University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Journal Articles

Computer Science and Engineering, Department
of

10-16-2002

Multiresolution, dynamic, and adaptive image quantization methodology: automation and analysis

Leen-Kiat Soh

Follow this and additional works at: <https://digitalcommons.unl.edu/csearticles>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Journal Articles by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Multiresolution, dynamic, and adaptive image quantization methodology: automation and analysis

Leen-Kiat Soh

University of Nebraska

Department of Computer Science and Engineering

Lincoln, Nebraska 68588-0115

E-mail: lksoh@cse.unl.edu

Abstract. We describe a multiresolution, dynamic, and adaptive image quantization methodology with automation being the goal of our research. To improve the robustness of the approach, we incorporate dynamic local thresholding and multiresolution peak detection. The first strategy extracts bisector values from local regions of the image and builds a histogram based on those values. The second strategy maps the derived histogram into multiple levels of resolution, allowing peaks be scored for their significance and localized. We conduct several experiments to analyze different versions of our quantization methodology and to compare it with the equal probability quantization. We also investigated the relationships between image attributes and the key parameters in our quantizers. Based on the findings, we developed a fully automated quantizer called QTR0.5. We have applied QTR0.5 to a variety of images—airial, photographic, and satellite images—and have also used it as a pre-processor in an image segmentation software tool. © 2003 SPIE and IS&T. [DOI: 10.1117/1.1557158]

1 Introduction

As summarized in Ref. 1, the goal of quantization is to encode the data from a source, characterized by its probability density function, into as few bits as possible (i.e., with low rate) in such a way that a reproduction may be recovered from the bits with as high quality as possible (i.e., with small average distortion). Some have also termed quantization as a reduction of psychovisual redundancy^{2,3} to improve efficiency of image representation in gray levels. Psychovisual redundancy is based on the fact that human vision and cognitive systems do not fully comprehend all gray levels and combinations of them when viewing an image. Gray levels not vital to a human's understanding of the image can thus be considered redundant and be eliminated. The inherent dilemma with developing a quantization algorithm lies at the tradeoff between the number of gray levels and the overall quality of the image. The fewer number of gray levels, the better the efficiency of the data reduction is. However, overzealousness in pursuing high efficiency may result in severe distortions to the image quality such as false contours. Thus, there is a tradeoff

between these two primary performance measures: average distortion and rate.¹ We sometimes use efficiency to refer to a low bit rate, and good preservation of the original image to a small average distortion.

The focus of our research in quantization is the automated determination of the number of quantization levels in the image. As reported in Ref. 4, “There seems to be a widespread feeling in the literature that image analysis can often gain by using a relatively small number of colors or ‘true gray levels’.” Also, there are applications where the determination of the number of quantization levels in images can be useful.⁴ First, in medical image segmentation, a segmentation is often desired to delineate organs, tumors, or other pixels in CAT or other images. If the image is segmented into too many colors, it may make the border finding problem more difficult. If the image, on the other hand, is segmented into too few colors, then the borders between objects may be blurred. Also, real-world images typically come in many colors, whereas output devices can often display far fewer. In such cases, quantization is often useful to map the original image into an output image with a limited number of colors, while preserving the image quality. In addition, the correct number of gray levels maximizes the image fidelity in computer-generated holograms. A fourth area of application arises in the use of cooccurrence matrices, where a reduced set of gray levels is often enough to build effective and efficient matrices. Moreover, we see automated quantization as a useful and important tool in image indexing for quick browsing and searching in digital image repositories in today's IT world. Therefore, our research focus is more for automated determination of quantization levels than for improving the quality of quantization results.

We describe a histogram-based, multiresolution, dynamic, and adaptive methodology for performing automated quantization on gray level images. The approach first builds a histogram for local bisector values computed via dynamic local thresholding, then generates a map of peak detection signals from the image histogram, and finally selects peaks as decision points. The multiresolution strategy refines the decision points by analyzing the signals at different resolutions and is noise-resistant. Finally, the meth-

odology is adaptive, since it creates varying quantization intervals based on the contextual information in the image. In Sec. 2, we review some existing quantization schemes. In Sec. 3, we describe the underlying quantization methodology and its two components. In Sec. 4, we present several different designs of our quantization approach. In Sec. 5, we discuss a set of comprehensive experiments to investigate the automation of our quantizers and report on the results, and compare our final, automated design to the equal probability quantization. Finally, we point out the weaknesses and strengths of our design and provide conclusions.

2 Background

In general, a quantization function is represented as $t = q(s)$, where t is the collection of reconstruction levels and s is the collection of decision levels. Reconstruction levels are the quantization levels where decision levels are a subset of the intensity levels of an image. The mapping procedure is $c(s) = t_i$ for $s_i \leq s < s_{i+1}$, $s = 0 \dots 255$. The quantization problem is to determine the best s_i and t_i values that optimize certain criterion.

The most simplistic quantization scheme is uniform quantization, in which all gray levels are divided into same-size intervals. This quantization scheme is not adaptive and does not perform well when the distribution of an image's histogram is not uniform; and most histograms are not uniform. As a result, many have proposed approaches that coarsely quantize low-frequency gray levels and finely quantize high-frequency gray levels. This approach is also generally known as tapered or adaptive quantization. A popular scheme is the Gaussian quantization. This quantization scheme models the histogram of the image as a Gaussian distribution and repairs the histogram to fit the curve. Thus, each quantization interval under the curve is approximately the same. Areas close to the mean will thus be finely quantized, and areas at the tails will be coarsely quantized. However, this quantization strategy applies well only to Gaussian-type histograms. On the other hand, the equal probability quantization⁵ scheme obtains a set of quantization decision points in which each interval has a similar probability.

