Proc. of SPIE Vol. 1569, Stochastic and Neural Methods in Signal Processing, Image Processing, and Computer Vision, ed. S Chen (Oct 1991) Copyright SPIE

A Bayesian Matching Technique for Detecting Simple Objects in Heavily Noisy Environment

John S. Baras and Emmanuil N. Frantzeskakis

Electrical Engineering Department and Systems Research Center,
University of Maryland, College Park, MD 20740

ABSTRACT

The template matching problem, for binary images corrupted with spatially white, binary, symmetric noise, is studied. We compare matching based directly on the pixel-valued image data as well as on data coded by two simple schemes: a modification of the Hadamard basis and direct coarsening of resolution. Bayesian matching rules based on M-ary hypothesis tests are developed. The performance evaluation of these rules is provided.

The paper presents a study of the trade-off between the quantization level and the ability of detecting an object in the image. This trade-off depends on the (external) noise generated at the moment we receive the uncoded image.

The sum-of-pixels and the histogram statistics are introduced in order to reduce the computational load inherent in the correlation statistic, with the resulting penalty of a higher probability of false alarm rate.

In the present work we demonstrate by examples that it is beneficial for recognition to combine an image coding technique with the algorithm extracting some "basic" information from the image. In other words coding (for compression) helps recognition. Numerical results illustrate this claim.

1. INTRODUCTION

Consider the following simple instance of the Template Matching Problem: A binary image represents an $m \times m$-pixel black square in white background. The image is transmitted over a noisy channel causing several pixel inversions. We want to find the $m \times m$ windows on the image containing part or all of the black square. For this purpose we may scan the noise corrupted image, correlate with a full black $m \times m$ template and decide "black square detected" if the correlation is higher than a certain threshold. We model the noise effect by saying that the image is corrupted by White (in space) Binary Symmetric Channel (BSC) Noise with some inversion probability ϵ. We use the hypothesis testing framework to develop a Bayesian matching test and more specifically determine a threshold value that minimizes the Bayes cost. The computational complexity for designing this test is reduced by using the *sum-of-pixels statistic*. The analytical performance evaluation we obtain gives us the opportunity to compare the matching rules based on a sequence of instances of the same image but on different resolutions.We conclude that compression of noise corrupted data, via coding, may "kill part of the noise" providing thus more reliable data for object detection.

A Bayesian matching rule, which is applied directly to block coded image data[1,2,3,4] is also developed. The coding scheme we introduce is inspired by the usage of the Hadamard basis in image coding[4] and it encodes binary image data with a *modified Hadamard basis*. Here we use the *histogram statistic* to reduce the computational complexity of the matching rule with a penalty on the reliability of the rule. Nevertheless, with this technique we can detect quite robustly some class of simple binary objects in a heavily noisy environment.

In section 2 we show how the template matching problem and the problem of matching based on the sum-of-pixels statistic can be formulated as hypothesis testing problems when both problems have non coded image data as their input. In section 3 the modified Hadamard basis is introduced. In section 4 the histogram matching test is formulated as a hypothesis testing problem. In section 5 we compare two classes of statistics : the *position dependent* ones, like the correlation statistic, and the *position independent* ones like those we introduce in sections 2 and 4. In section 6 we present performance evaluation results for these two rules. In section 7 we rely on our conclusions from the matching problem to visualize the Image Compression and Image Understanding problems as two possibly related abstractions of the same information retrieval problem.

2. DETECTION OF A BLACK SQUARE IN WHITE BACKGROUND

Suppose we are given an $n \times n$-pixel binary (zero/one) image which represents an $m \times m$-pixel black square in white background and is corrupted by BSC noise with pixel inversion probability ϵ. We want to locate the black square on the image.

2.1 Detection Based on the Sum-of-Pixels Statistic

The use of the sum-of-pixels statistic y is equivalent to counting the black pixels on the image window we scan. We construct an M-ary hypothesis test for which the states

$$H_w,\ w \in \{0\} \cup \{1,2,\cdots,m^2\} = \{0\} \cup S$$

are characterized by the Hamming weights of the possible m×m patterns generated by scanning the noise free image ; more specifically S is the set of non all-white such patterns. The prior probability of a state h, i.e. he probability that the Hamming weight of the test window on the noise free image has the value $w = h$, is denoted as p_h.

$$c(.,.) = \begin{array}{c|cccccc} & 0 & 1 & \cdots & & & M \\ 0 & 0 & 1 & 1 & \cdots & & 1 \\ 1 & 1 & & & & & \\ \vdots & 1 & & & \bigcirc & & \\ & \vdots & & & & & \\ M & 1 & & & & & \end{array} \tag{1}$$

By selecting the cost matrix in eq.1 for our M-ary test we implicitly impose the grouping of the states $H_i,\ i = 1,2,\ldots,M$ into a new one $\widetilde{H}_1$, thus reducing our M-ary rule into a binary one. More explicitly we can apply the theory supporting the M-ary test[6]; this will give the following decision rule :

$$d\ :\ d(\mathbf{y}) = i \Leftrightarrow g_i(\mathbf{y}) \leq g_h(\mathbf{y}),\ h \in S \cup \{0\}, \qquad (2) \qquad where$$

$$g_i(\mathbf{y}) = \sum_{\substack{h \in S\cup\{0\} \\ h \neq i}} p_h \left[c(h,i) - c(h,h)\right] P_h(\mathbf{y}), \quad or \quad g_i = \begin{cases} \sum_{h\in S} p_h P_h(\mathbf{y}), & i = 0 \\ p_0 P_0(\mathbf{y}), & i \in S \end{cases}$$

which minimizes the mean cost value : $J(d) = E[c(H, d(\mathbf{Y}))]$. Note that all g_i's are identical for $h \in S$. This means that essentially we have two hypothesis for our test :

$$\widetilde{H}_0\ :\ h = 0, \qquad \widetilde{H}_1\ :\ h \geq 1$$

The decision rule is :

$$d\ :\ d(y) = i \Leftrightarrow g_i(y) \leq g_{1-i}(y), \qquad g_i(y) = \begin{cases} \sum_{h\in S} p_h P_h(y), & i = 0 \\ p_0 P_0(y), & i = 1 \end{cases} \tag{3}$$

