Hierarchical Error Diffusion

*IEEE Transactions on Image Processing*

Vol. 18, No. 7, 2009

Zhen He

Presented by Il-Su Park

School of Electrical Engineering and Computer Science
Kyungpook National Univ.
Abstract

◆ Proposed method

– Hierarchical error diffusion (HED)
  • Achieving perceptually pleasing color halftone
  • Controlling three critical factor
    – Dot-overlapping control
      » Achieving minimum brightness variation color density (MBVCD)
    – Dot-positioning control
      » Employing the embedded monochrome error diffusion
    – Dot-coloring control
      » Thresholding the elements in partial density sum vector
Error diffusion

- Dispersed-dot halftoning scheme
  - Rendering continuous tone images on devices with limited native primary colors
  - Achieving continuous-tone illusion through dot frequency modulation
  - Diffusing the pixel quantization error to local neighboring pixels
– Grayscale halftoning algorithm
  • Original Floyd-Steinberg error diffusion
    – Two most noticeable halftone texture artifact
      » Directional “wormy” halftone texture in the highlight and shadow areas
      » Regular halftone pattern at the multiples of $1/4$ and $1/3$
– Error diffusion to color case
  • Separable error diffusion (SED)
    – Processing each color channel independently
    – Not control the intercolor correlation among channel
  • Vector error diffusion (VED)
    – Exploring the interchannel color correlation
    – VED type I
      » Executing joint quantization in a device-independent space
      » Diffusing error channel-independently
    – Semi VED
      » Tradeoff the merits between SED and VED
    – VED type II
      » Scalar quantization decision for each channel
      » Diffusing color error across channel
Error diffusion formulation

◆ Standard error diffusion algorithm
  
  – Denotation
    
    • \((i, j)\)
      
      – Pixel grid, \(1 \leq i \leq M, 1 \leq j \leq N\)
    
    • \(\bar{x}(i, j)\)
      
      – L-dimensional input pixel vector with each element value with the range \([0.0, 1.0]\)
    
    • \(\bar{b}(i, j)\)
      
      – Binary vector halftone

Fig. 1. Flowchart of standard error diffusion algorithm.
– Modified pixel vector
  • Adding the filtered pixel errors from the previously processed pixel to the current input vector
    \[
    \bar{x}^*(i, j) = \bar{x}(i, j) + \sum_{(m,n)\in S} \bar{e}(i-m, j-n)\bar{h}(m,n)
    \]
    where $S$ is the support of the low-pass filter $\bar{h}(\cdot)$.

– Binary halftone
  • Quantizing the modified pixel vector
    \[
    \bar{b}(i, j) = Q(\bar{x}^*(i, j)) = \arg \min_b D(\bar{x}^*(i, j), \bar{b})
    \]
    where $D$ is an error metric for quantization.

– Quantization error
  \[
  \bar{e}(i, j) = \bar{x}^*(i, j) - \bar{b}(i, j)
  \]
Motivation

- **Color halftoning algorithm**
  - Two questions to answer
    - Which pixel location to put a dot?
    - Which dot color to choose?
  - Set of the halftone dot color
    - Composite-black, blue, red, green, magenta, cyan, yellow and white

\[
\begin{align*}
R &= M + Y \\
G &= C + Y \\
B &= M + C \\
K_p &= C + M + Y
\end{align*}
\]  

where "+" means physical dot overlapping, not the density summation.
– Defining dot-position bitmap as a binary image $b_p(i, j)$
  • At least one of the elements in vector $\hat{b}(i, j) = 1$
    – $b_p(i, j) = 1$
  • Otherwise
    – $b_p(i, j) = 0$

– Defining the total dot density $d_t$
  • Ratio between the number of pixels where $b_p(i, j) = 1$ and the number of the total pixels
- Rendering of an equal-density CMY color with density $d=1/16$ each channel
  - Some observation by examining Fig.2.
  - Quite difference in perceptual image quality among different rendering schemes
  - Blue noise characteristics

![Fig. 2. Rendering color $c = m = y = 1/16$ with different color overlapping strategy: (a) $K_p$, (b) B/Y, (c) R/C, (d) G/M, (e) C/M/Y.](image-url)
General framework of the algorithm

