Inverse Halftoning Algorithm Using Edge-Based Lookup Table Approach

Kuo–Liang Chung and Shih–Tung Wu

***Abstract*—The inverse halftoning algorithm is used to reconstruct a gray image from an input halftone image. Based on the recently published lookup table (LUT) technique, this paper presents a novel edge-based LUT method for inverse halftoning which improves the quality of the reconstructed gray image. The proposed method first uses the LUT-based inverse halftoning method as a preprocessing step to transform the given halftone image to a base gray image, and then the edges are extracted and classified from the base gray image. According to these classified edges, a novel edge-based LUT is built up to reconstruct the gray image. Based on a set of 30 real training images with both low- and high-frequency contents, experimental results demonstrated that the proposed method achieves a better image quality when compared to the currently published two methods, by Chang *et al.* and Meşe and Vaidyanathan.**

***Index Terms*—Edges, halftone image, inverse halftoning, lookup table (LUT), smooth regions.**

I. Introduction

HALFTONING is a process which transforms gray images into halftone images. It has been widely used in the publishing applications, such as newspapers, books, magazines, etc. Halftone images are typically difficult to manipulate. Many image processing procedures, such as scaling, compression, and enhancement could cause severe image degradation [6]. To enable these kind operations, gray images need to be reconstructed from the halftones through inverse halftoning.

Since there is no way to reconstruct a perfect gray image from the given halftone image, many efficient inverse halftoning algorithms have been developed in the past several years to improve the quality of the reconstructed image. These developed inverse halftoning algorithms include the Gaussian lowpass filtering approach [3], the spatial varying FIR filtering approach [6], the nonlinear filtering technique [10], the maximum *a posteriori* estimation [11], the projection onto convex sets apporach [5], the wavelet approach [13], the vector quantization technique [7], and so on. Recently, based on the lookup table (LUT) approach, three efficient inverse halftoning algorithms [2], [8], [9] were presented. Among the developed LUT-based inverse halftoning algorithms (LIHs), Chang *et al.* [2] presented a hybrid inverse halftoning algorithm which combines the LUT approach and the filtering technique. Meanwhile, Meşe and Vaidyanathan [8] independently presented the same LUT approach to inverse halftoning. Later, Meşe and Vaidyanathan [9] presented a tree structured LIH which further reduces the required memory for LUT.

Based on the recently published LUT technique mentioned above, we first apply the previous LIH as the preprocessing step to transform the given halftone image to a base gray image. Then, the edges of the base gray image are classified into two categories—one consisting of four regular edge types and the other an irregular edge type. Furthermore, we investigate the distribution of the edges and the smooth regions for different binary patterns considered in LIH. Following the investigation, a novel edge-based LUT is built up to improve the quality of the reconstructed base gray image. Finally, the memory requirement in our proposed edge-based LIH (ELIH) is investigated. Experiments on a set of thirty real training images with both low and high frequency contents are performed to investigate the efficiency of the proposed ELIH. Experimental results show that the proposed ELIH achieves a better image quality when compared to the currently published two methods, by Chang *et al.* [2] and Meşe and Vaidyanathan [8].

The rest of this correspondence is organized as follows. In Section II, the previous LIH is introduced. In Section III, the proposed ELIH is first presented, and then its memory reduction is investigated. In Section IV, experiments are performed to demonstrate the quality improvement of the proposed ELIH. Finally, conclusions are addressed in Section V.

II. Past Work

In this section, the previous LIH [2], [8] is revisited. Fig. 1 shows a 4×4 template T with symbol X denoting the estimated 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 given halftone image. For convenience, the constructed gray image is called the base gray image.