$P_h(y), h \in S \cup \{0\}$, is the probability of observing an image window with y black pixels in it, while the corresponding noise free window had h ones. We will call these probabilities *transition probabilities* and they are shown[7] to be equal to

$$P_h(y) = \sum_{k=\max(0,h+y-\tilde{m})}^{\min(h,y)} \binom{h}{y} \binom{\tilde{m}-h}{y-k} (1-\epsilon)^{\tilde{m}-h-y+2k} \epsilon^{h+y-2k}\ . \tag{4}$$

If we substitute this expression of $P_h(y)$ in the decision rule the latter becomes:

$$d\ :\ d(y) = 0 \Leftrightarrow \frac{1}{p_0} \sum_{h\in S} p_h \sum_{k=k_{min}}^{k_{max}} \binom{h}{k} \binom{\tilde{m}-h}{y-k} \left(\frac{\epsilon}{1-\epsilon}\right)^{h-2k} < \binom{\tilde{m}}{y} \tag{5}$$

and by a further numerical simplification :

$$d\ :\ d(y) = 0 \Leftrightarrow y \leq y_0\ , \tag{6}$$

where y_0 is a threshold in the range from 0 to m^2.

Performance evaluation :

Probability of false alarm = Pr{ decide there is a square while it is not present } :

$$P_{fa} = Pr\{d \neq 0 \mid H = 0\} = Pr\{y > y_0 \mid H_0\} \Rightarrow$$
$$P_{fa} = \sum_{y=y_0+1}^{\tilde{m}} \binom{\tilde{m}}{y} (1-\epsilon)^{\tilde{m}-y} \epsilon^y \quad (7)$$

Probability of detection = Pr{ decide there is a square while it is present } :

$$P_d = Pr\{d \neq 0 \mid H \neq 0\} = \sum_{h \in S} Pr\{d \neq 0 \mid H = h\} Pr\{H = h \mid H \neq 0\}$$
$$= 1 - \sum_{h \in S} Pr\{y \leq y_0 \mid H = h\} \frac{p_h}{\sum_{h \in S} p_h} \Rightarrow$$
$$P_d = 1 - \frac{1}{\sum_{h \in S} p_h} \sum_{h \in S} p_h \sum_{y=0}^{y_0} P_h(y) \quad . \quad (8)$$

2.2 Detection Based on Image Data of a Coarsened Resolution

Suppose now that we code each 2×2–pixel block of the noise corrupted image with one bit indicating all-black or all-white block, depending on the bit value. The noise of the original image will "propagate" to the coded data, resulting in bit inversions. The white noise in the original data will cause the noise to be white in the coded data too.

The relation of the noise-free-coded data and the noise-corrupted-and-then-coded data is depicted in figure 2. The inversion probabilities ϵ_0=*Pr{0 is inverted}* and ϵ_1=*Pr{1 is inverted}* depend on the specific coding rule. We adopt the following coding (quantization) scheme :

For each 2 × 2–pixel block (call it y) of the noise corrupted image do :

1. if w(y) < 2 then decode y→0
2. if w(y) > 2 then decode y→1
3. if w(y) = 2 then
   decode y→0 with probability 0.5
   decode y→1 with probability 0.5

where w(y) is the Hamming weight of y. We will have $\tilde{\epsilon}=\epsilon_0=\epsilon_1$ because of the symmetry of the rule. So the noise on the coded data can be modeled again as BSC noise with inversion probability $\tilde{\epsilon}$, which depends on the noise parameter ϵ of the original data as follows :

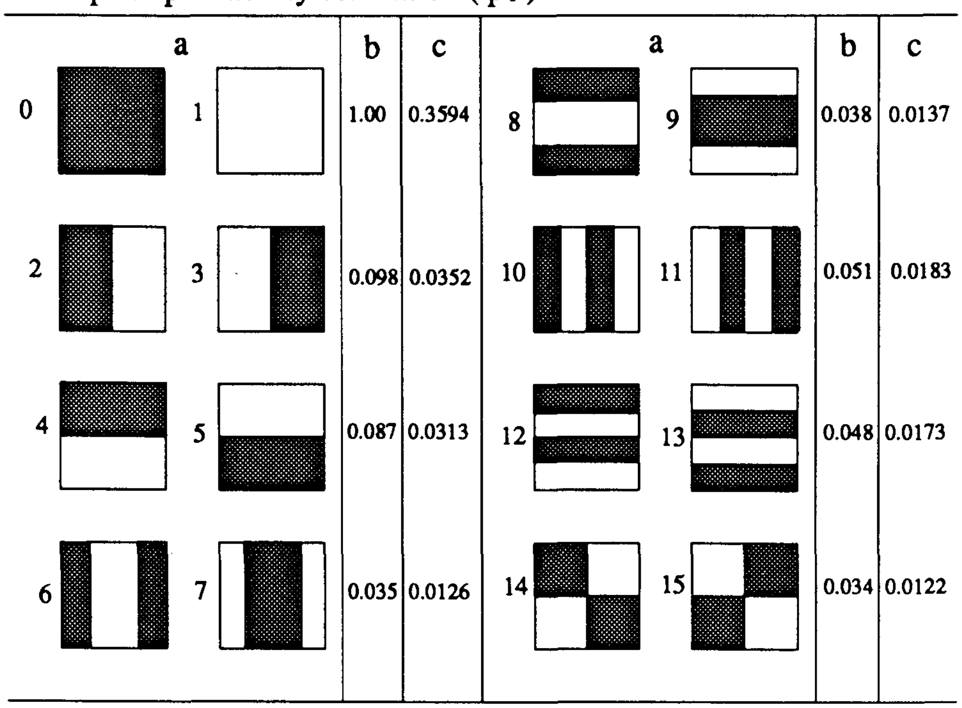
$$\tilde{\epsilon} = Pr\{have\ more\ than\ 2\ bit\ inversions\} + \frac{1}{2} Pr\{have\ 2\ bit\ inversions\} \Rightarrow$$
$$\tilde{\epsilon} = \sum_{i=3}^{4} \binom{4}{i} \epsilon^i (1-\epsilon)^{4-i} + \frac{1}{2}\binom{4}{2} \epsilon^2 (1-\epsilon)^2 \quad . \quad (9)$$

An analysis for the detection of the black square based on these coded data is identical to the one we have seen in subsection 2.1.

