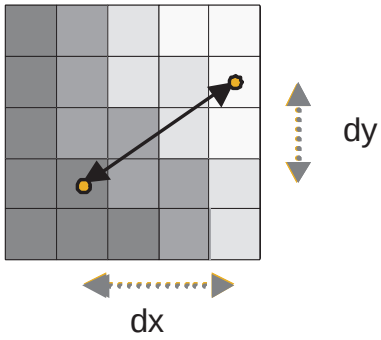

Figure 2: Example of a gray level pair stored in a co-occurrence matrix.

4.2 Estimating post-channel probabilities via co-occurrence

Consider the post-channel probabilities $[\mathbf{P}_{Z_0,\mathbf{z_c}}]_i = \Pr(Z_0 = \alpha_i, \mathbf{Z_c} = \mathbf{z_c})$ we aim to estimate. Applying Bayse rule, we write

$$\Pr(Z_0 = \alpha_i, \mathbf{Z_c} = \mathbf{z_c}) \;=\; \Pr(\mathbf{Z_c} = \mathbf{z_c} \,|\, Z_0 = \alpha_i) \cdot \Pr(Z_0 = \alpha_i) \tag{5}$$

Let $\{Z_r\}_{r\in\,\mathbf{c}}$ denote the set of random variables corresponding to the values of the context $\mathbf{z_c} = (z_1, \ldots, z_{|K|})$. Suppose that these random variables are conditionally independent given Z_0, namely

$$\Pr(\mathbf{Z_c} = \mathbf{z_c} \,|\, Z_0 = \alpha_i) \;=\; \prod_{r\in\,\mathbf{c}} \Pr(Z_r = z_r \,|\, Z_0 = \alpha_i) \tag{6}$$

We can thus replace Equation (5) by

$$\begin{array}{l} \prod_{r\in\,\mathbf{c}} \Pr(Z_r = z_r \,|\, Z_0 = \alpha_i) \cdot \Pr(Z_0 = \alpha_i) \;\; = \\ \prod_{r\in\,\mathbf{c}} \left\{ \frac{\Pr(Z_r = z_r\,, Z_0 = \alpha_i)}{\Pr(Z_0 = \alpha_i)} \right\} \cdot \Pr(Z_0 = \alpha_i)\,. \end{array} \tag{7}$$

The probabilities $\Pr(Z_r = z_r\,, Z_0 = \alpha_i)$ are naturally estimated by the statistics contained in the co-occurrence matrix, given that the relative position set Δ covers the required window. The probabilities $\Pr(Z_0 = \alpha_i)$ are estimated simply by gray level statistics collected from the data.

In general, the independence assumption (6) does not hold for images because of two-dimensional correlations and the structure of blocks. It turns out, nevertheless, that the approximation in (7) allows a rather good denoising. In order to improve the probability estimation, we refine the assumption (6) by incorporating an additional descriptor of the window joint distribution. Let $\sigma(\mathbf{Z}_K)$ denote the standard deviation of the random variables in $\mathbf{Z}_K$, and let $\sigma = \sigma(\mathbf{z}_K)$ be the standard deviation computed for a given window $\mathbf{z}_K$:

$$\sigma(\mathbf{z}_K) = \frac{1}{|K|} \cdot \sqrt{\sum_{r\in K} (z_r - \bar{z})^2}\;,$$

where $\bar{z}$ is the average of gray level values in the window. We then replace (6) by the assumption

$$\Pr(\mathbf{Z_c} = \mathbf{z_c} \,|\, Z_0 = \alpha_i \,, \sigma(\mathbf{Z}_K) = \sigma) \;=\; \prod_{r \in\, \mathbf{c}} \Pr(Z_r = z_r \,|\, Z_0 = \alpha_i \,,\, \sigma(\mathbf{Z}_K) = \sigma) \,.$$

We then get

$$\begin{aligned} \Pr(Z_K = \mathbf{z}_K) &= \Pr(Z_0 = \alpha_i \,, \mathbf{Z_c} = \mathbf{z_c} \,, \sigma(\mathbf{Z}_K) = \sigma) \\ &= \prod_{r \in\, \mathbf{c}} \left\{ \frac{\Pr(Z_r = z_r \,, Z_0 = \alpha_i, \sigma(\mathbf{Z}_K) = \sigma)}{\Pr(Z_0 = \alpha_i, \sigma(\mathbf{Z}_K) = \sigma)} \right\} \\ &\quad \cdot \Pr(Z_0 = \alpha_i, \sigma(\mathbf{Z}_K) = \sigma) \,. \end{aligned} \tag{8}$$

In order to estimate the probabilities in (8), we change the structure of the co-occurrence matrix constructed in the statistics collection phase. A fifth dimension is added to the co-occurrence matrix as representing the standard deviation of the block in which a specific pair of values is observed. We quantize the values of the standard deviation in order to achieve a finite set. The matrix element $C_{\alpha,\beta,\delta_x,\delta_y,\sigma}$ stands for the number of times that the values (α, β) are located in a relative distance (δ_x, δ_y) inside a window of (quantized) standard deviation σ.

