Iowa State University

From the SelectedWorks of Sarah A. Rajala

November 1, 1992

Segmentation-based texture coding algorithm for packet video: a goal-oriented approach

Day-Fann Shen, North Carolina State University
Sarah A. Rajala, North Carolina State University

Available at: https://works.bepress.com/sarah_rajala/28/
Segmentation-based texture coding algorithm for packet video: 
a goal oriented approach

Day-Fann Shen and Sarah A. Rajala
Department of Electrical and Computer Engineering
North Carolina State University
Campus Box 7911
Raleigh, NC 27695-7911

ABSTRACT

The design of an image codec for packet-switched transmission is formulated as a minimization problem. A general set of design requirements is derived and used to design a segmentation-based texture coding algorithm. The segmentation process is performed on a pyramid data structure and uses the just noticeable difference (JND) of the human visual system as the merge criterion. To reduce the bit-rate while maintaining image quality, each region is classified as either texture or non-texture. Texture regions are approximated by a one-dimensional polynomial, while the non-texture regions are approximated by the region's mean intensity. A set of parameters for bit-rate/image quality tuning is identified and their effect evaluated on LENA and HOUSE.

1. INTRODUCTION

If it is important to reduce the impact of a packet-switched network on the performance of an image coding system, then the system design criteria must be respecified. We accomplish this by formulating the design process as a minimization problem. Mathematically, it is given by:

Min \( \text{COST}(R_1, ..., R_N, C_L, H) = \sum_{n=1}^{N} a_n R_n + b \cdot D(R_1, ..., R_N, C_L) + c \cdot H \)

Subject to

1. \( \sum_{n=1}^{N} R_n \leq R_{max} \)
2. The distortion \( D(R_1, R_2, R_3, ..., R_N, C_L) \leq \) Specified distortion
3. All processing must be done in real-time.

\( R_n \) is the bit-rate for encoding the \( n_{th} \) image class; \( C_L \) is the level of spatial complexity and temporal activity on the images sequence; and \( H \) is the initial cost which primarily includes the hardware cost. \( N \) is the total number of classes of image data. \( a_n \) is the transmission cost associated with the \( n_{th} \) image class in terms of dollars. \( b \) is a weighting factor which converts the image degradation or image quality into a cost in terms of dollars; and \( c \) is a weighting factor which converts the initial installation cost to the same basis as the transmission cost. \( R_{max} \) is the maximum allowable bit-rate for the image codec; this value will vary depending on the network traffic conditions. \( d \) is a measure of the image distortion and is a function of the bit-rate for each class of image data. The control variables are the bit-rate for each class and the algorithm complexity.

Ideally, one would like to explicitly solve this minimization problem to determine the optimal image codec. The minimization problem can be solved if: (1) A physically meaningful and mathematically tractable quantitative measure of image quality (or distortion) can be specified. The commonly used mean square error (MSE) measure does not reflect the actual perceived image quality. (2) The weighting factor \( b \), which converts the image degradation into a cost, can be specified. Unfortunately, this is difficult.

Even without directly solving the minimization problem, the above formulation relates all the important factors of designing an image codec. Several examples are given below: First, if the cost of transmission is reduced (e.g., using...
fiber optics), an evaluation of the generalized cost suggests that an optimal image codec should take advantage of the lower transmission cost \( (a_n) \) by increasing the bit-rate \( (R_n) \) to obtain better image quality which in turn reduces the cost associated with image degradation. Alternatively, one could choose a codec that has higher bit-rate but cheaper initial installation cost. Second, hardware technology has been improving in performance, cost and reliability at exponential rate [2]. Thus, the computing power is increased and hardware costs are reduced. As a result, the optimization principle suggests using a more sophisticated coding algorithm to improve image quality and/or lower bit-rate. Third, the performance of different existing image codecs can be compared based on their generalized cost. Fourth, the optimization principle suggests that as the technologies in computing and transmission advance an efficient packet video codec should be adjusted accordingly. Finally, The efficiency of an image codec can be defined as

\[
E_{\text{codec}} = \frac{\sum_{n=1}^{N} a_n R_n + \frac{1}{\delta(R_1, \ldots, R_N, C_L)}}{c \cdot H}
\]  

1.136 / SPIE Vol. 1818 Visual Communications and Image Processing '92

2. REQUIREMENTS FOR THE DESIGN OF A SEGMENTATION ALGORITHM FOR IMAGE CODING

In order to come up with a segmentation algorithm that is suitable for image coding, we derive a set of requirements for the segmentation algorithm consistent with those for the image codec, see [24].