Suppose initially we are given a set of M training image pairs $\{(O_m, H_m) \mid 1 \leq m \leq M\}$ where O_m denotes the mth original gray image and H_m the corresponding halftone image of O_m. The training halftone image H_m is obtained by applying the existing halftoning algorithm [12] to the original gray image O_m. The following six-step algorithm is used to build up the LUT.

```
Procedure Building-Up LUT
Step 1. Initiate $LUT[\ ] = 0$ and $N[\ ] = 0$,
  where the array $LUT[\ ]$ is used to map the
  input halftone image to the base gray
```

Manuscript received August 30, 2004; revised September 30, 2004. K.–L. Chung is supported by the National Council of Science of R.O.C. under Contracts NSC91-2213-E011-028 and NSC92-2213-E001-079. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhigang (Zeke) Fan.

The authors are with the Department of Computer Science and Information Engineering National Taiwan University of Science and Technology, Taipei, Taiwan 10672, R.O.C. (e-mail: klchung@cs.ntust.edu.tw).

Digital Object Identifier 10.1109/TIP.2005.854494

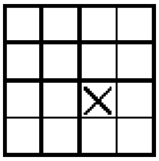
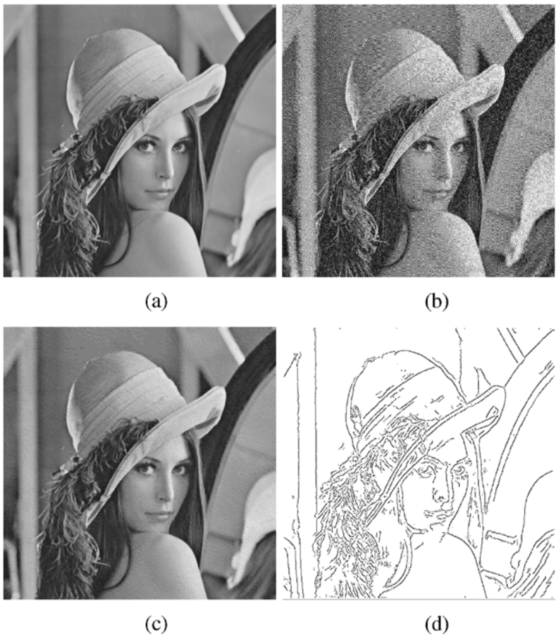


Fig. 1. A 4 × 4 template T.

```
 image and the array $N[\ ]$ will be used in
 Step 3. Set $m = 1$, that is, the first
 image pair of the training images is
 taken into consideration.
Step 2. Put the template $T$ at the
 left-upper corner of the $m$th training
 halftone image $H_m$.
Step 3. Denote $S^h$ the subimage of $H_m$ cov-
 ered by the $4 \times 4$ template $T$. Based on the
 raster scan order, denote $S^h_{15}, S^h_{14}, \ldots, S^h_1$,
 and $S^h_0$ the 16 binary values of $S^h$, and
 encode $S^h$ by the index

   $I = \sum_{k=0}^{15} 2^k S^h_k.$   (1)

 Consequently, at most 65 536 (= $2^{16}$) dif-
 ferent $I$s can be generated and the en-
 coded index $I$ is used as a mapping ad-
 dress of the constructed $\mathrm{LUT}[\ ]$. Denote
 $S^g$ the gray subimage of $O_m$ corresponding
 to $S^h$, and denote $S^g_{15}, S^g_{14}, \ldots, S^g_1$, and $S^g_0$
 the sixteen gray values of $S^g$. For the
 gray subimage $S^g$, $S^g_{10}$ represents the tem-
 plate $T$-mapped gray value. For the en-
 coded index $I$, the following two assign-
 ments are performed:

   $N[I] = N[I] + 1$
   $\mathrm{LUT}[I] = \mathrm{LUT}[I] + S^g_{10}.$

Step 4. In the halftone image domain $H_m$,
 if the template $T$ can be moved one posi-
 tion from left to right legally based on
 the raster scan order, go to Step 3; oth-
 erwise, go to Step 5.
Step 5. Increment $m$ by one. If $m > M$, go
 to Step 6; otherwise, go to Step 2.
Step 6. For each encoded index $I$, $0 \leq I \leq$
 $65\,535$, perform

   $\mathrm{LUT}[I] = \frac{\mathrm{LUT}[I]}{N[I]}$   (2)

 to obtain the mean gray value of $\mathrm{LUT}[I]$.
 Stop this procedure.
```

Following the procedure building-up LUT, the LUT array is constructed accordingly. In what follows, we review the previous LIH [2], [8], which is used to construct the base gray image B from the input halftone image H. The four-step LIH is listed as follows.



