Multiscale Probabilistic Dithering for Suppressing Contour Artifacts in Digital Images

IEEE Transactions on Image Processing,
Vol. 18, No. 9, 2009

Sitaram Bhagavathy, Joan Llach, and Jiefu Zhai

Presented by Il-Su Park

School of Electrical Engineering and Computer Science
Kyungpook National Univ.

Abstract

- Proposed method
  - Reduction in visibility of contour artifacts
  - Procedure
    - Multiscale analysis on neighborhood of each pixel
    - Determining presence and scale of contour artifacts
    - Probabilistic dithering
  - Output image
    - Use same or higher bit depth as input image

Introduction

- Contour artifacts
  - Requantization to reduce bit depth
    - Displaying 10-bpp image on 8-bpp display
      - Compress with 8-bpp encoder
  - Bit depth reduction
    - Smaller number of available colors in palette
    - Impossible to represent some areas
    - Some bands of constant color

- Contour artifact removal (CAR) methods
  - Apply before or during requantization process
    - Adding noise to image prior to quantization
    - Error diffusion methods
    - Feedback-based quantization strategy
  - Apply after quantization process
    - Reducing visibility of contour artifacts

- Previous methods
  - Lee et al.
    - Two-stage false contour detection algorithm
      - » Eliminating smooth regions
      - » Separating false contours from edge and texture
        - » Use of directional contrast features
    - False contour reduction algorithm
      - » Using 1-D directional smoothing filters
  - Ahn et al.
    - Detection of flat region or regions of low frequency content
      - » Higher likelihood of containing contour artifacts
    - Bit-depth extension for removing false contours

- Daly et al.
  - Predictive cancellation algorithm
    - » False contours reduction without adding noise or dither to image
    - » Desirable in case of noise-free images
- Daly's method
  - Multiband coring algorithm
    - » Separation of low-pass and high-pass components
    - » Removing low amplitudes in high-pass band

- Problem of previous methods
  - Bit depth of output greater than bit depth of input
    - Use of smoothing filters for reducing false contours
  - No address issue for scale of contour artifacts
    - Scale of contour artifacts
      - » Degree of proximity between adjacent false contours in local region
- Proposed method
  - Multiscale analysis on neighborhood of each pixel
  - Determining presence and scale of contour artifacts
  - Probabilistic dithering method based on distribution of colors in neighborhood

Multiscale contour artifact detection

- Contour artifact detection
  - First step in contour artifact reduction
  - Detecting presence and scale of contour artifact in neighborhood of each pixel
  - Confidence score
    - Likelihood of contour artifacts

$$c(s) = p(0,s) \times MAX\left[\frac{p(-1,s)}{p(0,s)+p(-1,s)}, \frac{p(1,s)}{p(0,s)+p(1,s)}\right] \quad (1)$$

where $p(k,s)$ refers to the fraction (or probability) of all pixels in $N_s(x,y)$ having the value $I(x,y)+k$,
$MAX[a,b]$ refers to the greater of a and b .

– Criteria on probabilities, $p(k,s)$

- Excepting regions with low likelihood of contour artifacts
- Detection of contour artifacts at pixel (x,y)

$$p(0,s) > T \; and \left[p(-1,s) > T \; or \; p(1,s) > T \right] \tag{2}$$

where $T \in (0, 0.5)$ is a preset threshold (typically, $T \geq 0.1$).