(1) **REQUIREMENT 1:** The segmented image should be perceptually close to the original image. This requirement is essential if a human being is the final receiver.

(2) **REQUIREMENT 2:** The number of regions in a segmented image should be proportional to the complexity of an image.

(3) **REQUIREMENT 3:** The segmented image should be adjustable from fine to coarse in a smooth manner.

(4) **REQUIREMENT 4:** The performance should be robust on most images. It is desirable to use one codec to handle all of the images in a given application.

(5) **REQUIREMENT 5:** The computational structure should be suitable for parallel VLSI implementation.

3. THE SEGMENTATION ALGORITHM

There are two key elements to any segmentation algorithm: the pixel or region merge criterion and the clustering procedure [26]. In this work, the just noticeable difference (JND) property of the human visual system is chosen for the merge criterion and a pyramid structure is used to perform the clustering. Both are considered in more detail below.

The JND is an important characteristic of the human visual system. It indicates the level at which two neighboring pixels or regions with different intensities can be distinguished [22]. The JND is a function of intensity, viewing distance, and lighting conditions. In any case, the human visual system is most sensitive to intensities in the middle range of gray values. This has important implications in image coding. Using the JND as a merge criterion helps us meet the design requirements in the following ways. (1) The JND merge criterion, unlike other statistical based criterion, performs the merges from an HVS perspective. Therefore, the resulting segmented image more closely approximates the original image from an HVS point of view, thus fulfilling Requirement 1. (2) For a fixed JND threshold, the resulting number of regions is proportional to the complexity of an image, thus, satisfying Requirement 2. (3) By scaling the JND threshold, the number of regions in a segmented image and therefore the perceived image quality can be controlled. The more regions used in representing an image, the better the approximation of the original image will be. Thus, the JND criterion helps to fulfill requirement 3. (4) Finally, unlike a statistical based merge criteria, the JND value can be easily obtained from a lookup table.
A pyramid structure is chosen to implement the clustering procedure. Research has found that human beings do not perform segmentation pixel-by-pixel, but rather using some form of local average [27]. The pyramid provides a structure for simulating this local averaging process. It allows global information in the upper layers to be used as a guide for clustering in the lower layers. Wilson [27] proved that the pyramid data structure is the most effective way of minimizing the segmentation uncertainty. Thus, using a pyramid structure for segmentation is more likely to yield more accurate and robust results and helps to meet Requirements 1, 4 and 5.

It is known that the boundaries obtained by local edge detection and line formation techniques often do not form closed regions and that region growing without proper selection of the starting points results in improper segmentation results. For these reasons, Riseman et al. [21] and M.D. Levine [25] suggest a combination of these two techniques for the best results. Our segmentation algorithm is based on the approach proposed by Levine [25]. We replace Levine's nearest neighbor list merge criterion with the JND criterion, and use it for both region growing and small region removal. By using the JND criterion, we are able to eliminate the complicated texture pyramid used by Levine and still obtain good segmentations for a variety of images, including LENA and HOUSE. The flowchart of the segmentation algorithm is illustrated in Figure 1.

**STEP 1: Edge map generation** The purpose of the edge map is to assist in the selection of proper starting points for clustering (region growing).

**STEP 2: Building of the gray level pyramid and edge map pyramid** The original image forms the bottom layer of the gray level pyramid. The gray level of a parent node at level \( k \) is the average gray level of its four children at level \( k+1 \). Similarly, the edge map created in Step 1 is the bottom level in the edge pyramid. The parent node is an edge pixel if any of its four children is an edge pixel. Thus, higher layers in the pyramid contain more global information in the image.

**STEP 3: Searching for the starting layer and pixels to be classified** The selection of the starting points for the segmentation process affects the accuracy and results of the segmentation. In this algorithm, a mechanism for global initialization using the pyramid data structure is employed to select the largest, most uniform regions to start the region growing. The starting layer is the highest layer in the pyramid which contains at least one non-edge pixel. When the starting layer is found, the algorithm selects non-edge pixel(s) as the seed(s) for the region growing. Non-edge pixels are good candidate for region seeds, because they are relatively far away from the region boundary.

**STEP 4: Region growing based on the JND criterion** In the beginning each seed forms a region by itself. Regions with pixels in the eight-connected neighborhood of the seed are examined for similarity in terms of the JND criterion. The seed and the eight-connected neighboring region are tested for possible mergers. They are merged into a single region if their difference in mean intensity is smaller than the JND threshold. The JND threshold, \( T_{JND} \), is given by \( T_{JND} = \alpha J(I_s) \). \( \alpha \) is a scaling factor, \( J(I_s) \) is the JND value for intensity \( I_s \) and \( I_s \) is the intensity of the seed. A seed with intensity \( I_s \) is merged into a neighboring region with mean intensity \( I_r \) if \( |I_s - I_r| < T_{JND} \) and \( I_r \) absolute difference between \( I_s \) and \( I_r \) is smallest. To reduce the number of regions, the remaining connected neighboring regions are examined for possible mergings using the same JND criterion.

