High Quality Inverse Halftoning Using Variance Gain-, Texture- and Decision Tree-Based Learning Approach*

KOALONG CHUNG, YONG-HUAI HUANG† AND KANG-CHEH WU
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 106 Taiwan
†Institute of Computer and Communication Engineering
Jinwen University of Science and Technology
Taipei, 231 Taiwan

Inverse halftoning (IH) is used to reconstruct the gray image from an input halftone image. This paper presents a machine learning-based IH algorithm to reconstruct the high quality gray images. We first propose a novel variance gain-based tree construction approach to build up an approximate decision tree (DT). Based on the constructed DT, a texture-based training process is presented to construct a lookup tree-table which will be used in the reconstructing process. In our implementation, thirty training images are used to build up the lookup tree-table and five popular testing images are used to justify the quality performance of our proposed IH algorithm. Experimental results demonstrate that although our IH algorithm needs longest execution-time, it has the highest image quality when compared to the published three IH algorithms.

Keywords: decision tree, discrete cosine transform, inverse halftoning, lookup tree-table, machine learning, texture, vector quantization

1. INTRODUCTION

Inverse halftoning (IH) is used to reconstruct the gray image from an input halftone image. IH has been widely used in the publishing applications, such as newspapers, books, magazines, and so on [1]. It is not a good way to manipulate halftone images since any image processing procedures, such as scaling, compression, and enhancement, on halftone images could cause severe image degradation. To enable these kind operations, gray images need to be reconstructed from the halftone images through IH. Because reconstructing the perfect gray image from the given halftone image is impossible, many IH algorithms have been developed to reconstruct the original one as best as possible.

These developed IH algorithms can be classified into two categories, the filtering-based IH algorithms and the learning-based IH algorithms. In the developed filtering-based IH algorithms, they include the Gaussian lowpass filtering approach [2], the wavelet approach [3-5], the spatial varying FIR filtering approach [6], the non-linear filtering technique [7], the maximum a posteriori estimation [8], the human visual system-based approach [9], and the iterative projection-based approaches [10-14].

In the developed learning-based IH algorithms, they include the vector quantization (VQ) technique [15, 16], the lookup table (LUT)-based IH algorithm (LIH) [17, 18], the...
tree structure-based LUT (TLUT)-based IH algorithm (TLIH) [19], the neural network-based IH algorithm (NNIH) [29], the parallel LUT-based IH algorithm (PLIH) [30], the edge-based LUT IH algorithm (ELIH) [20], and finite state machine-based ELIH algorithm (FELIH) [31]. Experimental data reveal that these latter six approaches have better image quality when compared to the VQ approach. Based on the same thirty training image pairs [18], each image with size, the TLIH reduce the memory required for the LIH to reconstruct gray image with similar quality. The PLIH can speed up the execution-time of LIH by the parallel process. For the high quality consideration, both the NNIH and the ELIH achieve better image quality when compared to the LIH by Chang et al. and Meşe and Vaidyanathan.

In this paper, a machine learning-based IH algorithm is presented to realize the high quality reconstruction of gray images. By using a set of training samples, we first propose a variance gain-based tree construction approach to build up an approximate DT. Based on the constructed DT, a texture-based training process is presented to classify the training samples into many classes where each trained class is attached to each leaf node and then a local codebook is generated by performing the discrete cosine transform (DCT)-based VQ to the attached trained samples. Consequently, a lookup tree-table, which is composed of the approximate DT and the local codebooks, can be constructed by our proposed training process and it will be used to reconstruct gray images from input halftone images. In our implementation, thirty training images from Meşe’s web site are used to build up the lookup tree-table and five popular testing images are used to justify the quality performance of our proposed machine learning-based IH algorithm. Experimental results demonstrate that our proposed IH algorithm has 0.9 dB, 0.4 dB, and 0.5 dB peak signal to noise ratio (PSNR) improvement when compared to the LIH [17, 18], the ELIH [20], and the NNIH [29], respectively.

The rest of this paper is organized as follows. In section 2, the previous IH algorithms are surveyed. In section 3, our proposed variance gain-based DT is presented. In section 4, our proposed machine learning-based IH algorithm is proposed to reconstruct the gray images from input halftone images. Section 5, some experimental results are demonstrated to show the image quality improvement of our proposed IH algorithm. Conclusions are addressed in section 6.