- Representative scale, $s^*(x,y)$
  – Highest confidence score among all scales

– Detection scheme works for different type

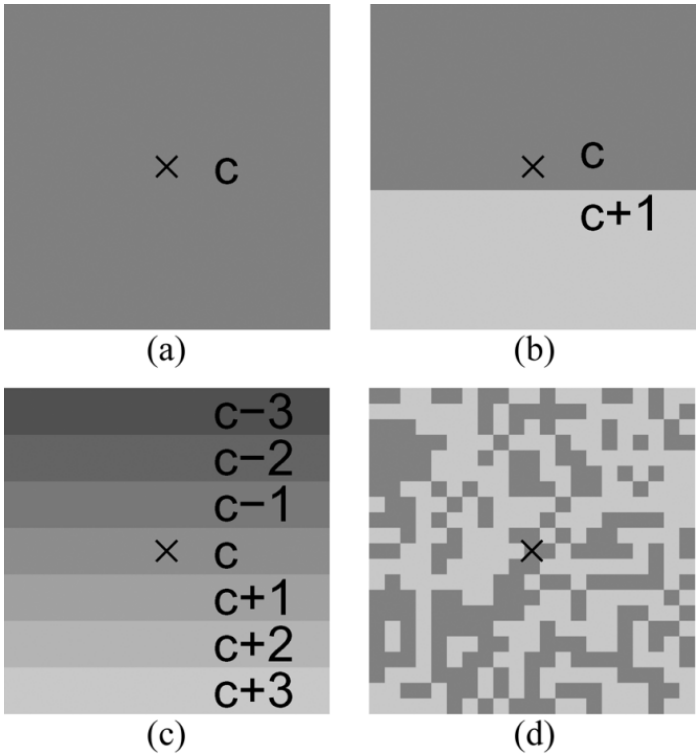


Fig. 1. Illustration of how contour artifact detection works for different types of image neighborhoods (centers marked by “X”): (a) a flat region of color c, (b) a step from color c to c+1, (c) a gradual slope from color c-3 to c+3, and (d) a texture formed by two colors c and c+1.

- Contour artifact detection with spatial constraints
  - Reducing false detection in textured areas
    - Need for spatial constraints
      - Formulation in terms of a threshold τ_g on gradient magnitude at each pixel

– Computation of probabilities in (1) and (2)

$$p(k,s)=\frac{\sum_{\{(x',y')\in N_s(x,y)\,|\,\|\nabla(x',y')\|<\tau_g\}}\delta\left(I\left(x',y'\right),I\left(x,y\right)+k\right)}{\sum_{\{(x',y')\in N_s(x,y)\,|\,\|\nabla(x',y')\|<\tau_g\}}1} \qquad (3)$$

where $\|\nabla(x',y')\|$ is the gradient magnitude at pixel (x',y')
$\delta(.,.)$ is an indicator function defined as follows:

$$\delta(a,b)=\begin{cases}1, & \text{if } a=b\\ 0, & otherwise\end{cases} \qquad (4)$$

$$\|\nabla(x,y)\|=\sqrt{\left(\frac{\partial I(x,y)}{\partial x}\right)^2+\left(\frac{\partial I(x,y)}{\partial y}\right)^2}$$

Contour artifact reduction by probabilistic dithering

- Probabilistic dithering
  - Destruction of false contours
  - Reducing visibility of contours artifacts
    - Use of distribution of color values in local neighborhood

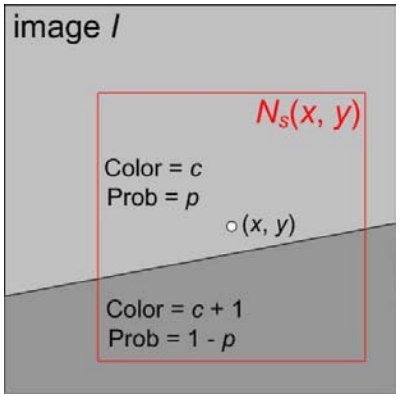


Fig. 2. Illustration of the principle of the probabilistic dithering method used for contour artifact reduction.

– Result of applying probabilistic dithering to image region containing false contour