- Color overlapping control module
  - Regulating the color overlapping strategy
  - Color preservation at digital bitmap level
  - Summation of output densities
    » No greater than 1.0 for any CMYK

Fig. 3. Architecture of hierarchical error diffusion.
– Dot-positioning module
  • Deciding where to enable dots
  • Employing a MED

– Dot-coloring module
  • Deciding which color to use at the dot-enabled pixel location
  • Preserving local average color
  • Taking the reduction of the halftone texture visibility into account

Fig. 4. Implementation of hierarchical error diffusion.
Color-overlapping control with minimum brightness variation color density (MBVCD)

- Rendering an input color with the color of less brightness contrast

- Procedure
  
  - Clipping the CMY input density to $1 - k$
  - Deciding the minimum $K_p$ required to accommodate the input CMY coverage

  $$k_p = \max\left(\text{sum}_i - 2(1.0 - k), 0\right)$$

  where $\text{sum}_i = c_i + m_i + y_i$.
• Subtracting $k_p$ from each $c_i, m_i, y_i$

• Total density of the secondary colors

\[ T_{rgb} = \max \left( \text{sum}_2 - \left( 1.0 - k - k_p \right), 0 \right) \]  

(6)

where $\text{sum}_2 = c_1 + m_1 + y_1$.

• Deciding the secondary color
  
  – Minimum inevitable density for B

\[ b = T_{rgb} - \min \left( y_1, T_{rgb} \right) \]  

(7)

• Updating the unsettled C, M, and Y densities to $c_2, m_2, \text{ and } y_2$
  
  – Subtracting the amount of B from C and M channels
  
  – Keeping Y channel

• Total secondary color density consists of R and G

\[ T_{rg} = T_{rgb} - b \]  

(8)
• Inevitable R density

\[ r = T_{rg} - \min\left( c_2, T_{rg} \right) \]  

(9)

• Remaining secondary color density

\[ g = T_{rg} - r \]  

(10)

• Remaining C, M, and Y densities

\[ c = c_2 - g \]

\[ m = m_2 - r \]

\[ y = y_2 - r - g \]  

(11)
The Algorithm of Computing MBVCD

Input: pixel color quadruple \( (c_i, m_i, y_i, k_i) \)
Output: \( \vec{v}_d = [k, k_p, b, r, g, m, c, y] \)

1) Clip \( c_i, m_i, y_i \) to \( 1.0 - k \)
2) \( k_p = \max \left( \sum_1 - 2 * (1.0 - k_i), 0 \right) \)
3) \( c_1 = c_i - k_p; m_1 = m_i - k_p; y_1 = y_i - k_p; \)
   \( \sum_2 = c_1 + m_1 + y_1 \)
   \( T_{rgb} = \max \left( \sum_2 - (1.0 - k_i - k_p), 0 \right) \)
4) \( b = T_{rgb} - \min \left( y_1, T_{rgb} \right) \)
5) \( c_2 = c_1 - b; m_2 = m_1 - b; y_2 = y_1 \)
   \( T_{rg} = T_{rgb} - b \)
   \( r = T_{rg} - \min \left( c_2, T_{rg} \right) \)
6) \( g = T_{rg} - r \)
7) \( c = c_2 - g; m = m_2 - r; y = y_2 - r - g; k = k_i \)
- Theorem 1
  - Counting $K$ and $K_p$ as two different colors
    - Maximum of five color with positive density including white
  - $K=0$
    - Maximum of four color with positive density including white

- Proof
  - $K_p > 0$
    
    $\text{sum}_1 = 2.0 + k_p - 2k$
    
    $\text{sum}_2 = 2.0 - 2k_p - 2k$
    
    $T_{rgb} = \max(1.0 - k_p, 0)$
    
    Equal to the space left after accommodating $K$ and $K_p$
    » Forming the secondary colors
• $k_p=0$
  - $T_{rgb}=0$
    » All primary colors
  - $T_{rgb}>0$
    » No white

• Two secondary colors
  - Completely consuming the density of at least one of Y and C during the construction of B and R
Dot-positioning control with sequential thresholding on PDSV

– Achieving the desirable dot-position bitmap with blue-noise characteristic

– Partial density sum vector (PDSV)