3. THE MODIFIED HADAMARD BASIS

In this section we introduce a simple block coding scheme for binary images. Both the observation data and the template we wish to match will be treated as matrices of block codewords instead of matrices of pixel values. A consequence of this fact is that the noise in the original image will "propagate" in the coded image data. For example, suppose that a noise free block of pixels is coded by a codeword i; but due to some pixel inversion(s) in this block the decoded codeword assigned to it is j, which is different from i. We would like to be able to evaluate the probabilities of such events called *block transition probabilities* $\epsilon_{ij} = Pr\{i \rightarrow j\}$ in terms of the pixel inversion probability of the original data. On the other hand the size of the block alphabet M will impose a second restriction on the selection of the coding scheme we can use. In the next section we will show that the complexity of the matching test will be exponential with respect to M. A third constraint which conflicts with the latter arizes from the fact that we do not want our template to be too minute. We wish to develop a matching rule for templates of size 8×8–pixel or 12×12–pixel.

The 16–element 4×4–pixel block alphabet shown in figure 1, called *the modified Hadamard basis* has been inspired by the 4×4 Hadamard basis[4] and responds satisfactorily to all of the above constraints. The Hadamard basis is used for multiple gray level images; negative intensity factors can reverse the pattern, e.g. a black block may be converted into a white one. In our application we do not use such a factor; instead we use the first 8 patterns of the Hadamard basis plus their reversed versions.



a. binary pattern b. normalized variance
c. prior probability estimation (p_i)

a		b	c	a		b	c
0	1	1.00	0.3594	8	9	0.038	0.0137
2	3	0.098	0.0352	10	11	0.051	0.0183
4	5	0.087	0.0313	12	13	0.048	0.0173
6	7	0.035	0.0126	14	15	0.034	0.0122

Figure 1: The modified Hadamard basis

The block transitions are caused by the white binary noise corruption. Nevertheless they are governed by the quantization procedure, i.e. the rule used to map the noise corrupted data back to the limited alphabet of the 16 blocks. Both the quantization procedure and the block transition probabilities evaluation rely on a useful property of this block alphabet : for each pair of distinct blocks i, j the Hamming distance d(i,j) is either 8 or 16. Therefore we may think of the codewords as the centers of spheres in the $\{0,1\}^{16}$ – space with radius half the Hamming distance ρ=4.

The quantization rule :

For each 4×4–pixel block (call it y) of the noise corrupted image do:

1. if d(y,i) < 4 for some codeword i then decode y→i
2. if d(y,i) > 4 for all i
   a. if "weight of y" (w(y)) > 8 decode y→0 (full black block)
   b. if w(y) < 8 decode y→1 (full white block)
   c. if w(y) = 8,
      decode y→0 with probability 0.5,
      decode y→1 with probability 0.5
3. if d(y,i) = 4 for one or more i's then
   decode y→j_m, where $j_m = \arg\max_j \{prior(j)\}$.

The transition probabilities : Recall that we call *transition probability* Pr{i→j} = P(j / i) = Pr{d=j / H=i} the probability of deciding that the element (codeword) j appears in some place of the image, given that the element (codeword) i was at that place of the original image. The probabilities $\epsilon_{ij} = P(j \mid j)$, $i, j = 0, 1, \cdots, 15$ are found to be expressed by fairly complex combinatorial formulas7.

4. OBJECT DETECTION BASED ON CODED IMAGE DATA

In section 3 we introduced a block coding scheme transforming a binary image into a matrix of codewords in the range 0, ..., 15. A 12×12–pixel image window is thus transformed into a 3×3–code sub-matrix, since 4×4–pixel blocks were considered. The 9 elements of this sub-matrix are independent random variables, because the BSC noise is white (in space) and their values define a configuration which will be called *the original state*. If we now group together all the original states having the same (1st order) histogram (i.e. all sub-matrices having the same multitude of each element) then we form a new set of states. We will call this new set *the histogram alphabet.*

Let M be the number of all possible distinct components (block codes) which may constitute the template. Let also n be the length of the template in blocks (in the example above we had $n = 9 = 3\times3$ blocks). We will denote by $S(M,n)$ the size of the histogram alphabet. Call **h** the M-vector representing the histogram of the noise-free coded template of size n; call also **y** the M-vector representing the histogram of a noise corrupted and afterwards coded image window of size n. The vector **h** represents some feature we want to detect on the image; our problem amounts to comparing the two histograms **h** and **y** and infer a decision about their matching.

We formulate this matching as an M-ary hypothesis testing problem which leads to an optimal binary decision rule. To do so we assume we are given the prior probabilities and the transition probabilities of the histogram elements (i.e. the block priors p_i, $i = 0, 1, \cdots, M-1$ and block transitions ϵ_{ij}, $i, j = 0, 1, \cdots, M-1$) so that we are able to find the *histogram priors* and *histogram transitions.*

In this section we develop a rule which is a generalization of the one developed in section 2. One may observe that if we set $M = 2$, i.e. if we have a binary code, then the histogram reduces to the "sum-of-pixels" statistic we discussed in section 2. The generalized rule can be applied to compare arbitrary histograms, provided we have the information about the histogram elements mentioned above.

4.1 The Histogram Alphabet

Let $\{\alpha_0, \alpha_1, \cdots, \alpha_{M-1}\}$ be the block alphabet. Evidently we will have M^n possible n-tuples of blocks. We introduce an equivalence relation onto this set of n-tuples. An equivalence class will be composed of all n-tuples having the same histograms, i.e. two n-tuples are in the same class if one is a permutation of the elements of the other. A representative element of an equivalence class will be called *a histogram pattern.* Consider now the following power expansion[8] :

$$(\alpha_0 + \alpha_1 + \cdots + \alpha_{M-1})^n = \sum_{\{n_i\}:\sum n_i = n} \frac{n!}{n_0! n_1! \cdots n_{M-1}!} \alpha_0^{n_0} \alpha_1^{n_1} \cdots \alpha_{M-1}^{n_{M-1}} . \tag{10}$$

Observe that :

1. Each term in the summation above can be considered as a histogram pattern, i.e.

$$\alpha_0^{n_0}\alpha_1^{n_1}\cdots\alpha_{M-1}^{n_{M-1}}$$

represents an n-tuple in which we have n_0 times the element α_0, n_1 times the element α_1 and so on; note that if $n_i = 0$ for some i, then α_i is not in the n-tuple.