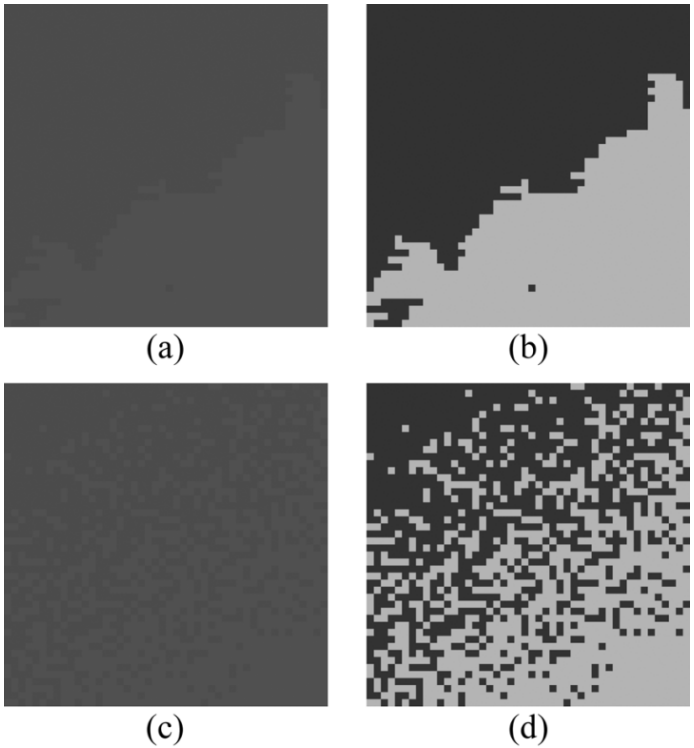


Fig. 3. Effect of probabilistic dithering on an image region: (a) a small region (46x46 pixels) containing a false contour from the grayscale image in Fig. 5(e), (b) binary representation of the region, (c) the region after applying probabilistic dithering to the whole image, and (d) binary representation of the dithered region.

◆ Probabilistic dithering method 1

– Simple extension of basic ideas

- Use of three probabilities
- Normalize probabilities $p(k, s^*)$

$$p'\left(k, s^*\right) = \frac{p\left(k, s^*\right)}{p\left(-1, s^*\right) + p\left(0, s^*\right) + p\left(1, s^*\right)}, \quad k \in \{-1,\ 0,\ 1\} \tag{5}$$

- Assignment to output $J(x, y)$

$$p'(0, s^*) \rightarrow I(x, y), \quad p'\left(-1, s^*\right) \rightarrow I(x, y) - 1, \quad p'\left(1, s^*\right) \rightarrow I(x, y) + 1$$

$$J(x, y) = \begin{cases} I(x, y), & r < p'\left(0, s^*\right) \\ I(x, y) + 1, & p'\left(0, s^*\right) \le r < p'\left(0, s^*\right) + p'\left(1, s^*\right) \\ I(x, y) - 1, & r \ge p'\left(0, s^*\right) + p'\left(1, s^*\right) \end{cases} \tag{6}$$

Where $r \in [0,1]$.

- Probabilistic dithering method 2
  - Dithering pixel based on expected mean value
    - Computation of expected mean color value

$$m = p'\left(-1, s^*\right)\left(z-1\right) + p'\left(0, s^*\right)z + p'\left(1, s^*\right)\left(z+1\right) \qquad (7)$$

where z is $I(x, y)$, and
$p'\left(k, s^*\right)$ are the normalized probabilities computed in (5).

  - Ideal output value $J\left(x, y\right) = m$

– Dithering strategy

- Neighborhood mean close to m
  - Lower value as $\lfloor m \rfloor$
  - Upper value as $\lfloor m \rfloor + 1$
  - Probability $q = m - \lfloor m \rfloor$

    where $\lfloor m \rfloor$ refers to the largest integer not greater than m.
  - Assignment to output value $J(x, y)$

$$J(x, y) = \begin{cases} \lfloor m \rfloor + 1, & \text{if } r < q \\ \lfloor m \rfloor, & \text{if } r \geq q \end{cases} \tag{8}$$

    » Maximum value, $2^{b_I} - 1$

    » Lower value, 0

- Bit depth extension
  - Reverse process of bit depth reduction
    - Increase of bit depth
  - Additional bits
    - Effective suppression of existing contour artifacts
  - Modification of probabilistic dithering method
    - Output value of no contour artifact

$$J(x,y) = d_{\min} I(x,y)$$

where $d_{\min}$ is $2^{b_J - b_I}$.

    - Computation of expected mean value

$$m = d_{\min}\left[p'\left(-1, s^*\right)(z-1) + p'\left(0, s^*\right) z + p'\left(1, s^*\right)(z+1) \right] \quad (9)$$

Experimental Results

- Comparison with predictive cancellation decontouring method
  - Simple synthetic example
    - 6 bits-per-pixel grayscale image
    - Set of intensity bands of varying width or scale
  - Few real-world examples
    - 6 bits-per-component color image