In Ref. 6, the author used the minimization of the mean-square quantization error s_i and t_i to derive, given a specific distribution, optimization conditions: the reconstruction levels are the centroids of the areas under the distribution over the specified decision levels, and the decision levels are halfway between the reconstruction levels. The corresponding quantizer is called a Lloyd-Max quantizer, and it has been implemented with Gaussian and Laplacian distributions.⁷ The Lloyd-Max quantizers are complex and the solutions to the optimization are difficult to obtain in closed forms. Other methods used to obtain the mapping procedure include: Kohonen neural networks,⁸ local K-means,⁹ dynamic programming and principal analysis,¹⁰ and others. The Kohonen neural network approach comprised neurons that each learned a RGB weight vector for a desired cluster. After learning, the 24-bit colors were quantized into 8-bit colors by mapping image pixels to their respective closest weight vectors. The local K-means algorithm was similar to the Kohonen neural net-

work approach, in which the competitive learning principle was used, but iteratively adapted the initial approximation of the color palette to the distribution of colors in the input image. The dynamic programming approach was an efficient algorithm for color quantization that allowed the users to choose a balance between the image smoothness and hue accuracy for a given number of colors. Note that these procedures assume that *a priori* the number of quantization levels in the results. To counter noise in quantization, some have used iterative and wavelet transform algorithms. Fetthauer and Bryngdahl^{11,12} used a windowed iterative Fourier transform algorithm to vary the spectral control of the quantization noise locally in the image. Also, the quantization information loss was adaptive to the important information found by the images. Others have used iterative quantization to lessen the noise effects (e.g., Ref. 13). However, our approach deals with the noise in the images, and only addresses the quantization information loss (through the measurements of entropy) as part of our evaluation criteria.

3 Methodology

In this section, we describe our quantization methodology. There are two important components in this methodology: dynamic local thresholding (Sec. 3.1) and multiresolution peak detection (Sec. 3.2). The first component is responsible for collecting points for the histogram. Briefly, the points are the collection of bisectors of local regions in the image. The dynamic local thresholding approach was chosen for its noise-resistant and range-resistant properties.¹⁴ The second component is responsible for finding the peaks in the histogram to obtain quantization decision points. The basis of this second module is the utilization of the cumulative distribution function (CDF) of the histogram to detect peaks, which has been proposed in other work to process histograms. For example, Boukharouba, Rebordao, and Wendel¹⁵ located peaks using the zero-crossing and local extrema of a peak detection signal determined from the curvature of the CDF. Subsequently, Sezan¹⁶ refined the method to reduce computational intensity and the number of user-specified parameters. This concept of peak detection using the CDF is to first generate a peak detection signal from the histogram and then locate the histogram peaks using the zero crossings of the peak detection signal and the local extrema between the zero crossings. To obtain the peak detection signal, η_Ω , the histogram, h , is convolved with a peak detection kernel, κ_Ω . The factor Ω is odd and referred to as the peak-detection parameter. The higher the parameter is, the more the histogram is smoothed. To obtain peaks, Sezan¹⁶ used the following to estimate the start, end, and maximum points of the peaks.

1. A zero crossing of the peak detection signal η_Ω to negative values—a negative crossover—signifies the start of a peak. The bin value ω , at which the negative crossover occurs, is defined to be the estimate of a start point. Thus, the start point of the i 'th peak is ω_i^s ; the next negative crossover at the bin ω_{i+1}^s estimates the start of the next peak, and so on.
2. A zero crossing of the peak detection signal η_Ω to positive values—a positive crossover—following a

Fig. 1 An ideal peak detection signal and the parameters of the peaks. Suppose that we are at the i 'th peak. A is the first negative crossover point, and it is the start of the i 'th peak, ω_i^s . D is the second negative crossover point, and it becomes the start of the $i+1$ peak, ω_{i+1}^s . C is the maximum value between the two start points, and it thus is the end of the i 'th peak, ω_i^e . Finally, B is the first positive crossover point following the first negative crossover point, and the location of the maximum of the i 'th peak, ω_i^m .

negative crossover estimates the bin values at which the peak reaches its maximum. The maximum of the i 'th peak is ω_i^m .

3. The bin value between two successive negative crossovers, ω_i^s and ω_{i+1}^s , at which the detection signal η_Ω achieves its local maximum, is defined to be the estimate of the end point of the peak. The end point of the i 'th peak is ω_i^e .

Hence, the i 'th peak can be represented by a triplet of $\langle \omega_i^s, \omega_i^m, \omega_i^e \rangle$. Figure 1 illustrates an ideal detection signal and the triplet parameters. Sezan¹⁶ then extracts the peaks and merges adjacent peaks as decision points for quantization. Figure 2 shows a flow diagram of Sezan's design. Our work is an extension of Ref. 16. In our research, we introduce dynamic local thresholding and multiresolution peak detection to Sezan's algorithm to improve its robustness and to facilitate its automation. A detailed treatment of the peak detection algorithm can be found in Ref. 16.

3.1 Dynamic Local Thresholding

Dynamic local thresholding is a segmentation approach that first divides an image into smaller regions, and then thresholds the regions locally under a global constraint. There are two main advantages that dynamic local thresholding brings to quantization. First, it adapts to local variances, especially instrumental errors such as range sloping in satellite images. Because of its adaptivity, dynamic local thresholding actually yields a set of so-called representative decision points. As these decision points are interpolated to all regions in the image, each pixel will have a set of its own real decision points, adapted to its surrounding pixel values. Second, dynamic local thresholding extracts an optimal bisector value from each high-variance region. We collect these bisectors to build a histogram from which

Fig. 2 The flow diagram of the original Sezan¹⁴ design of a histogram-based image quantization approach.

Fig. 3 Flow diagram of dynamic local thresholding as a quantizer.

peaks are extracted. This makes the subsequent quantization more noise resistant, and more representative of local variances.

Chow and Kaneko¹⁴ first applied dynamic local thresholding as an automated boundary detection method in the analysis of heart ventricles in cineangiograms. We adapted the technique to perform satellite sea ice trinary image segmentation¹⁷ and unsupervised segmentation.¹⁸ Our design of dynamic local thresholding (see Fig. 3) consists of six main stages: 1. regional histogram computation and selection, 2. Gaussian curve approximation, 3. bimodality filtering and optimal threshold determination, 4. peak detection, 5. regional and pixelwise interpolations, and 6. N -ary decision.