2. The size of each equivalence class will be equal to

$$\frac{n!}{n_0!n_1!\cdots n_{M-1}!}$$

3. If p_i is the prior probability for $\alpha_i, i = 0, 1, \cdots, M-1$, then the independence assumption implies that the prior probability for a specific element in the equivalence class will be

$$p_0^{n_0}p_1^{n_1}\cdots p_{M-1}^{n_{M-1}} ,$$

while the prior for the class itself, i.e. the prior of the histogram pattern will be

$$\frac{n!}{n_0!n_1!\cdots n_{M-1}!}p_0^{n_0}p_1^{n_1}\cdots p_{M-1}^{n_{M-1}} .$$

4. We can systematically construct the histogram patterns with the following procedure :

   a. Find all integer partitions of the number n in at most M places

   $$\{n_0, n_1, \cdots, n_{M-1}\} .$$

   b. For each such partition find all the distinct permutations of n_i's in M places. Note that we have

   $$\binom{M}{n_0}\binom{M-n_0}{n_1}\cdots\binom{n_{M-1}}{n_{M-1}}$$

   of them.

   Each permutation will determine a set of exponents in the expansion formula and therefore a histogram pattern.

5. The number of histogram patterns, i.e. the size of the histogram alphabet, $S(M,n)$ will be equal to the number of summands and therefore

$$S(M,n) = \binom{M+n-1}{n} . \tag{11}$$

Note that in the case of the binary block alphabet (e.g. black/white) we get $S(2,n) = n+1$ = the number of possible variations of black-white mixtures in a pattern of size n.

The equivalence classes described (histogram patterns) will determine the states for the M-ary test we will develop; the state space will be denoted by H.

4.2 The Observation Data and the Cost Function

The observation data in our test correspond to some n-vector of block codewords. This information can equivalently be represented by an M-vector y being the histogram pattern of the image window we scan. For example if $n = 4$ and $M = 16$ we may have the observation *[1 5 15 1]* which is equivalent to the histogram pattern y = [0 2 0 0 0 1 0 0 0 0 0 0 0 0 0 1] .

We introduce a partition H_0, H_1 of the state space H. Given an observation y we want to decide "matching" (H_1) or "not matching" (H_0). We will do so by imposing the cost function :

$$c(\mathbf{h},\mathbf{y}) = \begin{cases} 0, & if\ both\ \mathbf{h},\mathbf{y}\ belong\ to\ either\ H_0\ or\ H_1 \\ 1, & if\ \mathbf{h},\mathbf{y}\ do\ not\ belong\ to\ the\ same\ set\ H_q,\ q = 0,1 \end{cases} \tag{12}$$

Note that H_1 is not necessarily a singleton. This means we may consider matching with multiple histogram patterns simultaneously.

4.3 The Transition Probabilities Among the Histogram Patterns

Let us summarize our notation and introduce some new one. We have :

h : Histogram pattern, $\mathbf{h} \in H$; e.g. for $n = 4$: the template *[1 0 2 2]* gives the histogram $\mathbf{h}$ = *[1 1 2]*
$h_i, i = 0, 1, 2$: Number of occurrences of the block i in the pattern, e.g. $h_0 = 1,\ h_1 = 1,\ h_2 = 2$

y : Observation pattern : it has the same format as **h** and is subject to comparison with it.

$y_i,\ i = 0, 1, 2$: Number of occurrences of the block i in the observation vector.

ϵ_{ij} : Block transition probability *Pr{i→j}*.

k_{ij} : Number of (original) blocks i in **h** "transformed" (as a result of noise corruption) into blocks j in the observation pattern **y**.

It has been shown[7] that the histogram transition probability

$$
\begin{aligned}
P_{\mathbf{h}}(\mathbf{y}) &= P_{[h_0, h_1, \cdots, h_{n_h}]}([y_0, y_1, \cdots, y_{n_y}]) \\
&= Pr\{\mathbf{y}\ \textit{is composed by}\ y_0\ \textit{elements (blocks) of type}\ t_0, \\
&\qquad y_1\ \textit{elements of type}\ t_1, \\
&\qquad \cdots \\
&\qquad y_{n_y}\ \textit{elements of type}\ t_{n_y}\ / \\
&\qquad \mathbf{h}\ \textit{is composed by}\ h_0\ \textit{elements (blocks) of type}\ t'_0, \\
&\qquad h_1\ \textit{elements of type}\ t'_1, \\
&\qquad \cdots \\
&\qquad h_{n_h}\ \textit{elements of type}\ t'_{n_h}\ \} \\
&\quad \times\ \textit{the size of the histogram class of}\ \mathbf{y}
\end{aligned}
$$

is equal to

$$
P_{\mathbf{h}}(\mathbf{y}) = \frac{n!}{y_1! y_2! \cdots y_{n_y}!} \sum_{\substack{\text{a set of acceptable} \\ k_{ij}\ \text{tuples}}} \prod_i P_i\left(k_{i1}, k_{i2}, \cdots, k_{in_y - 1}\right) \quad , \tag{13}
$$

where

$$
P_i\left(k_{i1}, k_{i2}, \cdots, k_{in_y-1}\right) = \underbrace{\binom{h_i}{k_{i1}} \binom{h_i - k_{i1}}{k_{i2}} \cdots \binom{h_i - \sum_{j=1}^{n_y-2} k_{ij}}{k_{n_y-1}}}_{\leftarrow\ n_y - 1\ \rightarrow} \times \underbrace{\epsilon_{i1}^{k_{i1}} \epsilon_{i2}^{k_{i2}} \cdots \epsilon_{in_y}^{h_i - \sum_{j=1}^{n_y-1} k_{ij}}}_{\leftarrow\ n_y\ \rightarrow} \tag{14}
$$

and $n_h,\ n_y$ are the number of distinct elements in the template and the observation data respectively[7]. The acceptable k_{ij}—tuples satisfy the inequalities :

$$
\begin{aligned}
&\max\left\{0, h_i - y_{n_y-1}\right\} \le \sum_{j=1}^{n_y-1} k_{ij} \le h_i, \quad i = 1, 2, \cdots, n_h - 1 \\
&\max\left\{0, y_j - h_{n_h-1}\right\} \le \sum_{i=1}^{n_h-1} k_{ij} \le y_j, \quad j = 1, 2, \cdots, n_y - 1
\end{aligned} \tag{15}
$$