  • Encouraging better dot-positioning for more visible color dots

$$
\mathbf{s} = [s_1, s_2, \ldots, s_8]^T
$$

$$
s_1 = \mathbf{x}^* [1];
\quad s_{l+1} = s_l + \mathbf{x}^* [l];
\quad 1 \leq l \leq 7
$$

(12)
– Change of total accumulated density error at any future pixel location \((i, j)\)

\[
\Delta e(i, j) = \sum_{l=1}^{8} \left( \tilde{x}^* [l] - \tilde{b}_1 [l] \right) h(i, j)
\]

\[
= h(i, j) \left( \sum_{l=1}^{8} \tilde{x}^* [l] - \sum_{l=1}^{8} \tilde{b}_1 [l] \right)
\]

\[
= h(i, j) \left( s_8 - b_2 \right)
\]
Dot-coloring control with color visibility and density priorities

- Encouraging better dot-position for color of more contrast
  
  - Constraining the potential color selection to the subset of the color
  
  - Excluding less visible color from competing for dot-position
  
  - Choosing the halftone color to be the one with the greatest density contribution to $s_q$
In the experiment

– Comparing the halftone of HED with five state-of-the-art color error diffusion algorithm
  • SED
  • VED type I algorithm with joint quantization
    – using Euclidean distance in CIE XYZ space
  • VED type II algorithm
  • Semi-VED
  • MBVQ color diffusion
- Halftone result of a gray sweep of equal-RGB

Fig. 5. Comparison of halftone of an equal-RGB sweep (neutral gray) using algorithm (a) SED, (b) VED type I, (c) VED type II, (d) Semi-VED, (e) MBVQ, (f) HED.
– More-detailed examination of the dot-color scheme and dot-positioning behavior

**Fig. 6.** Enlarged halftone quality comparison, from left to right: SED, VED type I, VED type II, Semi-VED, MBVQ and HED, from top to bottom: equal-RGB input color density: (a) 240/255, (b) 160/255, (c) 80/255, (d) 12/255.
– Radial power spectral density (PSD)

Fig. 7. Luminance channel radial power spectral density (PSD) comparison of the six algorithms at equal-RGB input color density: (a) 240/255, (b) 160/255.
Fig. 8. Comparison of halftone of Kodak image “Macaw”: (a) original image, (b) SED, (c) VED type I, (d) VED type II, (e) Semi-VED, (f) MBVQ, (g) HED.
– Color dot percentage statistics of the halftone image of Macaw and the six demo images

Table 1. Comparison of dot coverage statistics between MBVQ and HED; unit: %

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Dot Color</th>
<th>Macaw</th>
<th>Boat</th>
<th>Girl</th>
<th>House</th>
<th>Lena</th>
<th>Peppers</th>
<th>Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBVQ</td>
<td>Kp</td>
<td>8.5362</td>
<td>15.4835</td>
<td>8.7613</td>
<td>2.824</td>
<td>5.9538</td>
<td>11.3044</td>
<td>10.9822</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>7.7476</td>
<td>10.4435</td>
<td>7.3452</td>
<td>8.2986</td>
<td>18.1126</td>
<td>15.462</td>
<td>8.292</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>2.7117</td>
<td>14.3202</td>
<td>5.0102</td>
<td>13.6162</td>
<td>2.3151</td>
<td>5.0582</td>
<td>12.8743</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>7.554</td>
<td>10.546</td>
<td>7.3055</td>
<td>7.9219</td>
<td>18.0233</td>
<td>15.0456</td>
<td>8.1433</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>4.4628</td>
<td>11.3276</td>
<td>2.7299</td>
<td>10.1923</td>
<td>4.0931</td>
<td>1.8441</td>
<td>14.3037</td>
</tr>
</tbody>
</table>
Conclusion

◆ Class of color error diffusion algorithm
  – Hierarchical error diffusion
    • Controlling three critical factor
      – Color overlapping
      – Dot-positioning
      – Dot-coloring at separate stages
    • Taking advantage of the rich MED design knowledge
  – Comparison with five state-of-the-art color error diffusion algorithms
    • Demonstrating excellent halftone image quality