First, we divide the image into overlapping regions such that each region shares 50% of the same pixels with each of its four neighbors. We then select the histograms of the regions with variances ranked in the top 25% for further processing. For each selected histogram, we perform a binormal Gaussian curve approximation to curve-fit the histogram, and obtain the following parameters: μ_1 , μ_2 , σ_1 , σ_2 , c_1 , and c_2 , the mean, standard deviation, and coefficient of mixture values of the two Gaussian functions, respectively. Given those parameters, we compute a bimodal measure as

$$\beta = \frac{\min_{i=\mu_1} \hat{f}_i^{\mu_2}(i)}{\min[\hat{f}(\mu_1), \hat{f}(\mu_2)]}, \quad (1)$$

where $\hat{f}$ is the approximated function, and $\hat{f}(\mu_1)$ and $\hat{f}(\mu_2)$ are the peak values (or heights) at the means of the first and second Gaussian functions, respectively. Given these two peaks (at the means), we use $\min[\hat{f}(\mu_1), \hat{f}(\mu_2)]$ to obtain the height of the smaller peak, $\min_{i=\mu_1} \hat{f}_i^{\mu_2}(i)$ to obtain the smallest point between the two peaks, and β as the ratio of the lowest point of the binormal Gaussian normalized by the height of the smaller peak. A small β indicates that the valley separating the two Gaussians is significant. Therefore, we further select regions with bimodality measures smaller than a threshold (currently set at 0.8) and compute, for each such region, a bisector value ϕ , which minimizes the probability of misclassification based on the maximum likelihood method:

$$\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} \right) \phi^2 + 2 \left(\frac{\mu_2}{\sigma_2^2} - \frac{\mu_1}{\sigma_1^2} \right) \phi + \left(\frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2} + 2 \ln \frac{c_2}{c_1} \right) = 0. \quad (2)$$

Solving the quadratic equation for ϕ , by letting $A = 1/\sigma_1^2 - 1/\sigma_2^2$, $B = 2(\mu_2/\sigma_2^2 - \mu_1/\sigma_1^2)$, $C = \mu_1/\sigma_1^2 - \mu_2/\sigma_2^2 + 2 \ln(c_2/c_1)$, we have

Fig. 4 Flow diagram of multiresolution peak detection.

$$\phi = \frac{-B + (B^2 - 4AC)^{1/2}}{2A}.$$

Note that the other root of the solution is either negative or small and thus not used. Equation (2) is designed to obtain the ideal bisector between two Gaussian functions.

We collect all bisector values into a histogram, from which we extract peaks. The peak extraction process obtains a set of representative (or significant) bisector values as the quantization decision points. Up until now, disqualified regions have had no decision points. We interpolate the set of decision points throughout the entire image using regional and pixelwise interpolations. The regional interpolation, a concentric, weighted average of neighboring values, allows local bisector values to influence all other regions and thus imposes an implicit constraint on dramatic quantization changes. After each region is assigned a set of interpolated decision points, we perform bilinear pixelwise interpolation to label each pixel with yet another set of interpolated decision points to ensure continuity at the regional borders. Finally, we quantize every pixel. A pixel, p_i , is assigned a quantization value: $c(p_i) = j$, for $\lambda_{pij} \leq g(p_i) < \lambda_{pik}$, $j = 0 \dots k-1$, where $g(p_i)$ is the gray level value at p_i , and λ_{pij} is the j 'th decision point at the pixel p_i . A detailed treatment of dynamic local thresholding can be found in Ref. 19.

3.2 Multiresolution Peak Detection

One disadvantage of Sezan's algorithm is the determination of the peak-detection parameter Ω . If this number is small, then the algorithm responds to local variations and noise in the histogram; if this number is large, then the algorithm may overlook legitimate peaks because of the averaging effect. To address this issue, we use multiresolution to handle noise and details.^{20,21} First we create a multiresolution map of the histogram peaks. Second, we track each peak through the scale space to score each peak's significance. Third, we filter out spurious peaks and localize significant peaks. The scoring algorithm is based on the following principles: 1. peaks found at low resolution are more significant than the peaks found at high resolution; 2. peaks found at high resolution are more accurate in terms of localization than those found at low resolution; 3. neighboring peaks suggest a significant peak—a peak is more significant if it is surrounded (in a 2- or higher-dimensional space) or flanked (in 1-D space) by other peaks—rather than a peak without such neighbors; and 4. the significance of a peak is proportional to its height.

Figure 4 shows the main modules of our multiresolution peak detection. The first module generates the cumulative distribution function from the histogram. Then the second module computes the maximum peak-detection parameter $\Omega_{\max}$, such that a multiresolution map can be created using $\Omega = 3, 5, 7, \dots, \Omega_{\max}$. The fourth module tracks the peaks to

assess their significance. The fifth module merges adjacent peaks. Finally, a filter extracts a set of peaks to become the set of quantization decision points.

To create a multiresolution map, we apply the peak detection algorithm to the histogram several times with different values of the peak-detection parameter, or scales. We set the minimum scale size to 3. To determine the lowest resolution level, we solve for the maximum scale size $\Omega_{\max}$. The solution is the smallest scale size value that does not incur overaveraging to the histogram to prevent substantial shortening of the image intensity range. This is achieved by smoothing the histogram with a window of size Ω (starting from $\Omega = 3$) until the range of the current, smoothed histogram is smaller than the previous histogram. After obtaining $\Omega_{\max}$, we generate the multiresolution map using $\Omega = 3, 5, 7, \dots, \Omega_{\max}$. For each value of Ω , we compute the smoothed CDF, $\overline{\text{CDF}}_{\Omega}$, and obtain the corresponding peak detection signal η_{Ω} , where

$$\overline{\text{CDF}}_{\Omega}(\omega) = \frac{\sum_{i=\omega-(\Omega-1)/2}^{\omega+(\Omega-1)/2} \text{CDF}(i)}{\Omega}, \quad (3)$$

and

$$\eta_{\Omega}(\omega) = \text{CDF}(\omega) - \overline{\text{CDF}}_{\Omega}(\omega). \quad (4)$$

A zero crossing of the peak detection signal to positive values following a negative crossover estimates the bin value at which a peak reaches its maximum. This is what we have defined previously as ω_i^m . And every such occurrence is a peak. So, for a peak located at ω_i at resolution level Ω , the significance or the weight of the peak is computed as:

$$W_{\Omega}(\omega_i) = \frac{\eta_{\Omega}(\omega_i) - \eta_{\Omega}(\omega_i - 1)}{1 + \eta_{\Omega}(\omega_i) - \eta_{\Omega}(\omega_i - 1)} + \frac{H(\omega_i)}{\max(H) \cdot N_{\text{scale}}}, \quad (5)$$

where H is the histogram, and $N_{\text{scale}} = [(\Omega_{\max} - \Omega_{\min})/2] + 1$. The first term in Eq. (5) measures the dominance of the peak—the magnitude of the positive crossover. The higher the magnitude (the greater the difference between the current bin and the previous bin), the closer the term approaches 1. The second term measures the significance of the peak—the frequency of the bin in the original threshold histogram. The denominator of the second term is a normalization factor. $\max(H)$ is the height of the tallest peak in the histogram and thus $H(\omega_i)/\max(H)$ is always less than or equal to 1. In addition, since we are summing the weights throughout N_{scale} levels, we divide the second term with N_{scale} . The multiresolution contour map is the collection of all peaks across all resolution levels.