4.4 The Decision Rule

Let y be the observation pattern, $p_{\mathbf{q}}$, $\mathbf{q} \in H$ be the prior probabilities for the histogram patterns and $P_{\mathbf{h}}(\mathbf{y})$, $\mathbf{h}, \mathbf{y} \in H$ be the transition probabilities. The rule *d(.)* that minimizes the mean cost of the "matching"/"non matching" decision is :

$$d(\mathbf{y}) = i \Leftrightarrow g^i(\mathbf{y}) \leq g^{1-i}(\mathbf{y}), \quad g^i(\mathbf{y}) = \begin{cases} \sum_{\mathbf{q} \in H_0} p_{\mathbf{q}} P_{\mathbf{q}}(\mathbf{y}), & i = 1 \\ \sum_{\mathbf{q} \in H_1} p_{\mathbf{q}} P_{\mathbf{q}}(\mathbf{y}), & i = 0 \end{cases} . \tag{16}$$

We can numerically determine a partition $\{R_0, R_1\}$ of H^7 such that

$$d(\mathbf{y}) = i \Leftrightarrow \mathbf{y} \in R_i \, , \quad i = 0, 1 \, .$$

We recall that

$$d(\mathbf{y}) = \begin{cases} 1 & if \ we \ decide \ matching \\ 0 & if \ we \ decide \ non \ matching \end{cases} .$$

Note also that the calculation of the partition $\{R_0, R_1\}$ corresponds to the calculation of a threshold value in the case of the binary block alphabet as we have seen in section 2.

4.5 Performance Evaluation of the Decision Rule

We will calculate the (P_{fa}, P_d) pair for our rule.

Probability of false alarm :

$$\left.\begin{array}{l} P_{fa} = Pr\{\mathbf{d} \in H_1 \mid \mathbf{h} \in H_0\} = \sum_{\mathbf{i} \in H_1} Pr\{\mathbf{d} = \mathbf{i} \mid \mathbf{h} \in H_0\} \\ Pr\{\mathbf{d} = \mathbf{i} \mid \mathbf{h} \in H_0\} = \sum_{\mathbf{i} \in H_0} Pr\{\mathbf{d} = \mathbf{i} \mid \mathbf{h} = \mathbf{j}\} \, Pr\{\mathbf{h} = \mathbf{j} \mid \mathbf{h} \in H_0\} \end{array}\right\} \Rightarrow$$

$$P_{fa} = \sum_{\mathbf{i} \in H_1} \sum_{\mathbf{j} \in H_0} Pr\{\mathbf{d} = \mathbf{i} \mid \mathbf{h} = \mathbf{j}\} \, Pr\{\mathbf{h} = \mathbf{j} \mid \mathbf{h} \in H_0\} \Rightarrow$$

$$P_{fa} = \sum_{\mathbf{j} \in H_0} Pr\{\mathbf{h} = \mathbf{j} \mid \mathbf{h} \in H_0\} \sum_{\mathbf{i} \in H_1} Pr\{\mathbf{d} = \mathbf{i} \mid \mathbf{h} = \mathbf{j}\} \quad , \tag{17}$$

$$where \ \ Pr\{\mathbf{d} = \mathbf{i} \mid \mathbf{h} = \mathbf{j}\} = \sum_{\mathbf{y} \in \widetilde{R}_{\mathbf{i}}} P_{\mathbf{j}}(\mathbf{y})$$

for some decision region $\widetilde{R}_i$. So

$$P_{fa} = \frac{1}{\sum_{\mathbf{l} \in H_0} p_{\mathbf{l}}} \sum_{\mathbf{j} \in H_0} p_{\mathbf{j}} \sum_{\mathbf{i} \in H_1} \sum_{\mathbf{y} \in \widetilde{R}_{\mathbf{i}}} P_{\mathbf{j}}(\mathbf{y}) \Rightarrow$$

$$P_{fa} = \frac{1}{\sum_{\mathbf{l} \in H_0} p_{\mathbf{l}}} \sum_{\mathbf{j} \in H_0} p_{\mathbf{j}} \sum_{\mathbf{y} \in R_1} P_{\mathbf{j}}(\mathbf{y}) \quad . \tag{18}$$

Similarly the *probability of miss* P_m is found to be

$$P_m = \frac{1}{\sum_{\mathbf{l} \in H_1} p_{\mathbf{l}}} \sum_{\mathbf{i} \in H_1} p_{\mathbf{i}} \sum_{\mathbf{y} \in R_0} P_{\mathbf{i}}(\mathbf{y}) \tag{19}$$

and as usually the *probability of detection* P_d is $P_d = 1 - P_m$.

5. POSITION INDEPENDENT STATISTICS v.s. POSITION DEPENDENT STATISTICS

The tests we have examined so far rely on the sum-of-pixels or the histogram statistic. As already demonstrated the common characteristic of these two statistics is that they do not take into consideration the position information. We will call this type of statistics *position independent statistics.* On the other hand we will call *position dependent statistics* the ones which rely on position information, like the correlation statistic or the number of matches between two binary patterns. In this section we will compare the two kinds of statistics in terms of computational complexity and the reliability of the related tests; the latter will be called *position dependent tests* and *position independent tests* respectively.

