THE MOMENT-PRESERVING PYRAMID FOR PROGRESSIVE IMAGE TRANSMISSION

Jeng-Jong Pan
Research and Data Systems Corporation
Greenbelt, Maryland
U.S.A.

Sheng-Tun Li
Dept. of Computer Science
University of Houston
Houston, Texas
U.S.A.

ABSTRACT

The pyramid structure used in image encoding provides an efficient approach for progressive image transmission. The pyramid consists of a sequence of reduced-resolution images generated from an image. The original image is processed and decimated to a coarser pyramid, and the lower level represents a higher resolution. The top level image can be transmitted rapidly due to its small size, it is then expanded and displayed at the receiving station. The difference images are also transmitted sequentially to improve the image quality. The viewer can terminate the transmission quickly if the image is not desired. Most approximations used to compute the reduced-resolution images are based on different filters. In this paper, a moment-preserving pyramid structure is proposed to generate the pyramid which can reduce the entropy and variance of the difference image at each level. Compared with Gaussian-Laplacian pyramid, it can achieve a higher compression ratio.

KEY WORDS: Pyramid, Moment, Encoding, Transmission.

INTRODUCTION

The multiresolution pyramid for progressive image transmission has been widely discussed (e.g., Anderson et al., 1984, Chin, et al., 1989, Riou and Vetterli, 1991, Goldberg and Wang, 1993). The pyramid structure used in the image encoding provides an efficient approach for progressive image transmission. The pyramid consists of a sequence of reduced-resolution images which resulted from an image. The original image is processed (e.g., filtering) and decimated to one-quarter of its original size, the reduced image is expanded and subtracted from the original image, the difference is then coded. Repeat the same procedure, we can generate a very small image at the top level of the pyramid and save a set of difference images which resulted from the difference between two adjacent levels. The top level image can be transmitted rapidly due to its small size, it is then expanded and displayed at the receiving station. The difference images can be expanded and displayed at the receiving station. The difference images are also transmitted to improve the image quality. The viewer can terminate the transmission quickly if the image is not desired.

There are two concerns in the progressive image transmission using the pyramid structure. The first one is that we hope that the expanded image at the receiving station looks like its original image, so the viewer can make a decision earlier to cease the image transmission or not. One way to evaluate the performance objectively is based on the variance of the difference image, which provides a quantitative measurement of the approximation. The second concern is how to minimize the entropy used to compute the reduced-resolution images are based on different filters. In this paper, a moment-preserving pyramid structure is proposed to generate the pyramid which can reduce the entropy and variance of the difference image at each level. Compared with Gaussian-Laplacian pyramid, it can achieve a higher compression ratio.

Most approximations used in the computation of reduced-resolution images are based on different low-pass or band-pass filters (e.g., Anderson et al., 1984). In this paper, we first review the Gaussian pyramid and the corresponding Laplacian pyramid. Then a new moment-preserving approach is discussed. Several test results indicate that the new approach, compared with the Gaussian-Laplacian pyramid, can improve the two requirements mentioned above.

GAUSSIAN-LAPLACIAN PYRAMID

Let us assume that an image at the level \( k \) in the pyramid structure is \( G_k(i,j) \), the reduced-resolution at the next level \( k+1 \) is \( G_{k+1}(i,j) \), and usually the original image is referred as \( k=0 \). Burt and Adelson (1983) proposed a 5 x 5 Gaussian-like weighting function, which is equivalent to a low-pass filter, to remove the pixel-to-pixel correlations when compute the \( G_{k+1}(i,j) \) from \( G_k(i,j) \).

The level-to-level reduction is performed as

\[
G_k(i,j) = \sum_{m=0}^{2} \sum_{n=0}^{2} w(m,n) G_{k-1}(m+2i,n+2j)
\]

where \( w(m,n) \) is a two-dimensional Gaussian-like weighting function which can be generated by \( w(m,n)=f(m)f(n) \), \( f(m) \) is a normalized and symmetric function defined as

\[
f(m) = \sum_{m=0}^{3} f(m) = F(-m) = 1, \quad 0, 1, 2.
\]

One of such functions which satisfies the above conditions is (Burt, 1983)

\[
\begin{align*}
f(0) & = c, \\
f(-1) & = f(1) = \frac{1}{4}, \\
f(-2) & = f(2) = \frac{-1}{4}.
\end{align*}
\]

where \( c \) is a constant and can be adjusted to minimize the entropy and variance of the Laplacian images which will be discussed later.