When two regions are merged, the region's mean and number of pixels must be updated.

\[
M_k = \frac{N_j \cdot M_j + N_k \cdot M_k}{N_j + N_k}
\]

\[
N_k = N_j + N_k
\]

\[
M_{\text{new}} = \frac{N \cdot M_{\text{old}} + I_2}{N + 1}
\]

\[
N_{\text{new}} = N_{\text{old}} + 1
\]

where \( M_{\text{old}} \) and \( M_{\text{new}} \) are intensities before and after the merge and \( N \) is the number of pixels in the region before merge. If regions \( j \) and \( k \) are merged, the region information is updated. \( M_x \) and \( N_x \) are the mean intensity and the number of pixels in region \( x \), respectively. By adjusting \( \alpha \), the \( T_{JND} \) is adjusted, which in turn controls whether the

SPIE Vol. 1818 Visual Communications and Image Processing '92 / 1137
Figure 1: The Segmentation Process
pixels and/or regions are to be merged. At the end of this step, each seed has been assigned a region number.

**STEP 5: Projection and Selection** The projection process projects the status of all pixels in the current layer into the next lower layer and the selection process selects the appropriate pixels at the new lower layer as seeds for region growing.

**Projection:**
If a parent at layer k has been assigned a region number in the region growing step, then its four children at layer k+1 are also assigned the same region number. Otherwise, no region numbers (Null) are assigned to its four children.

**Selection:**
After projection, the seeds are selected at the k+1 layer. Pixels in the k+1 layer are selected as seeds if they meet either of the following two criterion.

- Two parent pixels are neighbors in different regions, then their sons on the border are selected as seeds. See Figure 2 (a). This allows the pixels on the boundaries to be re-classified, thus reducing the uncertainty on the boundary and resulting in better segmentation.

- Pixels with no region number assigned (Null) and their status in the edge pyramid is non-edge (NE), see Figure 2 (b).

Region growing, projection and selection form a loop which is continued until the bottom (highest resolution) layer is reached, see Figure 1.

---

*SPIE Vol. 1818 Visual Communications and Image Processing '92 / 1139*
STEP 6: Classification of the edge pixels in the bottom layer: After projection to the bottom layer, those pixels without a region number must be edge pixels. Each edge pixel in the bottom layer is then merged into a neighboring region with closest mean intensity. Creation of new regions is not allowed.

STEP 7: Cleaning up small regions After all edge pixels in the bottom layer are merged, many small regions usually exist. Since the number of regions is proportional to the bit-rate, it is desirable to reduce the number of regions while maintaining the image quality. Thus, small regions are merged to a neighbor with closest mean intensity [19], [25]. The segmentation process is now complete.

4. TEXTURE CODING

4.1 An HVS-Based Region Complexity Indicator

A region complexity indicator (RCI) is proposed as a means for identifying texture regions. The RCI is an indicator of the spatial complexity of a region from an HVS perspective. The most widely used RCI in image coding is the variance of a region. Gilge, et al. [5] use the variance to indicate the complexity of the motion compensated residual images. Puri, et al. [18] use the variance of the residual image to determine whether an area is fully or partially motion compensated. In these two cases, variance is used as the indicator of the amount of information in an area. Other researchers have used RCIs in texture analysis for image classification and segmentation [8], [15]. More recently, the fractal dimension [17] has been used as a measure of texture complexity. [11], [16]. The fractal dimension based approaches have proven to be effective, however, the computations are normally very complicated. To come up with an RCI which is not only effective but also simple in computation, we propose a variance based RCI incorporating the JND property of the human visual system. The proposed RCI is referred to as \( \sigma_{JND} \) and is defined as follows:

\[
\sigma_{JND}(k) = \frac{\sigma(k)}{J(k)}
\]

where \( \sigma(k) \) is the sum of the absolute differences between the intensity of each pixel in region \( k \) and the mean intensity of \( k \), \( J(k) \) is the JND value associated with \( M(k) \). Compared to the widely used average absolute difference, \( \sigma \), the computation of \( \sigma_{JND} \) only requires one extra division by \( J(k) \).