Consider the case of a d×d-element template (note : d^2=n) of M-valued elements. The set of possible templates is of size $M^{d^2} = M^n$, which reflects the size of the state space of a position dependent test. On the other hand the state space size of the position independent tests is $S(M,n) = \binom{M+n-1}{n}$. The template size determines the computational complexity for determining the Bayesian rule, as well as the complexity and memory requirements for evaluating the rule. By using Stirling's approximation for the factorials one can easily show that for the hard case of M>>n we have

$$\frac{S(M,n)}{M^n} \underset{M>>n}{\longrightarrow} \left(\frac{e}{n}\right)^n$$

Still the position dependent rules imply computationally expensive matching tests, since at each time instant of the image scanning they need $O(n^2)$ operations, while the position independent ones require $O(min(max(d,M),d^2)$ or for smooth enough images $O(d)$ operations[7].

The penalty for the reduced complexity of the position independent statistics is a higher probability of false alarm compared to that of the position dependent ones. Let us suppose that among the $d^2=n$ elements of X^t we have k distinct ones namely n_0 of type *0*, n_1 of type *1*, ..., n_k of type *k*, so that $n_0 + n_1 + \cdots + n_k = n$. This histogram corresponds to $c(\mathbf{n}) = \binom{n}{n_0}\binom{n-n_0}{n_1}\binom{n-n_0-n_1}{n_2}\cdots\binom{n_k}{n_k}$ possible template patterns. Suppose that among these $c = c(\mathbf{n})$ patterns there is the one we want to detect. Ideally our rule will respond positively each time it scans one of these patterns. Consequently, each time we scan our target pattern, the rule will recognize it; but it will recognize all the $c-1$ patterns sharing the same histogram with our target as well. Let us denote with $h, P_{fa}, P_d, \tilde{h}, \tilde{P}_{fa}, \tilde{P}_d$ the state and performance evaluation parameters for the position independent test and the position depended test respectively and also with d the decision outcome *0* or *1*. In figure 2 we can see the relations of the regions characterized by the values of $h, \tilde{h}$ and d.

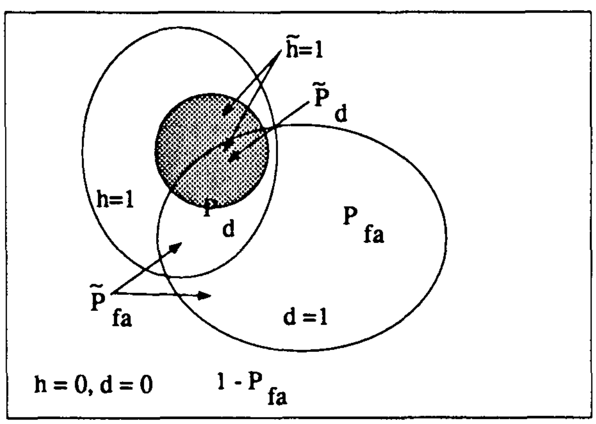


Figure 2: Comparison of the performance characteristics for the position dependent and position independent tests.

We now make the following assumption : Given that an image element lies in a template the probability mass function of the random variable indicating its position and ranging over all the allowable positions is a uniform pmf. In other words if c(**n**) distinct patterns result in the same histogram **n** (and no other does), given the histogram **n** any pattern resulting it may occur with probability 1/c(**n**).

From figure 2 we induce that the above assumption implies :

$$Pr\left\{\tilde{h}=1\right\}=\frac{1}{c}Pr\{h=1\} \tag{20}$$

$$Pr\left\{d=1,\tilde{h}=1\right\}=\frac{1}{c}Pr\{d=1,h=1\}. \tag{21}$$

From (21) we get :

$$\widetilde{P}_d=Pr\left\{d=1\mid\tilde{h}=1\right\}=\frac{1}{c}\frac{Pr\{d=1\mid h=1\}\,Pr\{h=1\}}{Pr\left\{\tilde{h}=1\right\}} \tag{22}$$

$$(20),(22)\Rightarrow\widetilde{P}_d=\frac{1}{c}P_d\cdot c=P_d$$

as it was expected. We also have :

$$\begin{aligned}Pr\left\{\tilde{h}=0\right\}&=Pr\{h=0\}+Pr\left\{h=1,\tilde{h}=0\right\}\\&=Pr\{h=0\}+Pr\{h=1\}-Pr\left\{\tilde{h}=1\right\}\\&=Pr\{h=0\}+Pr\{h=1\}-\frac{1}{c}Pr\{h=1\}\\&=Pr\{h=0\}+\left(1-\frac{1}{c}\right)Pr\{h=1\}\end{aligned} \tag{23}$$

and similarly

$$Pr\left\{d=1,\tilde{h}=0\right\}=Pr\{d=1,h=0\}+\left(1-\frac{1}{c}\right)Pr\{d=1,h=1\}\;. \tag{24}$$

$$\begin{aligned}\widetilde{P}_{fa}&=Pr\left\{d=1\mid\tilde{h}=0\right\}=\frac{Pr\left\{d=1,\tilde{h}=0\right\}}{Pr\left\{\tilde{h}=0\right\}}\\&=\frac{Pr\{d=1,h=0\}+\left(1-\frac{1}{c}\right)Pr\{d=1,h=1\}}{Pr\{h=0\}+\left(1-\frac{1}{c}\right)Pr\{h=1\}}\Rightarrow\\\widetilde{P}_{fa}&=\frac{\alpha P_{fa}+\beta P_d}{\alpha+\beta}\;,\;\alpha=\frac{1}{Pr\{h=1\}}\;,\;\beta=\frac{1-\frac{1}{c}}{Pr\{h=0\}}\;.\end{aligned} \tag{25}$$

We may observe that as n gets larger c gets larger and consequently β gets larger, so $\widetilde{P}_{fa}$ gets larger. Note also that the prior knowledge $Pr\{h=i\}$, $i=0,1$, explicitly affects $\widetilde{P}_{fa}$. For the full black template we have $c(\mathbf{n})=1$; so for equal priors we have $\widetilde{P}_{fa}=P_{fa}$. This result holds for the all-white template as well.

Thus our position independent tests :

- Are faster but tend to have higher probability of false alarm than the position dependent tests.
- Accept analytic performance evaluation and have the same power with the Bayesian position dependent tests.
- Are more attractive from the computational complexity point of view than the position dependent Bayesian tests, but still not adequately attractive for non trivial templates.
- Are equivalent with the Bayesian position dependent tests, in terms of the P_{fa} and P_d characteristics, for the case of the full black and all-white templates, under mild conditions.