– Simple synthetic example

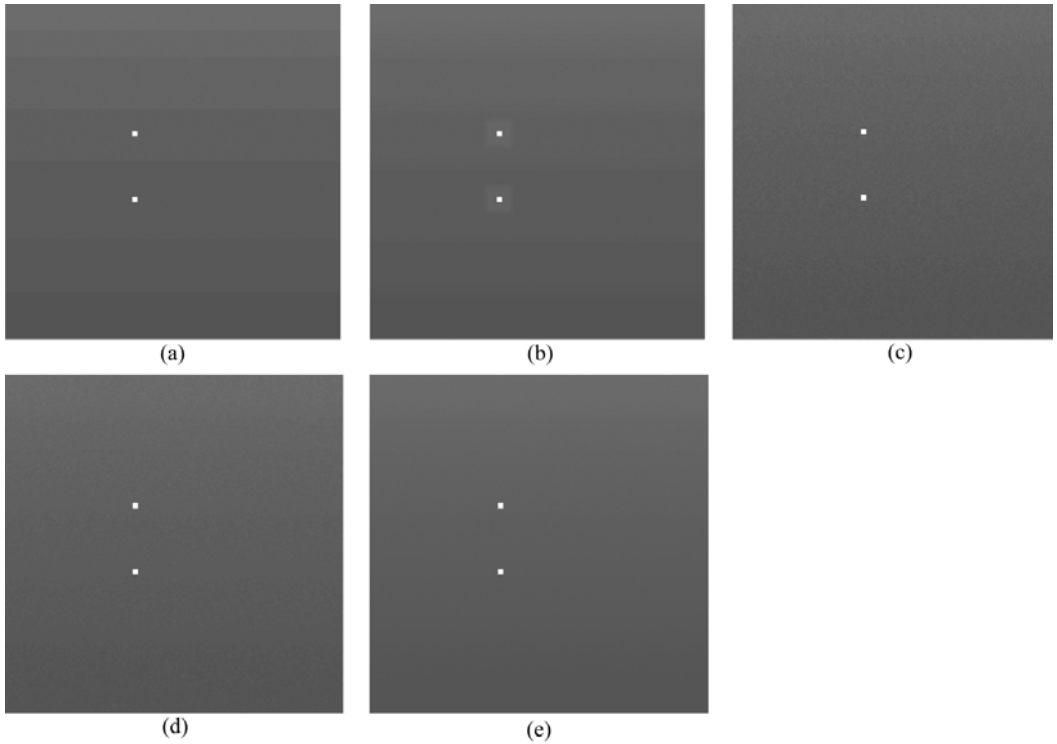


Fig. 4. (a) Input 6 bits-per-pixel (bpp) image with contours and small details (white dots), (b) result of the “decontouring” method of Daly *et al.* at 8 bpp, (c) result of the method proposed in 1 at 6 bpp, (d) result of the method proposed in 2 at 6 bpp, and (e) result of the proposed bit depth extension method at 8 bpp.