Now, we have to track the peaks through various resolution levels. At each resolution level Ω , for a peak situated at ω , we update its weight:

$$W'_{\Omega}(\omega) = W_{\Omega}(\omega) + \sum_{j=\omega-(\Omega-1)/2}^{\omega-1} \frac{P_{\Omega}(j)}{|j-\omega|} + P_{\Omega}(\omega) + \sum_{j=\omega+1}^{\omega+(\Omega-1)/2} \frac{P_{\Omega}(j)}{|j-\omega|}. \quad (6)$$

If a peak is found at a location ω , then $P_{\Omega}(\omega)$ is 1; otherwise it is 0. The two summation terms collect the neighboring peaks as evidence for ω to be a peak, weighted by the distance of those peaks from ω . This satisfies one of our principles that a peak is more likely if it is found in a cluster of other peaks than if it is isolated. The single term $P_{\Omega}(\omega)$ acts as a self-assurance weight. For the same location ω , there may not be a peak at all scales. By adding this term into the equation, we assure that if a peak exists at that same location across the scale space often enough, then it might be preserved as a peak. We accumulate all weights to obtain

$$W_{\Omega_{\min}, \Omega_{\max}}(\omega) = \sum_{\Omega=\Omega_{\min}}^{\Omega_{\max}} W_{\Omega}(\omega). \quad (7)$$

Note that our tracking approach—the process of fusing information across the scale space—is similar to those used in Refs. 22–24. We use the whole scale space but do not modify the information points, i.e., peaks in our case. When we combine the peaks, we use a higher-order attribute to accumulate the peaks. In our case, we score each peak's significance by summing up all occurrences of peaks at each intensity level based on weights. There are two other general approaches to tracking. One approach follows the information at the lowest resolution level to the highest or until the current trail has no way to continue, and has been employed in focusing edge pixels,^{21,25,26} the validation of edge pixels,^{27–29} and feature extraction.³⁰ The other approach interpolates all information points together to form a continuous sheet and has been used in wavelet representation of multiresolution.^{31,32}

If there is a group of adjacent peaks, then we locate the most significant peak within the group, assign to the most significant peak the total weights of all peaks in the group, and remove all other peaks in the group by setting their weights to zero.

To distinguish between weak and strong peaks, we compare each nonzero $W_{\Omega_{\min}, \Omega_{\max}}(\omega)$ to a high-pass threshold: $W_{\text{qualify}} = \alpha N_{\text{scale}}$. Note that this factor does not influence the process of peak detection, but only affects the number of selected peaks. A detailed treatment of our multiresolution peak detection design can be found in Ref. 18.

4 Various Quantizers

So far, we have described the underlying methodology of our quantization algorithm: one that derives a peak detection signal from an image histogram and extract peaks from the signal as quantization decision points. That is essentially the basic Sezan algorithm.¹⁶ To improve the quantizer, however, we have incorporated dynamic local thresholding (Sec. 3.1) and multiresolution peak detection (Sec. 3.2), and, as a result, have implemented four different quantizers. Q is the basic Sezan image histogram-based quanti-

Fig. 5 Photographs, from left to right, top to bottom: (a) girl, (b) couple, (c) buildings, (d) map, (e) moon, (f) x-ray, (g) airport, (h) street, (i) mandrill, (j) crowd, (k) photo, (l) sleep, (m) airplane, (n) city, (o) calibrated, and (p) port.

zation. QT uses Sezan's histogram-based quantization but with a histogram generated from bisector values computed via dynamic local thresholding. QR uses Sezan's image histogram-based quantization with multiresolution peak detection. Finally, QTR combines Sezan's algorithm with both dynamic local thresholding and multiresolution peak detection. When running Q and QT, one has to provide the peak-detection parameter Ω . When running QR and QTR, one has to provide the weight factor α .

5 Experiments and Analyses

We have conducted several experiments to evaluate the four quantizers, Q, QT, QR, and QTR. The goals of these experiments were threefold. First, we wanted to evaluate the visual quality of the quantization results of the different results. Second, we wanted to find out how important the peak detection parameter (Ω) and the weight factor (α) are in the four quantizers. We wanted to find out whether there was a quantizer that behaved consistently under a range of Ω values (for Q or QT) or a range of α values (for QR or QTR). If such a quantizer was observed, that would mean the parameter(s) played a minor role in the quantization process and thus the quantizer could be automated with high generality or applicability. In addition, we also wanted the eventual, automated quantizer to be efficient, and thus we examined other attributes of the quantization process as well. Third, we wanted to investigate whether Ω or α could be determined automatically based on some intrinsic image attributes.

Figures 5, 6, and 7 show the images that are used in the experiments. There are 16 images in Fig. 5, each 128×128 in dimension. In addition, the juxtaposition of the 16 images forms an image of 512×512 in dimension, called contact. Figure 6 features a Radarsat ScanSAR wide image in the gulf of Alaska, taken on 17 February 1997, showing ice and the signature of atmospheric rolls. We call this image roll. Figure 7 features an ERS-1 SAR sea ice image, taken on 17 March 1992, at 72.85°N and 143.83°W . We

Fig. 6 The roll image, a Radarsat ScanSAR wide image, in the Gulf of Alaska, on 17 February, 1997 (© ESA).

call this image ice. The size of the two satellite images is about 1000×1000 pixels each. Table 1 shows some intrinsic measurements of the images.

We define a set of terms to measure the image characteristics and the quantization process for automation analysis.