Before we close this section we would like to point out that in the case of color patterns the fact that position independent matching tests are faster than the position dependent ones reflects the human ability of recognizing the color faster than the shape of an object [9].

6. NUMERICAL RESULTS

The fundamental question raised in the previous sections is how the noise on the observed image will affect the capability of object detection. The noise level is expressed in terms of the pixel inversion probability ϵ, while the capability of object detection is captured by the pair : probability of false alarm P_{fa} and probability of detection P_d which correspond to the optimal Bayesian detection rule.

We use the results of section 2 to obtain the plots of the figures 4a and 4b, where we depict the ROC (Region Of Convergence) of the decision rule for pixel inversion probabilities ϵ=0.1 and ϵ=0.3 respectively. Recall that we want to detect an 8×8–pixel black square in white background. Figure 4a refers to the rule based on the original pixel data and figure 4b refers to the rule based on the coarsened resolution data. The first is characterized as a (2,64)-KB type test, since the observation data are composed by 8×8=64 2–valued pixels. The second test is characterized as a (2,16)-KB type test, since the observation data are composed by 4×4=16 2–valued blocks (of 2×2=4 pixels each one) as described in subsection 2.1. The symbol KB stands for " Known Background" and it is justified from the fact that we know the background of the target square. Indeed, the M states of the underlying M-ary test correspond to the possible distinct relative positionings of the target square and the scanning window in the noise free case.

In figure 4c we give the ROC curves for pixel inversion probabilities 0.1 and 0.3 when the target object is the one shown in figure 3 and the image is coded with the first four elements of the modified Hadamard basis. This test is a (4,2)-UB type test, since the target object is composed by 2×1=2 4–valued blocks. The symbol UB stands for "Unknown Background" and it is justified from the fact that we do not make any use of background knowledge as we have seen from the formulation of the problem in section 4. Nevertheless, we may observe that the(4,2)-UB test seems to perform much better than the (2,16)-KB and (2,64)-KB tests (the ROC curve is closer to the point $\{P_{fa}=0, P_d=1\}$). This is explained by the fact that in the KB type tests we wish to decide H_1, i.e. "target object detected", if a black square or even a part of a black square is observed, while in the UB-type tests we wish to decide H_1 only if the target object itself is observed.

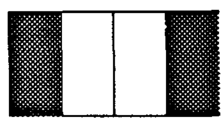


Figure 3: An example of 1×2–block template

Given a ROC curve there is a point on it that represents the performance of the related optimal Bayesian decision rule. In our problem we have such a point for each value of the pixel inversion probability ϵ. If we draw the curve passing from the points representing the optimal Bayesian rules for a continuity of values of ϵ we obtain a P_{fa}—P_d curve parametrized with ϵ. An example of a P_{fa}—P_d curve is given in figure 4d for the (2,64)-KB test. We observe that for low noise, i.e. for small values of ϵ, P_{fa} is small and P_d is high. This happens for small values of the threshold y_0 of the sum-of-pixels statistic (see section 2). For instance if ϵ=0.02 the optimal Bayesian rule is "decide black square detected if sum-of-pixels > 37". Adding more noise, i.e. letting ϵ get larger, causes P_{fa} to augment, while P_d gets smaller and y_0 gets larger; y_0 getting larger means that the rule gets conservative, i.e. it needs more evidence (darkness) in order to infer "black square detected". This process gradually leads to the rule "never decide black square detected", i.e. to the point (0,0) of the P_{fa}—P_d plot. The stairs-like shape of the curve is due to the discrete nature of the threshold and makes apparent the trade-off between the high P_d and the low P_{fa} that the optimal Bayesian rule attempts to compensate.

In figures 4e and 4f we compare the performance characteristics of four detection tests used for detecting a black square in white background. Note that the last of them does not make any use of the background information. The first two are the already known (2,64)-KB and (2,16)-KB tests. The third is a (2,4)-KB test, i.e. the 8×8–pixel black square is treated as a 2×2 matrix of 2 valued-blocks where each block is of size 4×4–pixel. The block value is determined by a majority rule similar to the case of the 2×2–pixel blocks used in the (2,16)-KB type test. The fourth test into consideration is a (16,4)-UB type test. The black square again is treated as a 2×2 matrix of blocks, but now the blocks take values in the modified Hadamard basis as presented in section 3.