– Intensity average of each row

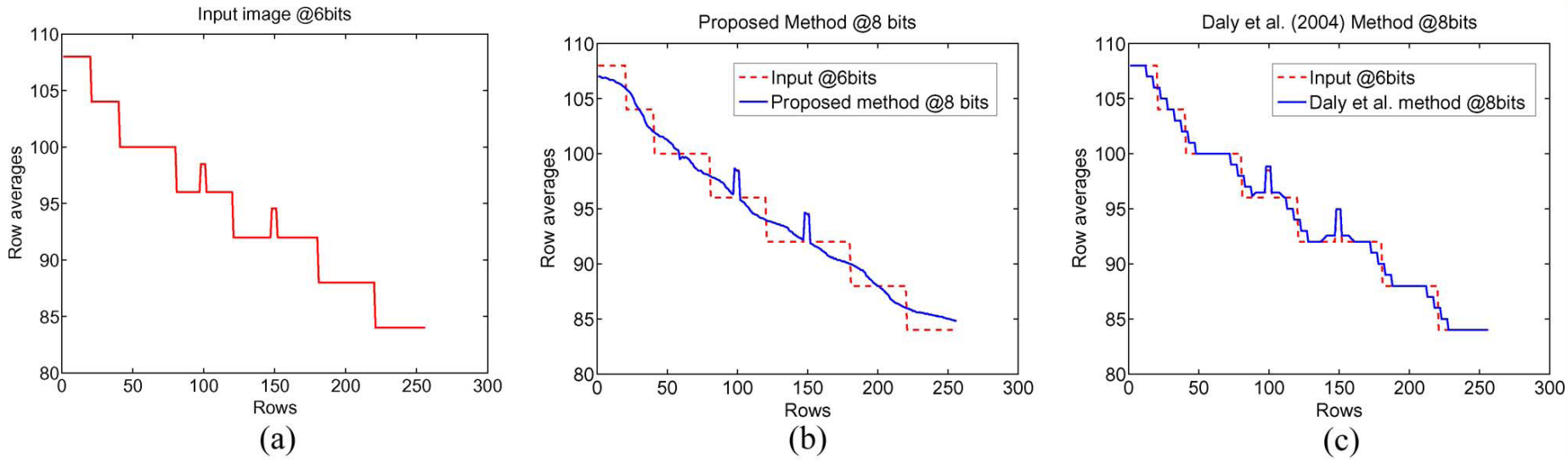


Fig. 6. Row averages of intensities for the image in Fig. 4(a) using (a) the input image, (b) the proposed bit depth extension method, and (c) the method of Daly *et al*. The two intermittent peaks in the profile are due to the presence of the white dots in Fig. 4(a).

– Few real-world example

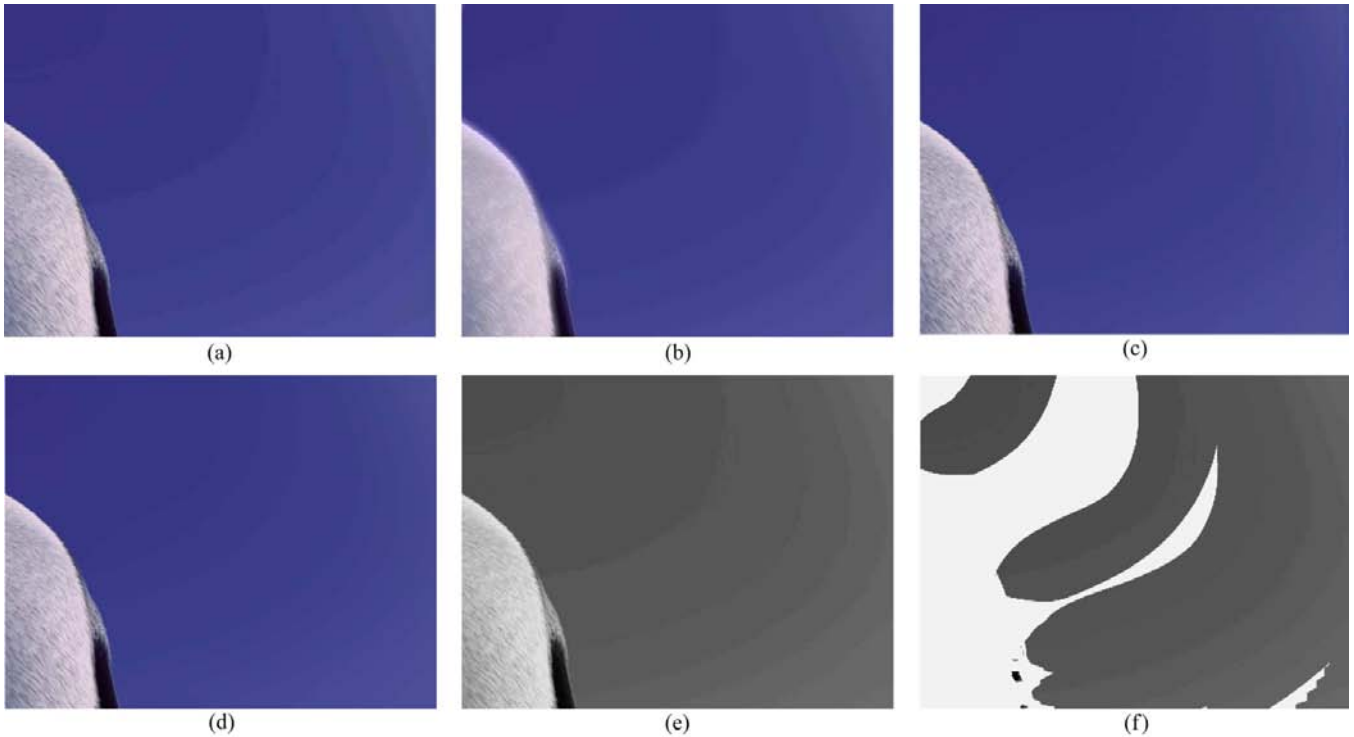


Fig. 5. (a) 6 bits-per-component (bpc) image with contour artifacts, (b) result of the "decontouring" method of Daly *et al. [7] at 8 bpc, (c) result of the method* proposed in B at 6 bpc, (d) result of the proposed bit depth extension method at 8 bpc, (e) the Y-component of the original 6 bpc image, and (f) the regions in the Y-component detected as constituting contour artifacts (same image as (e) but wherein regions *not containing contour artifacts are masked* out by white space).