2. RELEVANT PAST WORKS

Fig. 1 shows a $4 \times 4$ template $T$ used by the LIH [17, 18] and the kernel symbol $X$ denotes the estimated/reconstructed pixel. Template $T$ is used as a sliding window to build up the LUT. According to the constructed LUT, the gray image can be reconstructed from the input halftone image.

![Fig. 1. A $4 \times 4$ template $T$ used in the LIH.](image-url)
Before building up the LUT, suppose that initially we have a set of \( M \) training image pairs \( \{(G_m, H_m) | 1 \leq m \leq M\} \) where \( G_m \) denotes the \( m \)th original gray image and \( H_m \) is the corresponding halftone image of \( G_m \). The training halftone image \( H_m \) is obtained by applying the existing halftoning algorithm [1] to the original gray image \( G_m \). The LUT is realized by the array \( \text{LUT}[\cdot] \) to map the input halftone image to the gray image. In each window sliding step, the subimages of \( G_m \) and \( H_m \) covered by \( T \) are denoted as \( S_g \) and \( S_h \), respectively. \( S_{1h}^{\text{th}}, S_{1h}^{\text{th}}, \ldots, S_{1h}^{0h} \), and \( S_{0h}^{\text{th}} \) are the 16 binary values of \( S_h \), and \( S_{1h}^{\text{th}} \) is encoded by

\[
I = \sum_{k=0}^{15} 2^k S_h^{\text{th}},
\]

where \( I, 0 \leq I < 2^{16} \), is used as a mapping address of \( \text{LUT}[\cdot] \). \( S_{1h}^{0h}, S_{1h}^{1h}, \ldots, S_{1h}^{15h} \), and \( S_{0h}^{0h} \) are the 16 gray values of \( S_h \), and \( S_{1h}^{0h} \) is used as the \( T \)-mapped gray value of \( S_h \). In the LIH, the training process to build up the LUT is described as follows:

**Step 1:** Given \( M \) training image pairs \( \{(G_m, H_m) | 1 \leq m \leq M\} \), the template \( T \) is initially put at left-upper corner of \( G_1 \) and \( H_1 \). Initially, we have \( m = 1 \).

**Step 2:** For each subimage \( S_h \) in \( H_m \) covered by \( T \), the corresponding mapping address \( I \) is calculated by Eq. (1).

**Step 3:** According to index \( I \), the following two assignments are performed:

\[
N[I] = N[I] + 1, \quad (2)
\]
\[
\text{LUT}[I] = \text{LUT}[I] + S_{1h}^{0h}. \quad (3)
\]

where array \( N[] \) is used as a counter.

**Step 4:** Move \( T \) to the left-upper corner of next subimage pair and perform steps 2-3 until all the subimages in \( G_m \) and \( H_m \) are visited.

**Step 5:** \( m = m + 1 \); read the next image pair \( (G_m, H_m) \) and perform steps 2-4 until all image pairs are finished.

**Step 6:** For each encoded index \( I, 0 \leq I < 2^{16} \), perform

\[
\text{LUT}[I] = \frac{\text{LUT}[I]}{N[I]}, \quad (4)
\]

to obtain the mean gray value of \( \text{LUT}[I] \). Consequently, the LUT is constructed.

In the reconstructing process of the LIH, given an input halftone image \( H \), the template \( T \) is still used as a sliding window on \( H \). For each subimage \( S_h \) covered by \( T \), we calculate the mapping address \( I \). According to the value of \( I \), the reconstructed gray value of \( S_{1h}^{0h} \) can be gotten from array \( \text{LUT}[I] \). The gray image can be reconstructed from the input halftone image.

In the LIH, the memory cost is 64 kilobytes. To reduce the memory requirement, Meşe and Vaidyanathan [19] presented an efficient method to construct the TLUT by using the tree-structure approach. In the NNIH proposed by Huang et al. [29], the radial basis function neural network is used to represent the LUT for reconstructing the gray images. Experimental results demonstrate NNIH has better image quality than the LIH.
Based on the parallel process, Siddiqi et al. [30] present the PLIH to speed up the execution-time of LIH. The above LUT-based algorithms reconstructing the gray image by only considering the binary pattern of each halftone subimage. To improve image quality further, the ELIH [20] not only uses the binary pattern of each subimage, but also considers the edge information to improve the image quality.