Images in Figure 3 show that the proposed RCI \( \sigma_{JND} \) incorporating a HVS property is more effective than the commonly used \( \sigma \). Note that using the proposed \( \sigma_{JND} \) (images on right side of 3), the HVS sensitive areas (the face and shoulder) are classified as textural regions and the HVS insensitive areas (hair) are classified as non-textural regions, thus resulting in better image quality. Using the commonly used \( \sigma \) (images on the left side of Figure 3), a large portion of face and shoulder areas are classified as non-textural areas and the hair areas are classified as texture regions, resulting in visible blocky effects in the face and shoulder areas. The proposed RCI \( \sigma_{JND} \) can also be used to determine the number of coefficients to be sent in the 2-D polynomial approximations [6], so that the bit-rate can be allocated efficiently.

4.2 One-Dimensional Approximation of the Texture Regions

Each row in a texture region is approximated using 1-D polynomials. To understand the proposed 1-D approximation, the following terms are defined:

- **run**: is a line segment containing consecutive pixels of the same row in the same region. A run is approximated by a 1-D polynomial of the first order; that is,

\[
y = \lambda \cdot x + I_{im}
\]

where \( y \) is the approximated gray level for the \( x \)-th pixel in the run.

- \( \lambda \) is the coefficient of the 1-D polynomial.
Figure 3: LENA image: The dark areas indicates texture regions (70 percents). (Upper Left) Classification using $\sigma$; (Upper Right) Classification using $\sigma_{IND}$; (Lower Left) Reconstructed LENA based on $\sigma$ classification; (Lower Right) Reconstructed LENA based on $\sigma_{IND}$ classification.
\* \( I_{\text{lm}} \) is the gray level of the first (left most) pixel in a run.

\* \( L_{\text{run}} \) is the number of pixels in the run.

Thus a pixel's intensity at \((i,j)\) can be approximated by

\[
y(i, j) = y(i, k) + \lambda(j - k)
\]

where \((i,k)\) is the left most pixel of the run and \( j \in [k+1, k+ L_{\text{run}}] \). The \( L_{\text{run}} \) is determined by the accumulated absolute errors \( E(i,j) \)

\[
E(i, j) = \sum_{p=k}^{j} | I(i, j) - y(i, j) |
\]

If \( E(i,j) \) exceeds a predetermined error threshold \( T_{\text{run}} \), then the run is terminated at \((i,j)\). Thus, the larger the \( T_{\text{run}} \) threshold is, the longer the runs become. However, a run is always terminated at the region boundary. Using the proposed approach with the error threshold, \( T_{\text{RUN}} \), results in a shorter run length (fine sampling) in rough texture areas and a longer run length (coarse sampling) in smooth areas. This result is consistent with Huang's observation [9] that "In a slowly changing scene it is important to have fine quantization, but the sampling can be coarse; while in a scene with a larger amount of detail, it is necessary to sample finely, but quantization can be coarse."

5. EXPERIMENTAL RESULTS

Three parameters in the proposed texture coding technique are examined, they are \( \alpha \), \( T_{\text{RCI}} \) and \( T_{\text{run}} \). Recall that \( \alpha \) (JND coefficient), is a scaling factor which controls the merging of pixels and regions. A larger \( \alpha \) value causes pixels and regions to be merged thus producing fewer regions and a smaller \( \alpha \) value produces more regions. Region complexity indicator (RCI) indicates the spatial complexity of a region. A region with an RCI value greater than the predetermined \( T_{\text{RCI}} \) (RCI threshold) is classified as texture region, otherwise it is classified as constant region. The \( T_{\text{run}} \) is the maximum allowed accumulated error in the 1-D approximation as discussed earlier. We are interested in the effects of adjusting these parameters on the bit-rate, image quality and the characteristics of the proposed texture coding technique. The LENA and HOUSE images are chosen in the experiments.

5.1 Effects of adjusting the \( \alpha \)

Using a higher value of \( \alpha \) results in lower bit-rate; The bit-rate is decreases and errors increase consistently as \( \alpha \) increases. See Table 1 and Table 2. In general, the images with lower \( \alpha \) look sharper and have better image quality, but at higher bit-rates.