If the dimension of \( G_k(i,j) \) is \( 2^k \times 2^k \), then the dimension of \( G_{k+1}(i,j) \) is reduced to \( 2^{k+1} \times 2^{k+1} \), i.e., the resolution at level \( k \) is always reduced from the level \( k-1 \) by 2 in each dimension. The sequential images computed from this approach is organized as a pyramid, which is called Gaussian Pyramid, and the lower level represents a higher resolution. The \( G_{k+1}(i,j) \) can be expanded to the same dimension as \( G_k(i,j) \) through an appropriate interpolation. Let us assume that the expanded image from \( G_k(i,j) \) is \( G_k(i,j) \).
where only the pixels which make \([x]\) to be an integer are computed in the sum.

The difference image, which is called the Laplacian image if the Gaussian-like weighting function is used in eqs. 1 and 4, \(L_k(i,j)\) at the level \(k\) is defined as

\[
L_k(i,j) = G_k(i,j) - G_k^{(1)}(i,j)
\]

where \(G_k(i,j)\) is also recognized as the prediction error, which usually has small dynamic range, and the entropy of \(L_k(i,j)\) could be less than that of \(G_k(i,j)\). Therefore, the encoding of the image \(G_k(i,j)\) is replaced by encoding \(L_k(i,j)\), \(k = 0, 1, 2, \ldots, L-1\) and \(k = 0, 1, 2, \ldots, L/2-1\). The set of Laplacian images are organized as the Laplacian pyramid. One potential problem in the Laplacian pyramid generation is how to select the optimal coefficient \(c\) in eq. 3, so that the entropy and variance of each Laplacian image have the minimal values. Burt and Adelson (1983) found that \(c=0.6\) was an optimal value in their test. The same value of \(c\) is also used in the following images test.

In the next section, a moment-preserving processing is used to replace the Gaussian-like weighting function in the Gaussian-Laplacian pyramid generation. Rather than smoothing the pixels through a \(5 \times 5\) window, the first and second moments in a \(2 \times 2\) window at level \(k\) are still preserved maximally in the corresponding pixel at level \(k+1\).

\textbf{MOMENT-PRESERVING PYRAMID}

Moment-preserving method has been implemented recently in various image processing, such as image segmentation (Tsai, 1985), optical character recognition (Cash and Hatamian, 1987), image enhancement (Chen and Tsai, 1988), and quantization (Delp and Mitchell, 1991).

The difference image at each level is obtained through the same approach as defined in eq. 5. In order to evaluate the image transmission efficiency and quality, the impact on the entropy and variance of difference images using these two different interpolation methods is discussed in the next section. However, the interpolation method used at the receiving station could be irreducible to that used in the compression procedure.

\textbf{ENTROPY AND VARIANCE COMPARISONS}

In order to compare the performances of progressive image transmission using the Gaussian-Laplacian pyramid and the moment-preserving pyramid, three images appeared in most image analysis, 'Lena', 'Cameraman', and 'City', are used here for the experimental evaluation of entropy and variance at each level.

Figures 1 and 2 show these three original images and the corresponding histograms, respectively. The size of each original image is \(256 \times 256\) pixels, but it is expanded to \(257 \times 257\) by repeating the rightmost and the lowest row in the Gaussian-Laplacian pyramid computation. The entropy of image is 7.58 (Lena), 7.01 (Cameraman), and 7.34 (City). The histograms indicate that the dynamic range of pixels is wide enough to segment several objects.

Figures 3 and 4 show the Gaussian-Laplacian pyramid and the moment-preserving pyramid of image 'Lena', respectively. Visually, the reduced-resolution images generated from both methods still preserve the most significant features, which include the tone and shapes of major objects. The entropy and variance of each image at each level using these two methods are compared (Tables 1 to 3). The two different expansion methods, nearest neighbor and Gaussian-like expansions, are also included in the comparisons.

The new pixel \(q\) is determined by minimizing an objective function, which is defined as

\[
\text{minimize } a(m_{1,k} - m_{1,k+1})^2 + b(m_{2,k} - m_{2,k+1})^2
\]

where \(a\) and \(b\) are two weighting coefficients which can be used to emphasize the first or the second moment in the pyramid generation. Here we assume that both \(a\) and \(b\) are 1. In this case, the second moment could have more impact on the minimization. If \(b=0\), then the moment-preserving method will be equivalent to the mean pyramid (Goldberg and Wang, 1991).

The solution for eq. 9 can be simplified through the least-squares approximation. It is equivalent to find the root of the following equation,

\[
2q^3 + (1-2m_{1,k})q - m_{1,k} = 0
\]
Tables 4 to 6 list the variance of each difference image at the first five levels, respectively. Again, the variance of the difference image computed for the moment-preserving method with the nearest neighbor expansion has the minimal value, and which is around 30% to 60% less than that from the Gaussian-Laplacian method. The lower variance could reflect both the higher image quality and the lower tone variation.