Although the ELIH achieves better PSNR performance than that of the LIH, it reconstructs the gray value only from a $4 \times 4$ subimage and the pixels out of the subimage can not be considered to improve the image quality further. To achieve more accurate reconstruction, a larger template, for example, the $8 \times 8$ template, can be used to construct the LUT or the ELUT. However, it needs too large memory requirement, $2^{64}$ bytes for LUT and $39 \times 2^{64}$ bytes for ELUT, to be implemented. Under the acceptable memory requirement, the proposed IH which is based on the variance gain-based tree and DCT-based VQ will be presented in next two sections to improve the image quality by using the $8 \times 8$ template.

3. THE PROPOSED VARIANCE GAIN-BASED DECISION TREE

In the first subsection, as a new learning scheme, we propose a variance gain-based DT. Based on the minimal variance criterion, i.e., the maximal variance gain criterion, the input set of training image pairs, (gray image, halftone image)'s, are used to construct a DT. On the constructed DT, each leaf node contains a subset of gray images with less within-variance. In the second subsection, we explain why our proposed variance gain-based DT has better quality for reconstructed image in terms of PSNR when compared to the traditional decision tree ID3 [22-24]. In our experimental results [25], our proposed variance gain-based DT has 0.26 dB PSNR improvement.

3.1 Variance Gain-Based Tree Construction Approach

Considering more pixels, we take an $8 \times 8$ template $T'$ (see Fig. 2) to get $8 \times 8$ training subimage pairs. To build up the variance gain-based DT, the input training image pairs $\{(G_m, H_m) | 1 \leq m \leq M\}$ are given first. For the $m$th, $1 \leq m \leq M$, the halftone subimage of $H_m$ covered by $T'$ is denoted as $S'_h$. $S'_h$ is the 64 binary values of $S'_h$. $S'_h$ is the gray subimage of $G_m$ covered by $T'$, $S'_h$ is the $T'$-mapped gray value of $S'_h$. A pair $(S'_h, S'_3)$ is treated as a sample in the DT construction. The set $\{S'_3 | 0 \leq i < 64\}$ in each sample can be seen as a set of 64 features and the value of each feature is binary. After processing all the $M$ training image pairs, all the obtained samples form the so called sample set $Q$ and the variance of $Q$ is calculated by

$$V(Q) = \frac{\sum_{k=0}^{Q} (Q(k, S'_{36}) - \overline{Q(S'_{36})})^2}{|Q|}, \quad (5)$$

where $|Q|$ denotes the size of $Q$, $Q(k, S'_{36})$ and $\overline{Q(S'_{36})}$ denote the gray value $S'_{36}$ of the $k$th sample in $Q$ and the mean gray value of $Q(k, S'_{36})$, respectively.
Based on $i$th feature $S_i^h$, $0 \leq i < 64$, the sample set $Q$ can be partitioned into two sub-sets $Q^0$ and $Q^1$ where any sample in $Q^0$ ($Q^1$) has $S_i^h = 0$ ($S_i^h = 1$). The binary partition is depicted in Fig. 3. From $Q^0$ and $Q^1$, the variance gain $VG(Q, S_i^h)$ is computed by

$$VG(Q, S_i^h) = \frac{V(Q) - \left(\frac{|Q^0|}{|Q|}V(Q^0) + \frac{|Q^1|}{|Q|}V(Q^1)\right)}{V(Q)},$$

The variance gain $VG(Q, S_i^h)$ can be used to evaluate the quality of the reconstructed gray values resulted by considering $S_i^h$ in the inverse halftoning process. Return to the LUT-based approach, for each binary pattern of training halftone subimages, the corresponding gray values of training gray subimages are collected and the mean of these gray values are calculated as the reconstructed gray value. Thus, for each binary pattern, the mean square error (MSE) between the original gray values and the reconstructed gray values could be estimated by calculating the variance of these collected gray values. Similarly, in the proposed variance gain-based DT, the mean values $\bar{Q}_0^0(S_i^h)$ and $\bar{Q}_1^0(S_i^h)$ are used as the reconstructed gray values for the two cases $S_i^h = 0$ and $S_i^h = 1$, respectively. Thus, the variance gain can help us to select a suitable feature from $S_i^h$'s to partition $Q$ for obtaining better image quality. By considering all 64 features $S_i^h$'s, we have 64 variance gains $\{VG(Q, S_i^h) \mid 0 \leq i \leq 64\}$. For the argument $j$ satisfying