5 Implementation issues

The implementation currently supports salt and pepper noise on gray images. This section describes some of the implementation details that are necessary for the algorithm.

5.1 Channel definition

For 256 gray levels 0-255, the channel transition matrix Π for a salt and pepper noise is defined by

$$\Pi_{i,j} = \begin{cases} 1-p & (i=j) \wedge (i,j \notin \{0,255\} \\ p/2 & i \neq j \wedge (j \in \{0,255\}) \\ 1-p/2 & (i=j=0) \vee (i=j=255) \\ 0 & \text{otherwise} \end{cases}$$

where $p < 1/2$ is the probability of switching from any color to either salt or pepper. The transition matrix we obtain from this channel definition is invertible. This channel is, therefore, easy to work with in DUDE.

5.2 Cost function definition

Following is a short description of the cost function $\Lambda()$ we used in our implementation. The cost of estimating α by β $(\alpha, \beta \in \{0, \ldots, 255\})$ increases as the absolute difference between

the two values increases. The increase is defined by a gaussian centered on α with a standard deviation of 15.

5.3 Co-occurrence matrix implementation

The first step of the algorithm computes the co-occurrence matrix as counts. The window size used in our experiments was 3×3 pixels. Our specific implementation collects joint co-occurrence counts enhanced with the contrast feature (standard deviation of the neighborhood), as described in Section 4. Typically we used four quantization levels of the contrast feature, where the actual lines of demarcation between bins was computed adaptively so that each bin contains the same number of data points.

Based on the co-occurrence counts, we compute the conditional probabilities required by (8). Due to insufficient data, some entries in the matrix have a value of zero where, based on the nearby values, the corresponding probability should not be zero. To overcome this problem we interpolate the "missing counts" in the co-occurrence matrix from the neighboring relevant counts. The probability $\Pr(Z_0 = \alpha_i, \sigma(\mathbf{Z}_K) = \sigma)$ in (8) is estimated by summing up the relevant entries in the post-interpolation co-occurrence matrix.

Another problem in the computation of (8) arises from multiplying the directional conditional probabilities together to obtain the overall conditional probability of the context. When one directional conditional probability is zero for any gray level, it sets the overall conditional probability to zero for that gray level. Furthermore, when we multiply together many very small probabilities we quickly exceed the floating point accuracy and again values are set to zero. Our first experiments did not successfully denoise because of this problem. To avoid this problem we threshold the (directional) probability values by a small minimal probability (typically 0.005). We then move the entire computation to a more stable range by dividing by that same value. (That is, the minimal probability value is 1, and the maximum value is 1/.005.)

6 Results

This implementation successfully cleans up high levels of salt and pepper noise, as the example in Figures 3 and 4 shows. Note that we do not compute any mathematical measure to grade the quality of denoising, but we look for visually pleasing output images.

7 Future Work

Describing the pixel neighborhood (context) by the gray values of the neighboring pixels is one option out of many possible transformations. In future work, we will find the optimal representation, such that the correlation between the middle pixel to each context variable is maximal, using the Canonical Correlation Analysis (CCA).

To assess the validity of our approach, we plan to apply it to other noisy channels, and particularly to the Gaussian channel, which is assumed in many image denoising schemes.

8 References

[1] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, M. Weinberger, *Universal discrete denoising: Known channel,* HPL-2003-29.

[2] A. C. Copeland, G. Ravichandran, M. M. Trivedi, *Texture synthesi using gray-level co-occurrence models: algorithms, experimental analysis, and psychophysical support, Optical Engineering,* vol. 40, 2001, 2655-2673.

[3] R. C. Gonzalez, R. E. Woods, *Digital Image Processing,* 1993.

[4] T. Malzbender and S. Spach *A Context Sensitive Texture Nib, Proc. of Computer Graphics International*, June 1993.

Figure 3: An image with 30% salt & pepper noise.

Figure 4: The denoised image.