1. Tolerance. For a Q or QT design, this is based on the maximally acceptable value of the peak-detection parameter (Ω). To obtain this value, we run Q or QT on an image using a range of Ω values. Then we visually inspect the quantization results. As the value of Ω increases, the quantization result deteriorates (e.g., Fig. 8). The highest value of Ω that still yields an acceptable quantization result then is the tolerance value. Similarly, for a QR or QTR design, the value is based on the maximally acceptable value of the

Fig. 7 The ice image, an ERS-1 SAR sea ice image, taken on 17 March, 1992, at 72.85°N and 143.83°W . (© ESA).

weight factor (α). A design with a high tolerance implies that the accurate determination of the parameter is of low importance and the design may be automated with high generality. The normalized tolerance value is the actual tolerance divided by the scale of either Ω or α , for comparisons among all four quantizers.

2. Number of decision points. This measures the number of quantization decision points in a quantized image. If the number of decision points is N , then the number of quantized levels in the image is $N + 1$. For efficiency, we like to have a low number of decision points for our quantized images.
3. Decision point preservation rate. We run a quantizer on a range of Ω (or α) values. As either of the values increases, the number of decision points decreases. The decision point preservation rate is the ratio of the final number of decision points (from the last acceptable quantized result) to the initial number of decision points. A high decision point preservation rate (meaning the initial and final numbers of decision points are similar) is desirable as it implies that Ω (or α) influences the quantization process only peripherally and its accurate determination is less required.
4. Decision point preservation curve. The curve that connects each number of decision points from the first to the last for each image. A concave curve (that drops sharply) is desirable, since that means the choice of Ω (or α) stabilizes quickly after some initial offsets. Figure 9 shows one example of the decision point preservation curve.
5. Standard deviation preservation rate. This measures the standard deviation of all decision point preservation rates for a particular design on all test images. A design with a low standard deviation is desirable for automation, since that implies predictability in the quantized results given a Ω (or α) value.
6. Drop. The difference in the number of decision points between the two runs using successive values of Ω (or α). A design with a smaller average of drops is more desirable for automation, since it is more stable. Compared to the decision point preservation rate of 3., this measures the difference in the number of decision points between one value of Ω and the next, while the preservation rate measures the difference between the first value of Ω and the last. In a way, the drop measures the stability between neighboring resolutions, while the preservation rate measures how the image reacts to the range of resolution scales.

In general, a good design that is suitable for automation is one that uses a parameter that has a high tolerance, with a high decision point preservation rate, a small standard deviation preservation rate, and a small drop. For efficiency, a good design also requires the number of decision points to be as low as long as the quantization quality is still acceptable. In the following visual evaluation, we performed the judgment and the results were further verified by a second person.

Table 1 Intrinsic measurements of images: (a) number of gray levels, (b) minimum gray level, (c) maximum gray level, (d) dynamic range, (e) density=number of levels/dynamic range, (f) average image intensity, and (g) standard deviation of image intensity.

images	# levels	min	max	range	density	average	std. dev.
airplanes	249	0	248	249	1	124.81	44.57
airport	232	15	255	241	0.96	66.89	37.72
buildings	223	1	249	249	0.90	140.08	45.88
moon	232	0	249	250	0.93	127.64	27.34
city	182	68	255	188	0.97	158.46	37.85
couple	114	1	240	240	0.48	40.37	34.66
crowd	253	3	255	253	1	61.06	56.45
girl	127	1	255	255	0.50	73.53	43.08
mandrill	255	0	254	255	1	112.61	61.94
map	179	0	254	255	0.70	226.84	34.06
calibrated	252	4	255	252	1	168.94	75.06
photo	60	19	255	237	0.25	138.90	73.19
port	176	14	255	242	0.73	109.65	45.85
sleep	245	3	254	252	0.97	45.138	46.89
street	149	0	252	253	0.59	66.823	16.80
xray	69	0	183	184	0.38	181.21	44.78
contact	256	0	255	256	1	115.11	70.43
roll	208	0	255	256	0.81	61.36	14.73
ice	210	11	224	214	0.98	110.96	31.32

5.1 Experiment 1: Quantizer Q

The objective of this experiment was to identify, if any, correspondence between the Ω values and the intrinsic values of the images as tabulated in Table 1. We ran Q 17 times on each image, using Ω of 3, 5, 7, ..., and 35. We then evaluated visually the quality of the 17 quantization results

of each image. In general, all decision point preservation curves are concave. Each drops sharply and then dies out.

We did not identify any significant correlation between the intrinsic image measurements and Ω , meaning that Ω cannot be derived automatically from intrinsic image measurements. As the value of Ω increases, the number of de-

Fig. 8 Quantized results using Q with a range of Ω values from 3 to 35. Top to bottom, left to right: The first image is the original image. The second image is Q with $\Omega=3$, and so on. The last acceptable quantized image is Q with $\Omega=17$ (second row, third column). So the tolerance parameter value is 17. Quantization results after that point begin to have significant false contours and region mergings.

Fig. 9 An example of a decision point preservation curve. This curve is derived from image airplanes.

cision points decreases. The rate of the decrease does not correlate with the intrinsic image measurements either, except for the number of gray levels where there is a noticeable trend: the greater the number of gray levels, the lower the rate. This means that an image with a high number of gray levels is more susceptible to the Ω value than one with a low number of gray levels.

5.2 Experiment 2: Quantizer QT

We performed a similar experiment with quantizer QT and obtained similar observations as those with quantizer Q. We observed that the number of gray levels no longer correlates with the decision point preservation rate. This might be due to the fact that dynamic local thresholding extracts a group of bisector values and QT performs its peak detection on that group of values instead of on the entire image.

5.3 Comparison 1: Quantizer Q versus Quantizer QT

In this study, we compared quantizers Q and QT to examine the suitability of the quantizers for automation. Table 2 shows the statistics associated with the results of the prior two experiments. We observed the following.

1. Quantizer QT is more resistant and has more flexibility in selecting a satisfactory Ω . It has a higher tolerance: 20.74 to 18.63.
2. Quantizer QT is more economical. It has a lower average number of decision points: 11.07 to 24.32.
3. Both quantizers Q and QT are equally successful. Each failed to quantize an image, in which the quantization result did not preserve the original image satisfactorily.
4. Quantizer QT deals with satellite images better (roll and ice). This is expected, since dynamic local thresholding handles large satellite images better than small photographs.
5. Quantizer QT has an average starting number of decision points of 26.84, and an average preservation rate of 0.21. Quantizer Q has 66.84, and 0.16, respectively.