$$j = \arg\max_j (VG(Q, S_j^h)), \quad 0 \leq j \leq 64$$

we use the feature $S_j^h$ to partition the sample set $Q$ into two subsets $Q^0$ and $Q^1$ such that the average variance of the two partitioned subsets is minimal among the concerned 64 possible partitions. For each of $Q^0$ and $Q^1$, the above partition process is performed iteratively until the variance gain less than the threshold $T_V$. Empirically $T_V$ is set to 2.5%.

In the reconstructing process, given an input halftone image $H$, the template $T'$ is used as a sliding window on $H$. For each subimage $S^h$ covered by $T'$, according to its binary pattern, we traverse the DT from top to bottom until a leaf associated with a mean gray value is reached. Continuing this way, the gray image can be reconstructed from the input halftone image and this reconstructed gray image is only a preliminary reconstructed image. Based on the above constructed variance gain-based DT, section 4 will further propose a new texture-based learning scheme to classify the subset of gray images attached in each leaf node into smaller and finer subsets, each subset with similar texture. This classification process leads to build up the lookup tree-table which is used to reconstruct the final high-quality gray images.
3.2 The Quality Advantage of the Proposed Variance Gain-Based DT vs. ID3

In this subsection, we will explain why our proposed variance gain-based DT has better quality for reconstructed image in terms of PSNR when compared to the entropy approach to construct the traditional decision tree ID3 [22-24]. In the entropy-based ID3 method, the minimal entropy criterion is used to build up the decision tree and each leaf node on the constructed tree contains a subset of samples with less entropy. We now take two simple examples to clarify the quality advantage of our proposed variance gain-based DT when compared to the entropy-based ID3 method.

In the training process of the ID3 method, let the generated leaf node be denoted by \( L_1 \) and let \( Q_1 \) denote the attached sample set in \( L_1 \). In the first example, assume \( Q_1 \) is composed of eight samples and the gray values of these eight samples are 8, 8, 8, 8, 8, 16, and 16. The sample set \( Q_1 \) contains two different gray values, 8 and 16, and the entropy of \( Q_1 \) can be calculated by
\[
H(Q_1) = -p_8 \log_2 p_8 - p_{16} \log_2 p_{16} = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8} = 0.8113
\]
where \( p_8 = \frac{6}{8} \) and \( p_{16} = \frac{2}{8} \) denote the probabilities of the gray values 8 and 16 in \( Q_1 \), respectively. The mean gray value of \( Q_1 \) (= 10) is used as the reconstructed gray value for each binary pattern in \( L_1 \). To estimate the resultant reconstruction error due to using the mean gray value 10 to reconstruct the gray value of each binary pattern in \( L_1 \), we replace the gray values of all samples in \( Q_1 \) by the mean gray value 10 to obtain a reconstructed sample set \( Q'_1 \). The mean square error (MSE) between the gray values of \( Q_1 \) and \( Q'_1 \) is 12 (= (6 \times (8 - 10)^2 + 2 \times (16 - 10)^2)/8) and it can be used to estimate the reconstruction error.

Further, let \( L_2 \) denote another leaf node and let the attached sample set be \( Q_2 \). For ease of exposition, still assume that \( Q_2 \) contains eight samples and the eight gray values are 8, 8, 8, 8, 8, 120, and 120. It is easy to verify that the entropy of \( Q_2 \) is equal to that of \( Q_1 \). Although \( Q_1 \) and \( Q_2 \) have the same entropy, they have quite different MSEs. It is known that the MSE of \( Q_1 \) is 12. However, the MSE of \( Q_2 \) is 2352 = (6 \times (8 - 36)^2 + 2 \times (120 - 36)^2)/8.