The comparisons indicate that the moment-preserving pyramid with nearest neighbor expansion can speed the progressive image transmission by reducing both the entropy and variance of difference images. The transmitted images from level by level can be used to reconstruct its original image without any error if no quantization is implemented on the difference image and no other transmission noise is involved. Entropy can be further reduced if the pixels of the difference image at each level are quantized, however, at the expense of sacrifice image quality.

CONCLUSIONS

A moment-preserving pyramid is proposed to speed the progressive image transmission. Its performance, based on the entropy and variance criteria, is better than that of the Gaussian-Laplacian pyramid. Because the entropy and variance are reduced around 20% and 45%, respectively, the moment-preserving pyramid requires less bits in encoding of the difference images, but still preserves higher image quality at each level. The moment-preserving pyramid works on both lossless encoding and lossy encoding if quantization is involved.
Figure 2. The corresponding histograms of Fig. 1. (a) Lena, (b) Cameraman, and (c) City.

Figure 3. The first five levels of (a) the Gaussian pyramid and (b) the Laplacian pyramid of the image 'Lena'.

Figure 4. The first five levels of (a) the moment-preserving pyramid and (b) the corresponding difference pyramid of the image 'Lena'.

477
Table 1. Entropy comparison of the image 'Lena''

<table>
<thead>
<tr>
<th></th>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplacian</td>
<td>5.62</td>
<td>6.14</td>
<td>6.70</td>
<td>7.03</td>
<td>6.87</td>
</tr>
<tr>
<td>Moment 1</td>
<td>5.05</td>
<td>5.54</td>
<td>6.14</td>
<td>6.57</td>
<td>6.57</td>
</tr>
<tr>
<td>Moment 2</td>
<td>4.71</td>
<td>5.21</td>
<td>5.76</td>
<td>6.06</td>
<td>6.77</td>
</tr>
</tbody>
</table>

* Moment 1 is based on Gaussian-like expansion with $c=0.6$.
Moment 2 is based on nearest neighbor expansion.
The size of Laplacian pyramid is the number plus 1, i.e.
(257x257, 129x129, ...)

Table 2. Entropy comparison of the image 'Cameraman'

<table>
<thead>
<tr>
<th></th>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplacian</td>
<td>5.27</td>
<td>5.42</td>
<td>5.70</td>
<td>5.89</td>
<td>6.27</td>
</tr>
<tr>
<td>Moment 1</td>
<td>4.90</td>
<td>5.02</td>
<td>5.29</td>
<td>5.66</td>
<td>6.49</td>
</tr>
<tr>
<td>Moment 2</td>
<td>4.58</td>
<td>4.65</td>
<td>4.83</td>
<td>5.08</td>
<td>6.49</td>
</tr>
</tbody>
</table>

Table 3. Entropy comparison of the image 'City'

<table>
<thead>
<tr>
<th></th>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplacian</td>
<td>6.01</td>
<td>6.03</td>
<td>6.32</td>
<td>6.62</td>
<td>6.74</td>
</tr>
<tr>
<td>Moment 1</td>
<td>4.90</td>
<td>5.02</td>
<td>5.29</td>
<td>5.66</td>
<td>6.49</td>
</tr>
<tr>
<td>Moment 2</td>
<td>4.58</td>
<td>4.65</td>
<td>4.83</td>
<td>5.08</td>
<td>6.49</td>
</tr>
</tbody>
</table>

Table 4. Variance comparison of the image 'Lena'

<table>
<thead>
<tr>
<th></th>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplacian</td>
<td>342</td>
<td>608</td>
<td>1008</td>
<td>1382</td>
<td></td>
</tr>
<tr>
<td>Moment 1</td>
<td>144</td>
<td>255</td>
<td>443</td>
<td>672</td>
<td></td>
</tr>
<tr>
<td>Moment 2</td>
<td>101</td>
<td>186</td>
<td>302</td>
<td>380</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Variance comparison of the image 'Cameraman'

<table>
<thead>
<tr>
<th></th>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplacian</td>
<td>494</td>
<td>636</td>
<td>804</td>
<td>1047</td>
<td></td>
</tr>
<tr>
<td>Moment 1</td>
<td>252</td>
<td>302</td>
<td>370</td>
<td>459</td>
<td></td>
</tr>
<tr>
<td>Moment 2</td>
<td>190</td>
<td>247</td>
<td>291</td>
<td>343</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Variance comparison of the image 'City'

<table>
<thead>
<tr>
<th></th>
<th>256x256</th>
<th>128x128</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplacian</td>
<td>675</td>
<td>600</td>
<td>754</td>
<td>878</td>
<td></td>
</tr>
<tr>
<td>Moment 1</td>
<td>430</td>
<td>349</td>
<td>392</td>
<td>364</td>
<td></td>
</tr>
<tr>
<td>Moment 2</td>
<td>358</td>
<td>239</td>
<td>278</td>
<td>274</td>
<td></td>
</tr>
</tbody>
</table>