In general, QT is more robust than Q. It is also more economical in terms of the number of decision points and decision point preservation rates. More significantly, the accurate determination of the peak-detection parameter (Ω) is less important in QT than in Q. From this comparison, we see that QT is more suitable for automation than Q.

Table 2 Comparison between quantizers Q and QT. The tolerance parameter indicates the maximally acceptable Ω value, and # of decision points is the corresponding number of decision points.

Images	Quantizer Q		Quantizer QT	
	tolerance	# of decision points	tolerance	# of decision points
airplanes	17	16	19	12
airport	5	44	17	7
buildings	27	11	21	6
moon	13	15	19	15
city	13	15	21	5
couple	31	20	21	5
crowd	15	43	15	13
girl	17	59	25	9
mandrill	27	7	21	16
map	Bad (0)	11	Good (35)	9
calibrated	29	11	Good (35)	14
photo	Good (35)	36	Good (35)	8
port	31	19	25	8
sleep	17	20	21	16
street	7	22	17	10
xray	Good (35)	31	Bad (0)	9
contact	23	29	23	11
roll	3	46	3	9
ice	9	7	21	4
Average	18.63	24.32	20.74	11.07

5.4 Experiment 3: Quantizer QR

The objective of this experiment was to identify any correspondence between the weight factor (α) values and the intrinsic values of the images as listed in Table 1. We ran QR (the quantizer with multiresolution peak detection) 20 times on each image, using α values of 0.1, 0.2, 0.3,..., and 2.0.

We observed the following. Each of the image measurements—number of gray levels, dynamic range, density, average, and standard deviation—does not correlate with the tolerance parameter. However, larger average values tend to have smaller tolerance parameter values, and larger standard deviation values tend to have larger tolerance parameter values. In general, the greater the number of gray levels is, the higher the preservation rate is, and larger image standard deviation values tend to yield higher preservation rates as well. All decision point preservation curves are concave. Each drops sharply and then dies out. All curves are more tightly clustered together, compared to those of quantizers Q and QT.

We did not identify any significant correlation between any of the intrinsic image measurements and α , indicating that the α value cannot be determined by looking at only those measurements. As the value of the factor increases, the number of decision points increases, as expected, since the filter becomes less stringent. The rate of the increase does not correlate with the measurements either, except noticeably for the number of gray levels. This means that an image with a high number of gray levels is more susceptible to α than an image with a low number of gray levels.

Table 3 Comparison between quantizers Q and QR. The tolerance parameter indicates the maximally acceptable Ω value for quantizer Q and maximally acceptable α value for quantizer QR, and # of decision points is the corresponding number of decision points.

Images	Quantizer Q		Quantizer QR	
	tolerance	# of decision points	tolerance	# of decision points
airplanes	17	16	0.6	31
airport	5	44	0.1	54
buildings	27	11	Good (2.0)	22
moon	13	15	1.0	20
city	13	15	1.2	19
couple	31	20	1.4	43
crowd	15	43	0.7	55
girl	17	59	Good (2.0)	32
mandrill	27	7	Good (2.0)	23
map	Bad (0)	11	Bad (0)	
calibrated	29	11	1.3	31
photo	Good (35)	36	1.5	39
port	31	19	Good (2.0)	34
sleep	17	20	0.2	50
street	7	22	0.1	28
xray	Good (35)	31	1.6	32
contact	23	29	1.3	48
roll	3	46	Good (2.0)	13
ice	9	7	0.5	4
Average	18.63	24.32	1.13	32.11

Finally, we observed that the peak detection algorithm tends to detect more peaks in the beginning of the CDF, since it is more volatile, compared to the end of the CDF, where values start to marginalize. In other words, the peak detection algorithm tends to detect more decision points in a dark image than in a bright image.

5.5 Comparison 2: Quantizer Q versus Quantizer QR

In this study, we compared quantizers Q and QR to investigate the suitability of the designs for automation. Table 3 shows the statistics associated with the results of experiments 1 and 3. The average tolerance for quantizer QR is 1.13 on a scale of 2.0, and for quantizer Q is 18.63 on a scale of 35. Normalizing the two tolerance values gives us 0.57 and 0.53, respectively. This shows that the flexibility in tuning the weight factor (α) in QR is slightly higher than tuning the peak-detection parameter (Ω) in Q. We also observed the following:

1. QR extracts a higher number of decision points, on average, due to its multiresolution approach that admits more decision points, 32.11 to 24.32.
2. QR yields a higher average of preservation rates, 0.29 to 0.16. This tells us that the impact of α on the quantization results is lower than that of Ω . Thus, one has more freedom in choosing a satisfactory α for general image quantization tasks.

3. Q has a higher average of starting number of decision points, 66.84 to 60.89. This shows that quantizer QR is able to filter out some noisy decision points at the first resolution.

5.6 Experiment 4: Quantizer QTR

We performed a similar experiment with quantizer QTR and observed neither correlations nor noticeable trends between the intrinsic image measurements (as listed in Table 1) and the tolerance parameter and the decision point preservation rate. All previously observed tendencies were absent, implying that QTR is a consistent and general quantizer. In general, all preservation curves are very stable and conformed, compared to Q, QT, and QTR.

5.7 Comparison 3: Quantizer QT versus Quantizer QR versus Quantizer QTR

In this study, we compare quantizers QT, QR, and QTR on their suitability for automation. Table 4 shows the statistics associated with the results of the previous experiments. We observed the following.

4. Quantizer QT extracts fewer decision points than does QTR, and QTR extracts fewer than QR. This means that dynamic local thresholding-based techniques help reduce the number of decision points, making the design more economical. This concurs with the observations in Comparison 2.
5. Quantizer QTR is the most successful. It was able to process all images satisfactorily.
6. When comparing the preservation curves of the three quantizers we found the following. QTR has the most controlled curves, and the number of decision points never increases as higher weights are used. On the other hand, the curves in QT and QR are more chaotic and less predictable, making them less suitable for automation. QTR has the most compact and stable cluster of curves, implying the most consistent quantization behavior among the three designs.