Due to without considering the distribution of the gray values in each sample set, the above two sample sets reveal that the entropy-based ID3 method may lead to large MSE although the two sample sets have the same entropy. In fact, the MSE between the sample set \( Q \) and the reconstructed sample set \( Q' \) is equivalent to the variance of the gray values of \( Q \). In terms of MSE, PSNR is the most popular image quality measure and it is defined as follows:
\[
\text{PSNR}(G, G') = 10 \log_{10} \frac{255^2}{\text{MSE}(G, G')}
\]
where \( G \) and \( G' \) denote the original gray image and the reconstructed gray image, respectively; MSE \((G, G')\) denotes the MSE between \( G \) and \( G' \). From the definition of the PSNR, we can find that higher MSE results in smaller PSNR and this property implies that the entropy-based ID3 method may lead to poor image quality. In our proposed variance gain-based DT construction procedure, the goal is to minimize the variance, i.e., minimize MSE, and thus the PSNR can be maximized. Therefore, our proposed variance gain-based DT has better quality for reconstructed images in terms of PSNR when compared to the en-
tropy-based ID3 method. In our experimental results [25], our proposed variance gain-based DT has 0.26 dB PSNR improvement.

4. THE PROPOSED TEXTURE- AND VARIANCE GAIN-BASED IH ALGORITHM

In this section, our proposed texture-based training process is first presented to construct a lookup tree-table for reconstructing high-quality gray images. In the lookup tree-table, the tree contains an approximate DT created by the proposed variance gain-based partition strategy and the table contains a local codebook generated by performing the DCT-based vector quantization (VQ) on the sample set attached to each leaf node of the approximate DT. Further, based on the constructed lookup tree-table, the reconstruction process is presented to reconstruct the gray image from the input halftone image.

4.1 The Construction of Lookup Tree-Table

In section 3, the gray images reconstructed by the proposed variance gain-based DT is only the preliminary reconstructed images. Although the image quality of the preliminary reconstructed images can be improved by maximizing the variance gain at each partition step, the variance of the sample set attached to a leaf node may still be some large even if the maximal variance gain is reached. Since the training samples attached to each leaf node have been classified according to the binary features of the halftone subimages $S_h$'s, we further consider the other features which can be classified from the samples attached to each leaf in order to improve the quality of the preliminary reconstructed images. Before doing that, to construct an approximate DT, we stop the growing of DT mentioned in section 3.1 when the number of samples in each leaf node of the current DT is less than 200. Our stopping rule leads to that the size of sample set attached to each leaf node is large enough to contain more varied textures for texture classification.

Once the approximate DT has been built up, we further want to classify the sample set in each leaf node, and then create a dynamic local codebook which is used to store the typical feature vectors of this sample set. Before describing our proposed DCT-based approach to create the dynamic local codebook for each leaf node, we explain why we don’t use the edge classification method in the ELIH to classify the sample set. Let us revisit the ELIH and we know that there are thirty four regular edge types and five irregular edge types. Based on natural images, surprisingly, our experimental results demonstrate that under 12 percent of all edge patterns are regular and over 88 percent of all edge patterns are irregular. Unfortunately, the ELIH only considers five irregular types and each type is roughly determined by the number of edge pixels in the subimage, but the ELIH doesn’t endeavor to classify the dominant irregular edge patterns into meaningful patterns. Therefore, instead of using the edge classification method in the ELIH for each leaf node in the approximate DT, we propose a DCT- and VQ-based approach to classify the sample set of each leaf node into more detailed patterns. Plugging the proposed variance gain-based DT and the DCT- and VQ-based approach into our IH algorithm can obtain better image quality effect when compared to the ELIH.

In our proposed IH algorithm, the texture classification is performed in the DCT do-
main by using the VQ approach. The 2-D DCT [9] of an input \( N \times N \) gray image is defined by

\[
F(i, j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left( \frac{(2x+1)i\pi}{2N} \right) \cos \left( \frac{(2y+1)j\pi}{2N} \right)
\]

where \( 0 < i \leq N \) and \( 0 < j \leq N \). When \( i = 0 \) \((i \neq 0)\) we set \( C(i) = \frac{1}{\sqrt{2}} \) \((C(i) = 1)\) and when \( j = 0 \) \((j \neq 0)\), we set \( C(i) = \frac{1}{\sqrt{2}} \) \((C(j) = 1)\). The value of \( F(0, 0) \) \((i = 0 \text{ and } j = 0)\) is called the DC coefficient while the others are called the AC coefficients. Fig. 4 (a) shows the 64 basis vectors for \( 8 \times 8 \) DCT. Statistical data shows that the first 16 basis vectors are the most widely used basis vectors and Fig. 4 (b) illustrates these 16 basis vectors. Running these 16 basis vectors in Fig. 4 (b) on each \( 8 \times 8 \) gray sub-image in the leaf node, the generated 16 coefficients (one DC coefficient and fifteen AC coefficients) constitute a \( 1 \times 16 \) feature vector. For all samples in each leaf node, these generated feature vectors constitute a training vector set which will be further classified via the VQ approach to build up a dynamic local codebook.