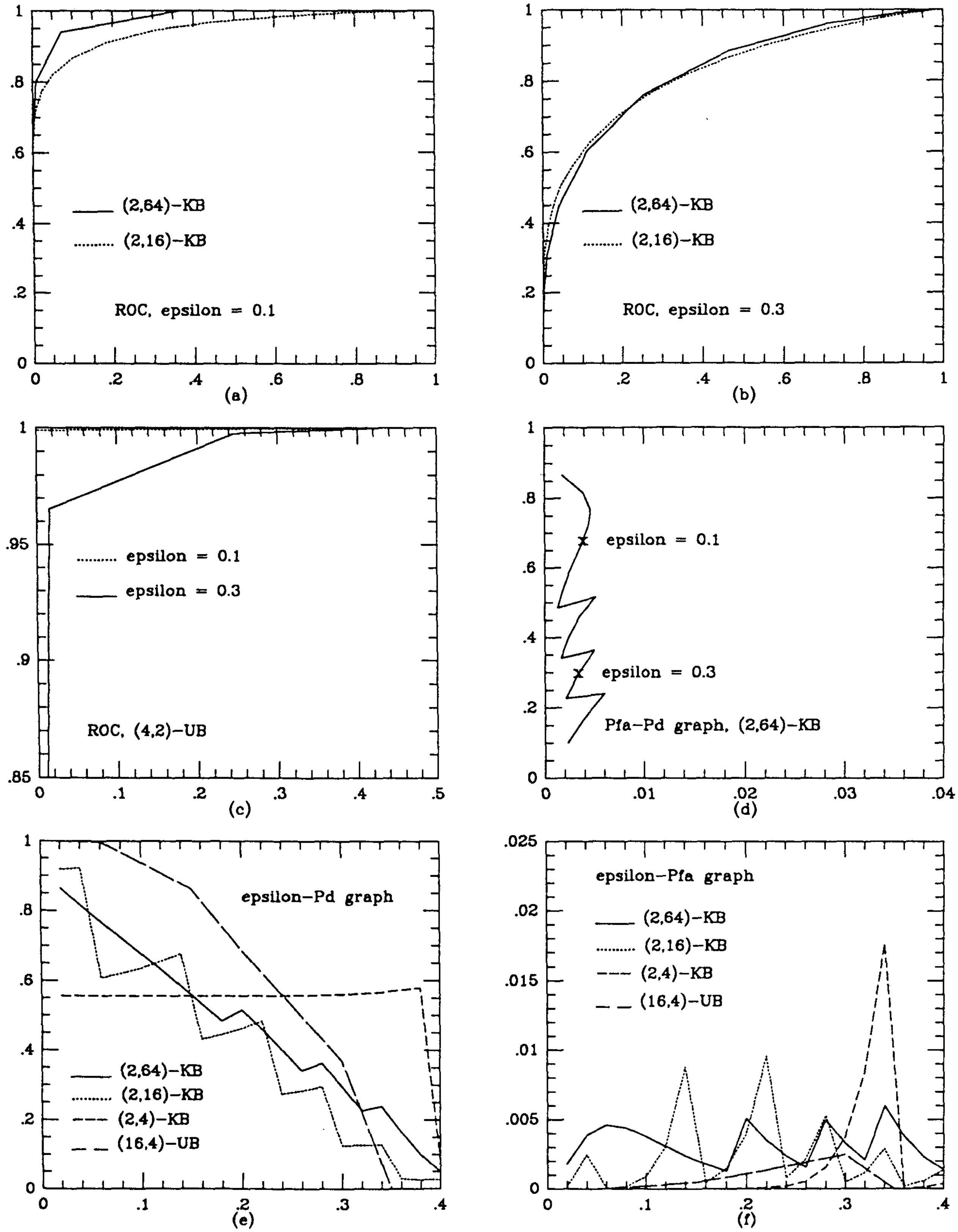


Figure 4: Performance evaluation plots

In figure 4e we represent the probability of detection P_d as a function of the pixel inversion probability ϵ. It is apparent that at certain noise level the compression of data favors detection capability, while for higher noise level better detection results are obtained by a rule based on the uncoded data. More specifically the (2,16)-KB test has higher P_d at three distinct intervals of ϵ (around the points ϵ=0.02, ϵ=0.14 and ϵ=0.22). Also the (2,4)-KB test is superior to the other two tests for heavily noise corrupted data (ϵ>0.15). In figure 4f however it becomes evident that these superiorities of the coarsened resolution data always are penalized with peaks of the ϵ—P_{fa} curve. Nevertheless, note that for ϵ<0.05 the (2,16)-KB test performs better than the (2,64)-KB test in terms of both P_{fa} and P_d. We also observe that the performance measures for the (16,4)-UB type test appears to behave in a much better way than the other tests do. This is again due to the fact that the last test concerns the detection of the black square and not the detection of any part of it.

7. DISCUSSION

Image compression and image understanding have been traditionally developed as separate fields of research. In image compression one is primarily interested in designing efficient coding schemes which allow the transmission and accurate reconstruction of images at low bit data rates. In image understanding one is primarily interested in extracting high level or "content" information from the image pixels. It is clear that both fields address the problem of efficient information extraction and that in principle they are strongly coupled. We believe that in order to understand the simultaneous top-down and bottom-up processing performed in human vision, one needs to unify these two fields. It was the purpose of this work to perform an initial investigation in this direction.

More specifically we were interested in linking image coding and object recognition quantitatively in some simple generic examples. We have considered recognition based on a well structured feature detection algorithm, which we developed, resembling Template Matching algorithms. In trying to link image processing with image understanding we introduced a novel idea. Namely the "numerical image" for us was not necessarily a pixel image but a block-coded image. In this way the image compression (performed by the code) is linked directly to image understanding. We can now expect to observe the two types of interference or contamination induced on the image data; namely the "external noise" introduced by the physical media (e.g. the camera or the transmission channel), and the "quantization noise" introduced by the coding procedure. We were interested in analyzing the process where image recognition can be performed based on the reduced (coded) image data. It is clear that if we reduce the image beyond certain point, recognition capability will be affected. On the other hand it is desirable from a practical point of view to design schemes which can perform object recognition on the basis of block-coded data and not requiring the full image reconstruction. It is concluded that what we need to understand and quantify, in a general setting, is the explicit relationship between code efficiency and image compression with the required performance.

8. ACKNOWLEDGEMENTS

This research was partially supported by grands AFOSR-87–0073E and AFOSR-90–0105.

9. REFERENCES

1. A.N. Netravali, J.O. Limb, "Picture coding : A Review", Proc. of the IEEE, vol.68, pp.366–406, 1980.
2. N.M. Nasrabadi, R.A. King, "Image Coding Using Vector Quantization : A Review", IEEE Trans on Communications,vol.36, pp.957–971, 1988.
3. J.J.Y.Huang, P.M.Schultheiss, "Block Quantization of Correlated Gaussian Random Variables", IEEE Trans. on Communication Systems, vol.11, pp. 289–296,1963.
4. H.J.Landau, D. Slepian, "Some Computer Experiments in Picture Processing for Bandwidth Reduction", The Bell System Technical Journal, vol.50, pp.1525–1540, 1971.
5. H. Vincent Poor, An Introduction to Signal Detection and Estimation, pp. 7–12, Springer-Verlag, New York, 1988.
6. Heinrich Niemann, Pattern Analysis, Springer-Verlang, Berlin, pp.110–114, 1981
7. E.N.Frantzeskakis, "On Image Coding and Understanding: A Bayesian Formulation for the Problem of Template Matching Based on Coded Image Data", SRC-M.S.-90–5, Univ. of Maryland, pp.35–36,48–53,60–67,79–80,87–88,90–94, 1990.
8. M.R.Spiegel, Mathematical Handbook of Formulas and Tables, Shaum's Outline Series, p. 4, McGraw-Hill, New York, 1968.
9. Richard E. Christ, "Review and Analysis of Color Coding Research for Visual Displays", Human Factors, vol.17, pp.542–570 , 1975.