5.8 Comparison 4: All Quantizers

In this study, we compared all four quantizers: Q (original Sezan design with a histogram of image values), QT (with a histogram of bisector values based on dynamic local thresholding), QR (with a histogram of image pixel values and multiresolution peak detection), and QTR (with a histogram of bisector values based on dynamic local thresholding and with multiresolution peak detection). Table 5 shows the statistics associated with the results of experiments 1, 2, 3, and 4.

Here are some observations. Comparing QT to Q, and QTR to QR, we see that dynamic local thresholding techniques have a much smaller average number of decision points, a much smaller average starting number of decision points, and a higher flexibility, based on the normalized tolerance values. This is because the dynamic local thresholding process uses only the regional bisectors and not the entire image pixel values. It is more selective and more noise resistant. Comparing QR to Q, and QTR to QT, multiresolution peak detection techniques have higher average preservation rates, better clusters of preservation rates (see

Table 4 Comparison among quantizers QT, QR, and QTR. The *tolerance* parameter indicates the maximally acceptable Ω value for quantizer QT and maximally acceptable α value for quantizers QR and QTR. # of decision points is the corresponding number of decision points.

Images	Quantizer QT		Quantizer QR		Quantizer QTR	
	tolerance	# of decision points	tolerance	# of decision points	tolerance	# of decision points
airplanes	19	12	0.6	31	2.0	12
airport	17	7	0.1	54	1.9	10
buildings	21	6	2.0	22	2.0	16
moon	19	15	1.0	20	2.0	16
city	21	5	1.2	19	1.1	16
couple	21	5	1.4	43	1.4	8
crowd	15	13	0.7	55	2.0	23
girl	25	9	2.0	32	1.4	18
mandrill	21	16	2.0	23	1.7	19
map	Good (35)	9	Bad (0)		2.0	11
calibrated	Good (35)	14	1.3	31	1.4	19
photo	Good (35)	8	1.5	39	0.8	20
port	25	8	2.0	34	1.4	12
sleep	21	16	0.2	50	1.3	26
street	17	10	0.1	28	1.1	18
xray	Bad (0)	9	1.6	32	1.9	19
contact	23	11	1.3	48	1.3	14
roll	3	9	2.0	13	2.0	5
ice	21	4	0.5	4	0.7	8
Average	20.74	11.07	1.13	32.11	1.55	16.11

the standard deviation preservation rate), a higher stability, based on the smaller average drop measurements, and a higher flexibility, based on the normalized tolerance values. We see that the weight factor (α) is a more general and less influential parameter than the peak-detection parameter (Ω), implying that its accurate determination is less required and its automation is more applicable. Note also that, unlike Ω , α does not interfere with the peak detection process, since it is only used at the end of the detection as a filter. This makes the multiresolution designs more suitable for automation. In addition, neither Ω nor α can be determined automatically based on simple image attributes. We have observed no strong correlations between the quantization behaviors and the intrinsic image measurements

listed in Table 1. Finally, the QTR design is the most stable (average preservation rate and average drop), tolerant (normalized tolerance), consistent (standard deviation preservation rate), general (as it did not exhibit any noticeable tendencies or trends between image measurements and the weight factor or preservation curve in experiment 4), and successful (number of bad quantization results). We conclude that the QTR design, combining dynamic local thresholding and multiresolution peak detection, is the most suitable for automation.

5.9 Experiment 5: QTR0.5 versus the Equal Probability Quantizer

In this experiment, we compared our final version of quantizer (QTR), using both multiresolution peak detection and dynamic local thresholding, with the weight factor α set to a constant 0.5, to the equal probability quantizer (EPQ).⁵ Note that we chose the equal probability quantizer for several reasons. First, it is easy to implement. Second, it is one of the few quantizers that quantizes an image into n number of quantization levels without fixing the value of n to a power of 2. Finally, the objective of this experiment is to show the feasibility of our technique in determining the number of quantization levels automatically and not improving the quality of quantization results. We call our final, fully automated quantizer QTR0.5. Since EPQ is not automated, we devised the experiment as follows. First, we ran QTR0.5 on all 19 images and recorded the number of quantization levels generated automatically by QTR0.5. Second, we ran EPQ using the same number of quantiza-

Table 5 Comparison of all quantizers, Q, QT, QR, and QTR, in automation analyses.

Measurements	Q	QT	QR	QTR
Average # of decision points	24.32	11.07	32.11	16.11
Average preservation rate	0.16	0.21	0.29	0.31
Average starting # of decision points	66.84	26.84	60.89	35.31
Normalized tolerance	0.53	0.59	0.57	0.77
Number of bad quantization results	1	1	1	0
Std. dev. preservation rate	0.16	0.12	0.11	0.12
Average drop	0.053	0.049	0.037	0.036

Table 6 Comparison between quantizers QTR0.5 and EPQ. #shared is the number of shared decision points between the quantizers. Similarity is #shared/#decision points. Darkres is the number of decision points detected by QTR0.5 that are darker than the first decision point detected by EPQ. Negative values denote decision points by EPQ that are darker than the first decision point by QTR0.5. Briteres is the number of decision points detected by EPQ that are brighter than the last decision point detected by QTR0.5. Negative values denote the decision points by QTR0.5 that are brighter than the last decision point by EPQ. The efficiency percent is $[1 - (\#decision\ points + 1)/\#levels] * 100\%$, indicating the savings in the number of gray levels.

Images	# decision points	# shared	Similarity	Darkres	Briteres	Efficiency %
airplanes	31	6	0.19	15	6	87.15
airport	19	9	0.47	-1	1	91.38
buildings	25	6	0.24	13	4	88.34
moon	32	12	0.37	12	-1	85.78
city	21	7	0.33	-1	2	87.91
couple	15	11	0.73	-1	1	85.96
crowd	33	10	0.30	-1	-1	86.56
girl	29	13	0.45	-1	-1	76.38
mandril	33	10	0.30	5	3	86.67
map	36	2	0.06	8	24	79.33
calibrated	31	4	0.13	-1	12	87.30
photo	28	9	0.32	-2	8	51.67
port	20	5	0.25	-1	5	88.07
sleep	30	16	0.53	-1	1	87.35
street	22	7	0.32	7	-3	84.56
xray	40	11	0.28	16	16	40.58
contact	22	7	0.32	1	-3	91.02
roll	9	3	0.33	-3	-2	95.19
ice	13	3	0.23	-1	1	93.33
average	25.74	7.95	0.32	3.32	3.84	82.87

tion levels obtained from QTR0.5 on each image. Table 6 shows some numeric results of the experiments to investigate the similarity or differences between the two techniques. We see the following.