To classify the training vector set, the pairwise nearest neighbor (PNN) classification method [27] is adopted. At beginning, the PNN algorithm treats each training vector as a group. In each merging step, the PNN method merges two groups into a new group when the variance of this new group is the minimum among all possible merging pairs. This merging operations are performed iteratively until the minimum variance of all possible merging pairs is larger than the specified threshold which is set to 20000 in our experiment. Then, the mean vectors of the groups are output as the codebook of the current leaf node.

Once the codebook for each leaf node is generated, the samples in the current leaf node can be clustered if they have the same corresponding codewords. The mean gray value of each resultant group is reconstructed as the reconstructed value and stored in the codebook. From the approximate DT and the codebook attached to each leaf node, a lookup tree-table has been constructed and it can be used to reconstruct a gray image from an input halftone image according to the reconstructing process which will be described in next subsection. Here, the lookup tree-table used in our proposed scheme is composed of two parts, namely the tree part for storing the approximate DT and the array part for recording these dynamic local codebooks associated with the mean gray values. Experimental results [25] show that we have 1.18 dB average PSNR improvement when applying the proposed texture- and VQ-based learning scheme to the approximate DT.
4.2 The Reconstructing Process

In this subsection, based on the constructed lookup tree-table, the reconstruction process of the proposed texture- and variance gain-based IH algorithm is presented to reconstruct the gray image from the input halftone image. To obtain the texture information which is used as the key to select the best reconstructed value from the local codebook of each leaf node, a temporary gray image $G_T$ is constructed by applying the Gaussian filter to the input halftone image $H$. Empirically, the standard deviation used in the Gaussian filter is set to 1.25. Then, the template $T'$ is used as a sliding window to operate on $H$ and $G_T$. According to the binary pattern of each subimage $S'$ covered by $T'$, we traverse the approximate DT from top to bottom until a leaf node is reached. In order to select the best reconstructed value from the local codebook of the reached leaf node, the selected 16 DCT basis vectors are operated on the subimage $S'$ to generate a $1 \times 16$ feature vector. Using this $1 \times 16$ feature vector as the key, a nearest codeword with the reconstructed gray value can be obtained from the local codebook. We repeat the above way, until the whole gray image is reconstructed.

In the above reconstruction process, the reconstruction quality may be degraded when the input subimage has non-smooth contents since some pixels in the subimage may have no strong relation with the estimated pixel, but they are still considered in the construction of our proposed approximate DT. In what follows, an edge-based refinement scheme is presented to alleviate this problem and improve the reconstructed image quality further.

In order to examine the edge type of each temporary gray subimage $S'$, we use DCT coefficients of $S'$ to compute the AC energy $E_a$, the vertical energy $E_v$, and the horizontal energy $E_h$ [9] as follows:

$$E_a = \sum_{i=0}^{3} \sum_{j=0}^{3} |F(i,j)| - |F(0,0)|$$
$$E_v = |F(1,0)| + |F(2,0)| + |F(3,0)|$$
$$E_h = |F(0,1)| + |F(0,2)| + |F(0,3)|$$

where the DCT coefficient $F(i,j)$ can be calculated by Eq. (9); $|F(i,j)|$ denotes the magnitude of $F(i,j)$. If the value of $E_a$ is less than the threshold $T_a$, $S'$ is a smooth subimage; otherwise, it is a non-smooth subimage and will be further examined by using $E_v$ and $E_h$. Empirically $T_a$ is set to 250. For each non-smooth subimage, if $E_v > E_h$ and $E_v/(E_h + 1) > T_h$, we say the subimage is a vertical edge-type subimage (VES); if $E_v < E_h$ and $E_h/(E_v + 1) > T_h$, we say that it is a horizontal edge-type subimage (HES); otherwise, it is a diagonal edge-type subimage (DES). The threshold $T_h$ is set to 1.25 empirically.