(a) (b) (c) (d)

Fig. 2. Obtaining the edge map E from the input halftone image H. (a) The original Lena gray image. (b) The input halftone image H generated by error diffusion with Floyd-Steinberg kernel. (c) Base gray image B. (d) Edge map E.

```
Procedure LIH
Step 1. Call procedure building-up LUT to
 build up the LUT.
Step 2. Put the template $T$ at the left-
 upper corner of $H$.
Step 3. Based on the raster scan order,
 denote $S^h$ the corresponding $4 \times 4$ subimage
 of $H$ covered by $T$, and denote $S^g$ the
 gray subimage of the reconstructed base
 image $B$ corresponding to $S^h$. For the
 gray subimage $S^g$, the template $T$-mapped
 gray value is denoted by $S^g_{10}$. From $S^h$,
 after encoding $I$ by (1), the recon-
 structed gray value is equal $S^g_{10} = \mathrm{LUT}[I]$.
Step 4. In the input halftone image do-
 main $H$, if the template $T$ can be moved
 one position from left to right legally,
 go to Step 3; otherwise, stop this proce-
 dure.
```

In practice, some binary patterns S^hs in the input halftone image H may not exist in the training images $\{H_m \mid 1 \leq m \leq M\}$, i.e., $N[I] = 0$ in (2). Therefore, these nonexistent binary patterns map nothing in the LUT. For predicting the gray value for the nonexistent binary pattern, Chan *et al.* [2] presented an adaptive filtering approach and Meşe and Vaidyanathan [8] adopted the best linear estimator approach. The two approaches have similar quality in the reconstructed gray images.

III. Proposed Edge-Based LUT Inverse Halftoning Algorithm: ELIH

In this section, the proposed ELIH is presented. The analysis of the proposed ELIH, which can produce a better reconstructed

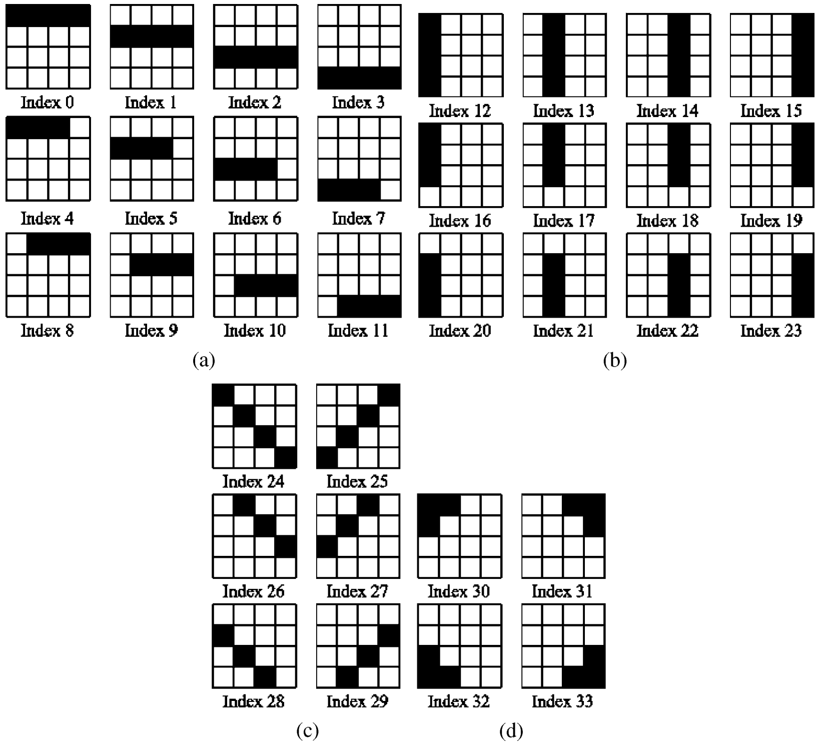


Fig. 3. Category_1: Four regular edge types. (a) Horizontal edge type: Twelve horizontal edge patterns. (b) Vertical edge type: Twelve vertical edge patterns. (c) Diagonal edge type: Six diagonal edge patterns. (d) Corner edge type: Four corner edge patterns.

image quality, is also given. In addition, memory reduction is investigated.