5.2 Effects of Adjusting the \( T_{\text{RCI}} \)

\( T_{\text{RCI}} \) determines the percentage of texture pixels. Higher \( T_{\text{RCI}} \) values leads to fewer texture pixels. The bit-rate decreases consistently as the texture percentage decreases (\( T_{\text{RCI}} \) increases.) When more pixels are classified as texture, their contents are coded with the more expensive 1-D approximation. However, it should be noted that the texture percentage is not recommended as a control parameter to trade-off between bit-rate and image quality because image degradation can be easily detected by human visual system.
Table 1: Summary of Coding Information For the LENA with $\alpha$ of 4.0 and 10.0 and with different percentage of texture pixels

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\alpha=4.0$</th>
<th>$\alpha=10.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{RCI}$</td>
<td>18.0</td>
<td>30.0</td>
</tr>
<tr>
<td>Pct. of texture</td>
<td>60</td>
<td>64</td>
</tr>
<tr>
<td>No. of Regions</td>
<td>274</td>
<td>71</td>
</tr>
<tr>
<td>bits/pixel</td>
<td>0.804</td>
<td>0.681</td>
</tr>
<tr>
<td>error/pixel</td>
<td>5.40</td>
<td>6.56</td>
</tr>
</tbody>
</table>

Table 2: Summary of Coding Information For the HOUSE with $\alpha$ of 4 and 10.0 and with different percentage of texture pixels.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\alpha=4$</th>
<th>$\alpha=10.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{RCI}$</td>
<td>29.0</td>
<td>53.0</td>
</tr>
<tr>
<td>Pct. of Texture</td>
<td>75</td>
<td>85</td>
</tr>
<tr>
<td>No. of Regions</td>
<td>183</td>
<td>59</td>
</tr>
<tr>
<td>bits/pixel</td>
<td>0.988</td>
<td>0.863</td>
</tr>
<tr>
<td>error/pixel</td>
<td>7.50</td>
<td>8.54</td>
</tr>
</tbody>
</table>

5.3 Effects of Adjusting the $T_{run}$

The error threshold ($T_{run}$) determines the absolute error allowed in a run. The larger $T_{run}$ is, the longer the run length would be. Thus, by adjusting the $T_{run}$ we control the error in the 1-D approximation and therefore control the image quality and bit-rate. The texture percentages are selected so that the most HVS sensitive areas of the LENA and HOUSE are classified as texture regions. For each image, experiments with $T_{run}$ value of 20, 50, 75 and 100 are conducted. The resulting bit-rate and error information is shown in Table 3 and Table 4.

We observed that for smaller $T_{run}$ values, the effects on bit-rate and error per unit adjustment is more significant than larger $T_{run}$ values. The effects of adjusting the $T_{run}$ on image quality and bit-rate are smooth, therefore can be used as fine tune for adjusting the image quality.

6. CONCLUSIONS

We have developed a segmentation based image coding system based on a set of design requirements for a packet video environment. A pyramid data structure and a JND merge criterion were used in the segmentation process. The resulting image segmentation algorithm has proven to be robust for a variety of images.

Table 3: Summary of Coding Information For the LENA with $T_{run}$ of 25, 50, 75 and 100. $\alpha=10.0$, $RCI=35$, 53 percent of texture pixels

<table>
<thead>
<tr>
<th>Parameters $T_{run}$</th>
<th>$T_{run}=20$</th>
<th>$T_{run}=50$</th>
<th>$T_{run}=75$</th>
<th>$T_{run}=100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>bits/pixel</td>
<td>0.719</td>
<td>0.589</td>
<td>0.524</td>
<td>0.487</td>
</tr>
<tr>
<td>error/pixel</td>
<td>5.81</td>
<td>6.62</td>
<td>7.42</td>
<td>7.58</td>
</tr>
</tbody>
</table>
Table 4: Summary of Coding Information For the HOUSE with $T_{run}$ of 20, 50, 75 and 100. $\alpha=10.0$, RCI=41, 90 percent of texture pixels

<table>
<thead>
<tr>
<th>Parameters</th>
<th>$T_{run}=25$</th>
<th>$T_{run}=50$</th>
<th>$T_{run}=75$</th>
<th>$T_{run}=100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>bits/pixel</td>
<td>1.12</td>
<td>0.831</td>
<td>0.683</td>
<td>0.599</td>
</tr>
<tr>
<td>error/pixel</td>
<td>7.41</td>
<td>8.82</td>
<td>9.71</td>
<td>10.42</td>
</tr>
</tbody>
</table>

To save bit-rate while maintaining the image quality, we have described a strategy that classify the segmented region into texture and non-texture regions. The texture regions are approximated with one-dimensional polynomials and non-texture regions are approximated by their means intensity. A region complexity indicator (RCI) incorporating HVS property has been developed for texture region classification purpose and its effectiveness demonstrated.

The results of applying the proposed image coding algorithm on the LENA and HOUSE images with different parameters are presented and discussed. We also proposed an strategy for best tradeoffs between the image qualities and the bit-rates. Finally, the performance of the proposed image coding algorithm is verified against the set of design requirements. The packet loss compensation based on this frame work will be a future research topic.

References