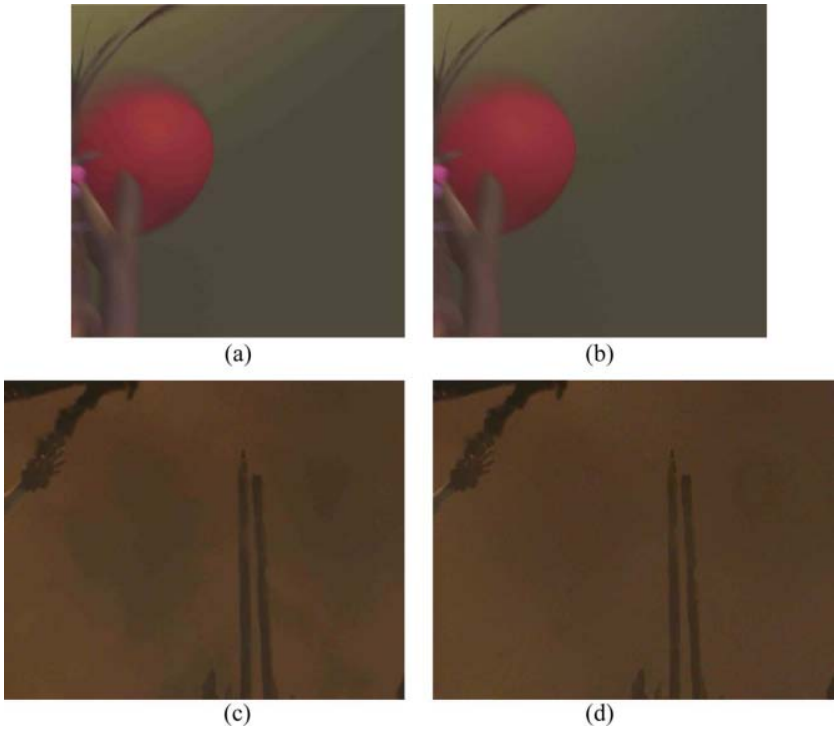
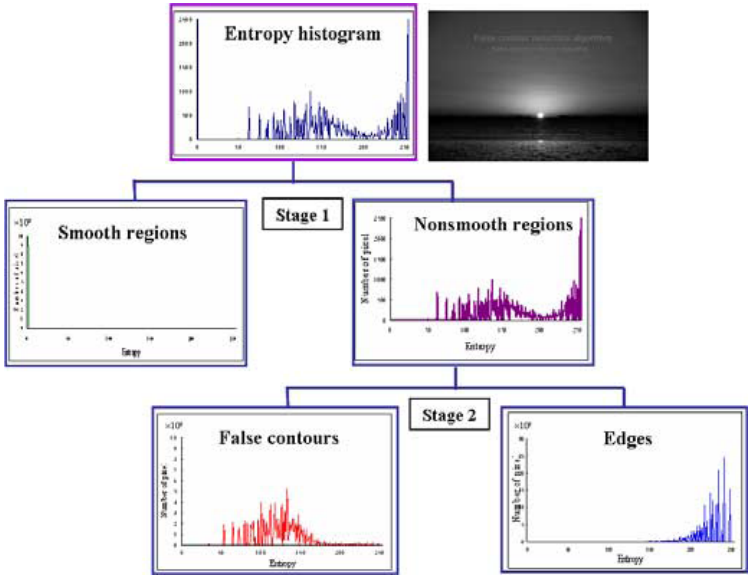


Fig. 7. More examples of contour artifact reduction using the method proposed in B. The left-hand side shows the original 6 bpc images and the right-hand side shows the corresponding output images at the same bit depth. The example in the top row is cropped from a frame in an animation movie and the one in the bottom row is from a nonanimation movie.

Conclusion

- Proposed method
  - Reducing visibility of contour artifacts
    - False contours arising from color quantization in digital images
  - Composition of two steps
    - Multiscale contour artifact detection
    - Probabilistic dithering
  - Advantage of proposed method
    - Effective handling of contour artifacts of various scales
    - Ensure artifact-free regions and small details
    - Provide output image with same bit depth as input image or with higher bit depths

◆ Lee et al.