A. *Obtain Edge Maps From Halftone Images*

First, LIH is applied to the halftone image H to construct the base gray image B, and then Canny edge detector [1] is applied to B to obtain the edge map E. Since Canny edge detector can reduce the influence of noise; eliminate the multiple response to a single edge, and has good localization with minimal distance between the detected edge position and the true edge position, the edges in the resulting edge map E are rather thin.

Fig. 2(a) denotes the original Lena gray image. Fig. 2(b) is used as the input halftone image H which is generated from Fig. 2(a) by error diffusion with Floyd-Steinberg kernel [12]. After applying LIH to H, the base gray image B is constructed as shown in Fig. 2(c) which is quite similar to Fig. 2(a) from human visual perception. Finally, the edge map E is illustrated in Fig. 2(d) by applying Canny edge detector to Fig. 2(c).

The similarity between Fig. 2(a) and (c) indicates that LIH is a very efficient way to reconstruct the base gray image B. Since edge detection can be more effectively performed on gray images than on halftones, the edge information is extracted from the base gray image B, instead from the input halftone image H.

B. *Classify Extracted Edges*

From the edge map E [see Fig. 2(d)], each 4×4 subedge map is examined and classified into one of two categories—Category_1 consisting of four regular edge types and Category_2 one irregular edge type.

In Category_1, the four regular edge types are determined according to their appearance frequency from all edge maps which are obtained from the thirty halftone images as the same as those in [8]. The horizontal (vertical) edge type contains twelve edge patterns with indices 0–11 (12–23) as shown in Fig. 3(a) [Fig. 3(b)] where each black pixel denotes an edge pixel. The diagonal edge type contains six edge patterns with indices 24–29, as shown in Fig. 3(c). The corner edge type contains four edge patterns with indices 30–33, as shown in Fig. 3(d).

Other than the regular edge patterns considered in Category_1, all the remaining edge patters are classified into Category_2. Category_2 is partitioned into five groups as shown in Fig. 4. In Fig. 4, group_1 with index 34 contains the edge pattern without any edge pixels. Group_2, group_3, group_4, and group_5 contain edge patterns with 1–4, 5–8, 9–12, and 13–16 random edge pixels, respectively. Based on the thirty training edge maps, group_1 in Category_2 has the highest appearance frequency, so we identify it to a separated group and the remaining edge patterns in Category_2 are evenly partitioned in terms of edge pixel count into four groups. Since the union of Category_1 and Category_2 spans the whole space of all possible 4×4 subedge maps, any input 4×4 subedge map can find its unique mapping index.

C. *Investigate the Distribution of Edges and Smooth Regions for Template T–Mapped Gray Values*

This subsection explains why the proposed ELIH can improve the image quality when compared to LIH. We use the pattern given in Fig. 5 as an example. Although the pattern contains vertical edge information, it can also occur in smooth regions due to limit cycle behavior, as explained by Fan and Eschbach [4]. This is illustrated in Fig. 6(a), which gives the positions that

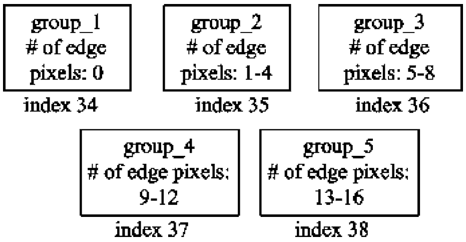


Fig. 4. Category_2: Irregular edge type.

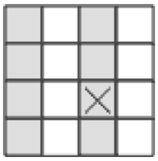


Fig. 5. Exemplar pattern.