From the above three edge-types, the template $T_v'$, $T_h'$, or $T_d'$ depicted in Fig. 5 is selected to help us to traverse the approximate DT for finding a small set of leaf nodes to determine a more satisfactory reconstructed gray value when the subimage $S'$ is the VES, the HES, or the DES, respectively. For each non-smooth subimage, instead of the pixels covered by the template $T'$, only the pixels contained by the template $T''$, where $T'' \in \{T_v', T_h', T_d'\}$ is selected according to the edge-type, are considered to traverse the approximate DT. As shown in Fig. 6, suppose that $P$ is the currently visited node and the two nodes
Fig. 5. Three templates for different edge-type subimages.

Fig. 6. The proposed edge-based refinement scheme.

\( P_L \) and \( P_R \) are the left child node and the right child node of \( P \), respectively. Let \( S_P \in \{ S_i \mid 0 < i \leq 64 \} \) denote the branch controlling feature of \( P \) to determine the next visited node. For the case that the input subimage is a smooth subimage, the next visited node is \( P_L \) (\( P_R \)) for \( S_P = 0 \) (\( S_P = 1 \)). However, for the non-smooth subimage, the above branching process is performed only if \( S_P \) is contained by \( T'' \). If \( S_P \) is not contained by \( T'' \), both \( P_L \) and \( P_R \) are needed to be visited since the determination of the next visited node is unrelated to the binary value of \( S_P \). Since both two child nodes are visited, a set of leaf nodes will be reached after we traverse the approximate DT. Assume \( k \) matched leaf nodes have been reached, and then \( k \) attached codebooks can be used to determine \( k \) reconstructed gray values. The mean value of these \( k \) reconstructed gray values is used as the final estimated gray value of the input halftone image.

Consequently, the above edge-based refinement scheme not only can narrow the range of concerned pixels, but also can improve the quality of the reconstructed images. Experimental results [25] show that we have 0.2 dB average PSNR improvement when applying the proposed edge-based refinement scheme to the reconstruction process.

5. EXPERIMENTAL RESULTS

In this section, some experimental results are demonstrated to show the quality performance comparison among the concerned four IH algorithms, the LIH [17, 18], the ELIH [20], the NNIH [29], and our proposed IH algorithms. Our proposed IH algorithm utilizes the 8 \( \times \) 8 templates to construct the lookup tree-table. The template size used in the LIH and the ELIH is 4 \( \times \) 4, but not 8 \( \times \) 8 since it needs too large memory requirement,
2^{64} bytes for the LUT and 39 × 2^{64} bytes for the ELUT, to be implemented. For the NNIH, a 7 × 7 template is used in the 49-49-1 radial-basis function neural network for realizing the LUT for recovering gray images. All the concerned experiments are performed on the IBM compatible Pentium IV microprocessor with 3.2 GHZ and 1 GB RAM. The operating system is MS-Windows XP and the program developing environment is Borland C++ Builder 6.0. The PSNR defined in section 3.2 is used to evaluate the quality of the reconstructed gray image.

In our experiment, thirty images, each with size 768 × 512, in the Meşe website’s training image set are used as training images. After creating the LUT, the ELUT, the MELUT, and our proposed lookup tree-table, we pick up each of the training images as the testing image to evaluate the quality performance comparison among the three concerned IH algorithms. The average PSNRs of the thirty reconstructed gray images by the LIH, the ELIH, the NNIH, and the proposed IH algorithm are 27.1, 27.3, 27.2, and 28.2, respectively. The proposed IH algorithm has the best image quality performance among the four concerned IH algorithms.

Further, besides the thirty training images, five popular 512 × 512 images, Lena, peppers, Barbara, mandrill, and airplane, are used as testing images. Table 1 shows the image quality and execution-time comparison among the four IH algorithms in terms of PSNRs and seconds, respectively. Table 1 demonstrates that the proposed IH algorithm needs longest execution-time, but has highest reconstruction quality for images with smooth, non-smooth, or mixed contents. In average, our proposed IH algorithm has 0.9 dB, 0.4 dB and 0.5 dB improvement when compared to the LIH, the ELIH, and the NNIH, respectively. Besides the PSNR advantage, we further demonstrate the visual effect of our proposed IH algorithm based on the same testing images. For clarification, we examine the shoulder part of Lena image, the stem part of pepper image, the face part of Barbara image, the face part of mandrill image, and the tail part of airplane image. Figs. 7-9 illustrate that our proposed IH algorithm has the best smoothness visual effect, i.e. least amount of noises, among the four concerned IH algorithms. From Table 1, the reconstructed peppers image by the proposed algorithm has the best PSNR performance.