Entropy histogram
Smooth regions
Stage 1
Nonsmooth regions
Stage 2
False contours
Edges

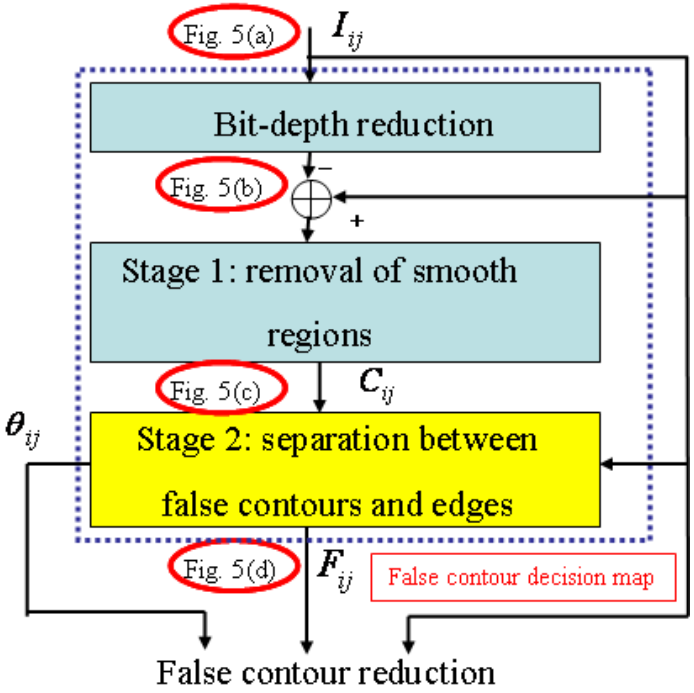
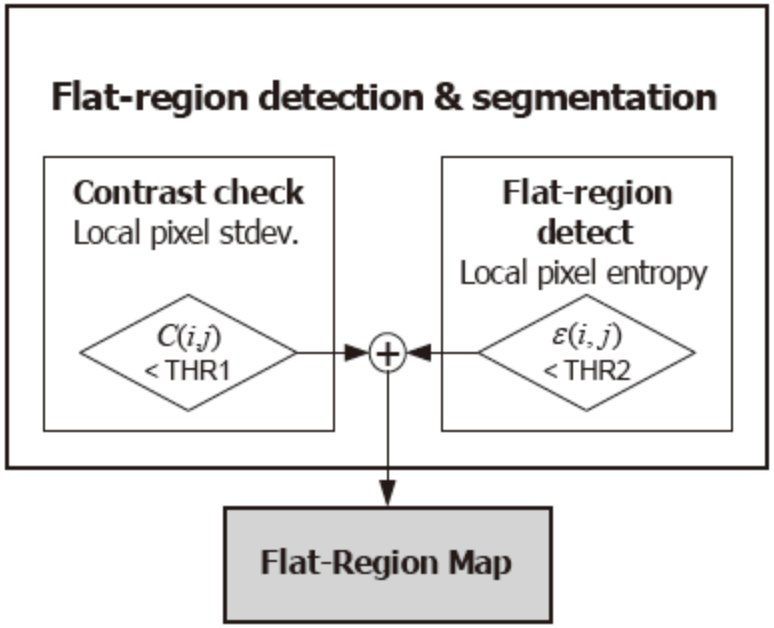
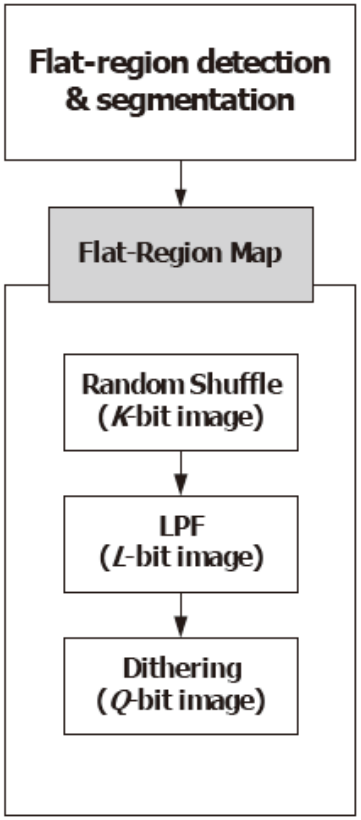