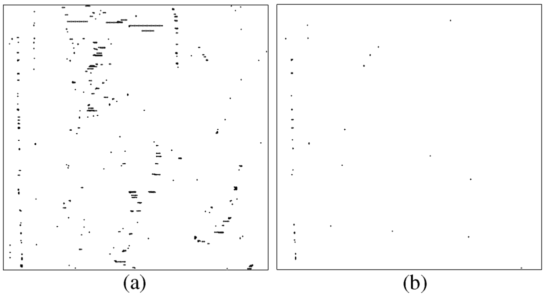


Fig. 6. Matched patterns in Lena halftone image (a) for patterns without considering edge classfication and (b) for patterns associated with the vertical edges (edge types 12–23).

the pattern appears in Lena image. Fig. 6(b) shows the pattern positions associated with the vertical edges (edge types 12–23).

We have observed, even with the same pattern, the associated gray values could have different distributions for different edge types. From Lena image, we collected 716 gray values associated with pattern Fig. 5. They range from 99 to 155. While the overall mean is 128, the mean values for those of edge type 34 (smooth regions), and edge types of 12–23 (vertical edges) are 128 and 134, respectively. Their distributions are given in Fig. 7, where Fig. 7(a) shows the overall distribution, and Fig. 7(b) and (c) depicts the distributions of edge type 34 and edge type 12–23, respectively. The reason that ELIH outperforms LIH is it selects more precise distributions, according to the edge context information.

D. Build Up the ELUT

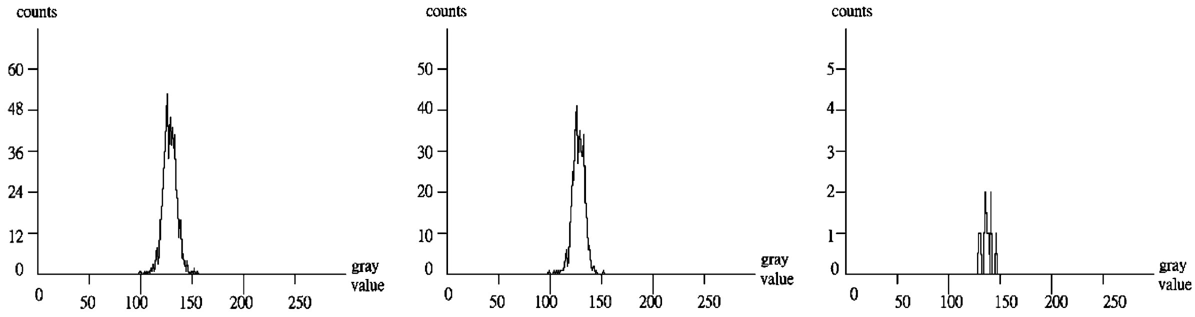
First, this subsection extends the address mapping scheme from the index I to the index pair (I, J). Given an input 4×4 binary pattern, by (1), it is easy to encode the index I; from the corresponding 4×4 subedge map, it takes linear time to obtain its mapping index J by searching the matched edge type in Category_1 or Category_2. Consequently, the encoded index pair (I, J) is used in the address mapping scheme of the proposed ELUT. Based on the index pair generation process, the following procedure is used to build up the proposed ELUT.

```
**Procedure** Building-up ELUT
Step 1. Call procedure building-up LUT.
  For each $H_m$, $1 \leq m \leq M$, call procedure
  LIH to construct the base gray image $B_m$.
  Apply Canny edge detector to $B_m$ to obtain the edge map $E_m$.
Step 2. Set $\mathrm{ELUT}[,] = 0$ and $N[,] = 0$, where
  the array $\mathrm{ELUT}[,]$ is used to map the
  input halftone image to the constructed
  image and the array $N[,]$ will be used in
  Step 4. Set $m = 1$, that is, the first
  training image pair is taken into consideration.
Step 3. Put the template $T$ at the
  left-upper corner of the $m$th training
  halftone image $H_m$.
Step 4. Let the corresponding $4 \times 4$
  subimage of $H_m$ covered by $T$ be denoted
  by $S^h$; let $S^g$ be the gray subimage of $O_m$
  corresponding to $S^h$, and let $S^e$ be the
  subedge map of $E_m$ corresponding to $S^h$.
  For $S^h$, calculate the encoded index $I$ via
  (1). Using the subedge map $S^e$ as a key,
  the index $J$ can be obtained by searching
  Category_1 and Category_2. Denote $S^g_{10}$
  the template $T$-mapped gray value of the
  gray subimage $S^g$. Based on the determined
  index pair $(I, J)$, perform the following
  two assignments:

      $N[I,J] = N[I,J] + 1$
      $\mathrm{ELUT}[I,J] = \mathrm{ELUT}[I,J] + S^g_{10}.$

Step 5. On the halftone image domain $H_m$,
  if the template $T$ can be moved one position
  from left to right legally based on
  the raster scan order, go to Step 4; otherwise,
  go to Step 6.
Step 6. Increment $m$ by one. If $m > M$, go
  to Step 7; otherwise, go to Step 3.
Step 7. For each index pair $(I, J)$, $0 \leq I \leq 65\,535$
  and $0 \leq J \leq 38$, perform

      $\mathrm{ELUT}[I,J] = \frac{\mathrm{ELUT}[I,J]}{N[I,J]}$          (3)

  to obtain the mean gray value of
  $\mathrm{ELUT}[I,J]$. Stop this procedure.
```