Table 1. Quality and time comparison for five images (PSNR vs. CPU seconds).

<table>
<thead>
<tr>
<th></th>
<th>LIH</th>
<th>ELIH</th>
<th>NNIH</th>
<th>Proposed IH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>30.3</td>
<td>31.0</td>
<td>30.3</td>
<td>31.3</td>
</tr>
<tr>
<td>Pepper</td>
<td>30.1</td>
<td>31.0</td>
<td>30.5</td>
<td>31.5</td>
</tr>
<tr>
<td>Barbara</td>
<td>25.8</td>
<td>25.8</td>
<td>25.7</td>
<td>25.9</td>
</tr>
<tr>
<td>Mandrill</td>
<td>24.1</td>
<td>24.1</td>
<td>24.7</td>
<td>24.7</td>
</tr>
<tr>
<td>Airplane</td>
<td>29.6</td>
<td>30.4</td>
<td>30.8</td>
<td>31.0</td>
</tr>
<tr>
<td>Average</td>
<td>28.0</td>
<td>28.5</td>
<td>28.4</td>
<td>28.9</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>0.08</td>
<td>1.5</td>
<td>1.6</td>
<td>4.4</td>
</tr>
<tr>
<td>Pepper</td>
<td>0.08</td>
<td>1.5</td>
<td>1.6</td>
<td>3.9</td>
</tr>
<tr>
<td>Barbara</td>
<td>0.08</td>
<td>1.5</td>
<td>1.6</td>
<td>4.1</td>
</tr>
<tr>
<td>Mandrill</td>
<td>0.08</td>
<td>1.5</td>
<td>1.6</td>
<td>3.9</td>
</tr>
<tr>
<td>Airplane</td>
<td>0.08</td>
<td>1.5</td>
<td>1.6</td>
<td>3.7</td>
</tr>
<tr>
<td>Average</td>
<td>0.08</td>
<td>1.5</td>
<td>1.6</td>
<td>4.0</td>
</tr>
</tbody>
</table>
Fig. 7. Visual effects for Lena’s shoulder part; (a) Original shoulder part; (b)-(e) The results by using LIH, ELIH, NNIH, and the proposed algorithm, respectively.

Fig. 8. Visual effects for pepper’s stem part; (a) Original stem part; (b)-(e) The results by using LIH, ELIH, NNIH, and the proposed algorithm, respectively.

Fig. 9. Visual effects for airplane’s tail part; (a) Original tail part; (b)-(e) The results by using LIH, ELIH, NNIH, and the proposed algorithm, respectively.

6. CONCLUSION

In this paper, a new high quality inverse halftone algorithm by using variance gain-, texture- and decision tree-based learning approach has been presented. Based on our proposed variance gain-based DT, a texture and variance-based training process is presented to construct a lookup tree-table. In the reconstructing process, we propose an edge-based refinement scheme to reconstruct the high quality gray image based on the lookup tree-table. Under two training and testing sets of images, experimental results show that although our proposed IH algorithm needs more execution-time, it provides better PSNR performance and visual quality when compared to the LIH, the ELIH, and the NNIH.

REFERENCES

2. N. Damera-Venkata, T. D. Kite, and B. L. Evans, “Fast blind inverse halftoning,” in


Kuo-Liang Chung (鍾國亮) received the Ph.D. degree from National Taiwan University. Prof. Chung received the Distinguished Research Award (2004 to 2007) from the National Science Council, Taiwan. He is now a University Chair Professor at National Taiwan University of Science and Technology. His research interests include image/video compression, image/video processing, and multimedia applications.
**Yong-Huai Huang (黃詠淮)** received the Ph.D. degree in Computer Science and Information Engineering from National Taiwan University of Science and Technology. He is now an Assistant Professor at Jinwen University of Science and Technology. His research interests include image processing, image/video compression, and multimedia applications.

**Kang-Chieh Wu (吳亢捷)** received the M.S. degree in Computer Science and Information Engineering from National Taiwan University of Science and Technology. His research interests include image processing and image compression.