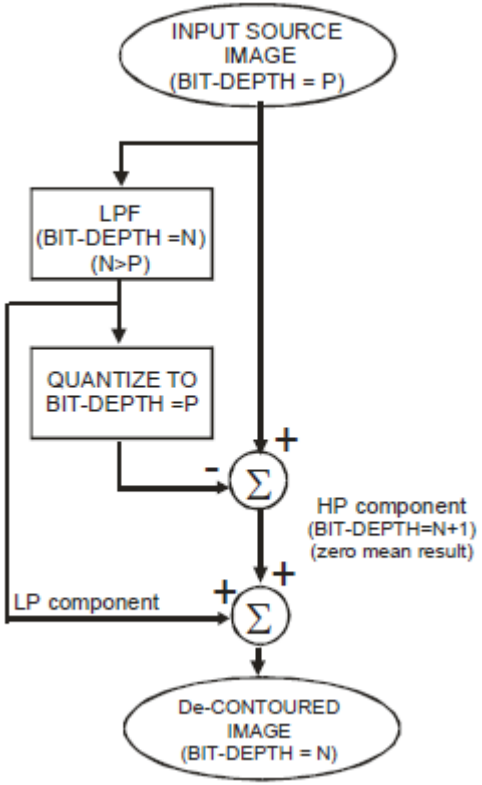
Fig. 5(a)
I_{ij}
Bit-depth reduction
Fig. 5(b)
Stage 1: removal of smooth regions
Fig. 5(c)
C_{ij}
θ_{ij}
Stage 2: separation between false contours and edges
Fig. 5(d)
F_{ij}
False contour decision map
False contour reduction

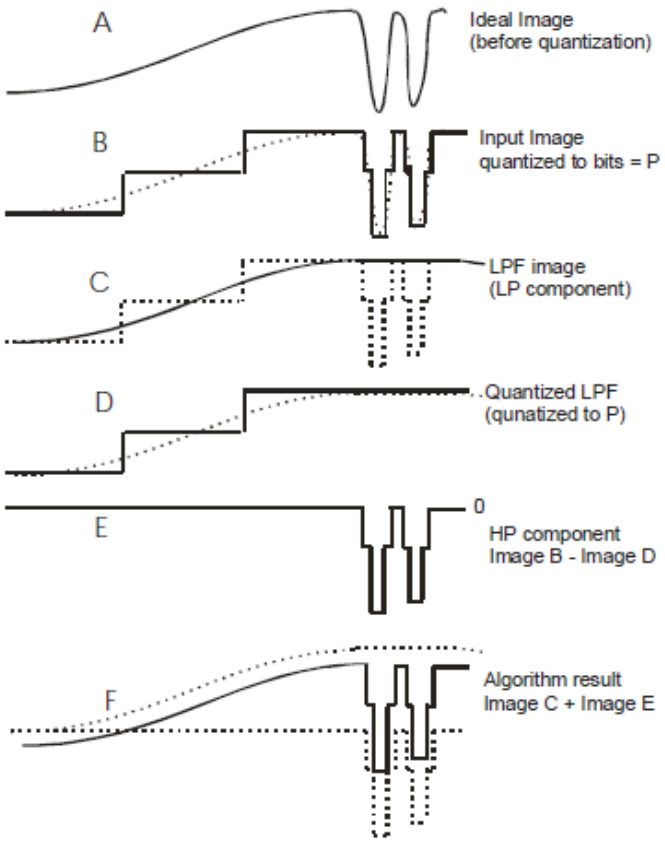
◆ Ahn et al.


Flat-region detection & segmentation
Contrast check
Local pixel stdev.
C(i,j)
< THR1
Flat-region
detect
Local pixel entropy
ε(i, j)
< THR2
Flat-Region Map


Flat-region detection
& segmentation
Flat-Region Map
Random Shuffle
(K-bit image)
LPF
(L-bit image)
Dithering
(Q-bit image)

◆ Daly et al.


INPUT SOURCE IMAGE (BIT-DEPTH = P)
LPF (BIT-DEPTH =N) (N>P)
QUANTIZE TO BIT-DEPTH =P
HP component (BIT-DEPTH=N+1) (zero mean result)
LP component
De-CONTOURED IMAGE (BIT-DEPTH = N)


A
Ideal Image (before quantization)
B
Input Image quantized to bits = P
C
LPF image (LP component)
D
Quantized LPF (qunatized to P)
E
0
HP component Image B - Image D
F
Algorithm result Image C + Image E