E. Proposed ELIH and Memory Reduction Consideration

After building up the ELUT, the five-step ELIH can be performed as follows.

```
**Procedure** ELIH
Step 1. Call procedure building-up ELUT.
  Call procedure LIH to construct the base
  gray image $B$ from the halftone image $H$.
```



(a) (b) (c)

Fig. 7. Distributions of template T-mapped gray values (a) for patterns without considering edge classification, (b) for irregular edge type with index 34, and (c) for regular edge type with indices ranging from 12 to 23.

```
  Running Canny edge detector on $B$, the
  edge map $E$ is obtained.
Step 2. Put the template $T$ at the left-upper corner of $H$.
Step 3. Denote $S^h$ the corresponding $4 \times 4$
  subimage of $H$ covered by $T$. For $S^h$, denote
  $S^g$ the constructed gray subimage,
  and denote $S^e$ the corresponding subedge
  map. Based on $S^h$ and by (1), obtain the
  encoded index $I$. After searching $S^e$ in
  the space of Figs. 3 and 4, the index $J$
  is obtained. The reconstructed gray value
  is equal to $S_{10}^g = \mathrm{ELUT}[I, J]$, where $S_{10}^g$ is
  the template $T$-mapped gray value of the
  gray subimage $S^g$.
Step 4. If $S_{10}^g \neq$ NULL, go to Step 5; otherwise,
  an nonexistent pattern problem
  arises. A two-phase approach is applied
  to solve this nonexistent pattern. In
  the first phase, the value of $\mathrm{LUT}[I]$
  is checked. If $\mathrm{LUT}[I] \neq$ NULL, perform
  $S_{10}^g = \mathrm{LUT}[I]$; otherwise, proceed to the
  second phase. In the second phase, the
  best linear estimator [8] is applied to
  predict the gray value.
Step 5. In the image domain of $H$, if the
  template $T$ can be moved one position
  from left to right legally based on the
  raster scan order, go to Step 3; otherwise,
  stop this procedure.
```

In LIH, the memory required for the LUT is 64 K bytes. ELIH has 39 edge types for each binary pattern. It, therefore, needs 2496 K ($= 39 \times 64$ K) memory. To reduce memory usage, we delete the entries that are not frequently used. From our experiments, we observed that edge types 34, 35, and 36 have the heaviest utilization. As a result, they are not subject to trimming. While for the rest of the edge types, we delete the entry if its count $N[I, J]$ is smaller than a threshold during the training. In our experiment, this threshold was set to 30. This memory reduction scheme leads to a total of 195 K bytes for the rest 36 edge types. Consequently, the total memory requirement is reduced to 387 K ($= 192\,\mathrm{K} + 195\,\mathrm{K}$). A detected entry is treated in the same manner as a missing entry in reconstruction. As mentioned in Step 4 of procedure ELIH, $\mathrm{LUT}[I]$ is assigned to the reconstructed gray value. If the entry $\mathrm{LUT}[I]$ does not exist either, the best linear estimator is used [8].

IV. Experimental Results

In this section, some experimental results are demonstrated to justify the image quality improvement of the proposed ELIH when compared to LIH. In this correspondence, the peak signal-to-noise ratio (PSNR) is used to evaluate the image quality and seven images, namely airplane, Barbara, boat, goldhill, Lena, mandrill, and peppers are used as our testing images. Three training sets are taken into consideration. The first training set, which is the thirty images containing both low and high frequency contents, comes from Meşe's website [8], [9]. In Table I, $\mathrm{LIH}_{(L+H)}$ and $\mathrm{ELIH}_{(L+H)}$ illustrate the PSNR performance of the reconstructed images using LIH and the ELIH, respectively, where the subscript $(L + H)$ denotes the first training set containing both low and high frequency contents. The average PSNR improvement of the proposed ELIH is 0.45 ($= 28.9 - 28.45$) dB higher than that of LIH. In the same table, LIH_L and ELIH_L illustrate the PSNR performance using LIH and ELIH, respectively, using the second training set of five images that contains mainly smooth regions, namely airplane, boat, goldhill, Lena, and peppers. The fourth and fifth columns in Table I indicate that for the second training set, the proposed ELIH has 0.45 ($= 29.13 - 28.68$) dB image quality improvement in average. The third training set is composed of two images, namely Babara and mandrill. Both contain more high frequency contents. The sixth and seventh columns in Table I indicate that the proposed ELIH has 0.51 ($= 28.67 - 28.16$) dB image quality improvement in average.

Besides using the PSNR to evaluate the image quality performance, the visual comparison is also given to demonstrate the image quality improvement. Fig. 8(a) and (b) are two magnified images with respective to Lena's shoulders which are reconstructed by using LIH and the proposed ELIH, respectively. Edge jaggedness can be observed in Fig. 8(a). This is particular obvious at the boundaries of the shoulder and face. In comparison, Fig. 8(b) showed much smooth boundaries. Most of the edge jaggedness has been removed.

TABLE I
QUALITY IMPROVEMENT FOR SEVEN TESTING IMAGES IN TERMS OF PSNR

	$\text{LIH}_{(L+H)}$	$\text{ELIH}_{(L+H)}$	LIH_L	ELIH_L	LIH_H	ELIH_H
airplane	29.59	30.39	30.92	31.21	28.73	29.4
Barbara	25.84	25.84	24.98	25.63	26.71	26.99
boat	29.15	29.47	29.84	30.2	28.48	28.76
goldhill	30.12	30.51	30.56	30.79	29.47	30.08
Lena	30.34	31.02	31.1	31.34	29.71	30.28
mandrill	24.07	24.08	22.56	23.7	24.83	25.16
peppers	30.07	30.97	30.77	31.04	29.19	29.97
average	28.45	28.9	28.68	29.13	28.16	28.67

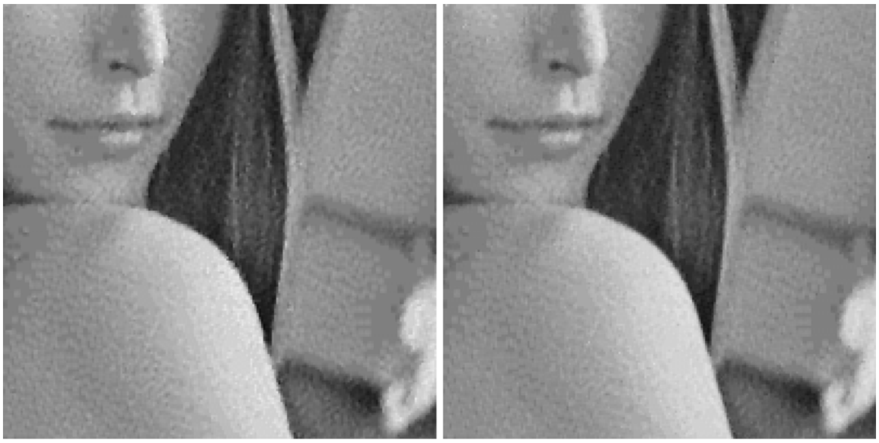


Fig. 8. Two magnified images with respective to Lena's shoulders. (a) After running LIH. (b) After running the proposed ELIH.

V. CONCLUSION

Based on the previously published LUT technique in [2] and [8], we have presented the edge- and LUT-based inverse halftoning algorithm. Due to employing the edge-based approach to enhance the edge effect in the reconstructed gray image, under different kinds of training sets and testing sets, experimental results have demonstrated that the proposed ELIH has a better image quality when compared to the previous LIH.

How to build up more versatile edge types than those in Figs. 3 and 4 is an interesting research issue.

ACKNOWLEDGMENT

The authors would like to thank three referees, Prof. W.–N. Yang, and F.–Y. Wang for their valuable comments that lead to the improved presentation of this correspondence.

REFERENCES

[1] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 11, pp. 679–698, Nov. 1986.

[2] P. C. Chang, C. S. Yu, and T. H. Lee, "Hybrid LMS–MMS inverse halftoning technique," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 95–103, Jan. 2001.

[3] N. Damera-Venkata, T. D. Kite, and B. L. Evans, "Fast blind inverse halftoning," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Oct. 1998, pp. 64–68.

[4] Z. Fan and R. Eschbach, "Limit cycle behavior of error diffusion," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Nov. 1994, pp. 1041–1045.

[5] S. Hein and A. Zakhor, "Halftone to continuous–tone conversion of error–diffusion coded images," *IEEE Trans. Image Process.*, vol. 4, no. 2, pp. 208–216, Feb. 1995.

[6] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, "A fast, high-quality inverse halftoning algorithm for error diffused halftones," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1583–1592, Sep. 2000.

[7] Z. C. Lai and J. Y. Yen, "Inverse error-diffusion using classified vector quantization," *IEEE Trans. Image Process.*, vol. 7, no. 12, pp. 1753–1758, Dec. 1998.

[8] M. Meşe and P. P. Vaidyanathan, "Look up table (LUT) method for inverse halftoning," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1566–1578, Oct. 2001.

[9] ——, "Tree-structured method for LUT inverse halftoning and for image halftoning," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 644–655, Sep. 2002.

[10] M. Y. Shen and C.-C. J. Kuo, "A robust nonlinear filtering approach to inverse halftoning," *J. Vis. Commun. Image Represen.*, vol. 12, pp. 84–95, Mar. 2001.

[11] R. L. Stevenson, "Inverse halftoning via MAP estimation," *IEEE Trans. Image Process.*, vol. 6, no. 4, pp. 574–583, Apr. 1997.

[12] R. A. Ulichney, *Digial Halftoning*. Cambridge, MA: MIT Press, 1987.

[13] Z. Xiong, M. T. Orchard, and K. Ramchandran, "Inverse halftoning using wavelets," *IEEE Trans. Image Process.*, vol. 8, no. 10, pp. 1479–1482, Oct. 1999.

Kuo–Liang Chung received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1982, 1984, and 1990, respectively.

From 1984 to 1986, he completed his military service. From 1986 to 1987, he was a Research Assistant with the Institute of Information Science, Academic Sinica. He has been a Chairman with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, since August 2003.

Prof. Chung received the Distinguished Research Award (2004 to 2007) from the National Science Council, Taiwan. His research interests include image/video compression, image/video processing, pattern recognition, coding theory, algorithms, and multimedia applications.

Shih–Tung Wu received the M.S. degree in electrical engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C., where he is currently pursuing the Ph.D. degree in the Department of Computer Science and Information Engineering.