1. The two quantizers share only about 32.5% of the decision points. A share means a distance of less than two intensity levels between two decision point values. This shows that the two quantizers differ significantly in their decision point determinations.
2. Quantizer QTR0.5 tends to have more decision points at the darker range (darkres). It has more instances (an average of 3.32) where it quantizes the darker range of an image more finely than EPQ.
3. Quantizer EPQ tends to have more decision points at the brighter range (briteres). It has more instances (an average of 3.84) where it quantizes the brighter range of an image more finely than QTR0.5.

In addition, we see quantization as a useful tool in traditional image processing areas such as segmentation and classification, and in digital image libraries such as quick browsing and content-based searching. A quantization technique that can effectively (while not distorting the original image too much) reduce the number of gray levels in an image and boost the image's homogeneity and contrast is desirable. An image with high homogeneity and contrast is

easier to segment; and a massive repository of such images is easier to browse through as well. Thus, we compared the quantization results based on contrast and homogeneity, as shown in Table 7. In general, QTR0.5's quantization results have the most contrast and homogeneity among the three sets of images (the original, the EPQ quantized, and the QTR quantized). Higher contrast and homogeneity values indicate that a feature in the QTR0.5 image is more compact, and, thus, features or regional integrity are better preserved (or enhanced). From this study, we conclude that QTR0.5 is more applicable than EPQ in generating quantized images with better visual cues (more compact and high contrast) for applications in quick browsing and searching and image segmentation and classification.

Further, we performed a visual evaluation of the quantization results, as shown in Table 8. In general, QTR0.5 handles dark images better than EPQ; and on some bright images, the QTR0.5 method does not do as well, for example, the x-ray image. QTR0.5 deals with the map image extremely well, and EPQ fails. QTR0.5 is also superior in handling satellite images (roll and ice), because these images are inherently noisy and suffer from range effects that are addressed by the dynamic local thresholding and multiresolution peak detection in QTR0.5. Figure 10 shows the images where either QTR0.5 or EPQ performed markedly better.

Figure 11 shows the EPQ and QTR0.5 quantization re-

Table 7 Contrast and homogeneity of the quantized results.

Image	Contrast			Homogeneity		
	Original	EPQ	QTR0.5	Original	EPQ	QTR0.5
airplanes	2268.74	2254.21	2935.99	0.085	0.236	0.380
airport	2375.21	2440.37	2439.14	0.067	0.185	0.229
buildings	3323.62	3384.49	4624.85	0.053	0.217	0.321
moon	1774.39	1776.37	2082.62	0.077	0.166	0.281
city	5157.92	5182.50	5665.65	0.055	0.146	0.171
couple	1229.63	1250.85	1439.36	0.182	0.412	0.434
crowd	2298.21	2343.76	2214.08	0.111	0.182	0.263
girl	1081.47	1090.27	1147.77	0.132	0.265	0.306
mandrill	3368.73	3396.94	3435.89	0.055	0.126	0.199
map	4893.39	4678.64	7959.05	0.245	0.284	0.759
calibrated	2458.69	2388.49	4040.93	0.216	0.351	0.513
photo	685.37	677.62	1254.12	0.364	0.484	0.566
port	2565.14	2592.94	3654.89	0.099	0.258	0.351
sleep	289.01	303.83	353.67	0.240	0.404	0.531
street	2452.30	2448.89	2700.78	0.089	0.225	0.291
xray	3253.52	3232.03	5297.25	0.113	0.139	0.502
contact	1126.83	1137.95	1212.10	0.172	0.392	0.424
roll	4324.25	4464.43	5339.51	0.119	0.235	0.431
ice	2250.96	2297.15	2233.56	0.059	0.268	0.366
average	2483.02	2491.67	3159.54	0.133	0.262	0.385

sults of the contact image. They are very similar. The QTR0.5-quantized image has a slightly higher contrast (1212.1 to 1137.9) and homogeneity (0.42 to 0.39), indicating that it has a better regional integrity and more noticeable visual cues. The EPQ-quantized image has better mean-square-error distortion (17.7 to 19.6), indicating that it is more truthful to the original image. The images are very similar, with a mean-square-error difference of less than 5.2 out of a scale of 256 gray levels.

Finally, we also compute the peak signal-to-noise ratio (PSNR) values comparing the results of EPQ and QTR0.5 using the following formula:³³

$$\text{PSNR} = 20 \log_{10} \left(\frac{2^n}{\text{MSE}^{1/2}} \right), \quad (8)$$

where

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [I(i,j) - \hat{I}(i,j)]^2, \quad (9)$$

and I is the original image, $\hat{I}$ is the quantized image, with pixel indices $1 \leq i \leq M$ and $1 \leq j \leq N$, image size $N \times M$, and n bits per pixel. Table 9 shows the results. We see that QTR0.5 yields smaller PSNR values than EPQ for all images except map. However, we see that, compared to the supervised EPQ, the fully automated QTR0.5 is able to hold its ground (with a difference of less than 1) in about 10 out of 19 images.

From Table 6, we see that Quantizer QTR0.5 is about 82.9% efficient in its savings in the number of gray levels. As discussed previously, a good quantizer balances between its representation of the original image and its efficiency. Our QTR0.5 design is comparable to EPQ in its quantization quality and fairly high in its efficiency.

Table 8 Visual comparison between quantizers QTR0.5 and EPQ. An exclamation point indicates that the better method yields a significantly better result.

Images	Better?	Images	Better?	Images	Better?	Images	Better?
airplanes	QTR0.5	couple	same	calibrated	EPQ	xray	EPQ(!)
airport	same	crowd	same	photo	EPQ	contact	same
buildings	QTR0.5	girl	same	port	same	roll	same
moon	same	mandrill	same	sleep	EPQ	ice	QTR0.5
city	same	map	QTR0.5(!)	street	same

Fig. 10 Original images, EPQ-quantized images, and QTR0.5-quantized images. (a) Airplanes: QTR0.5 is better for better contrast and regional integrity (runways and airplanes). (b) Buildings: QTR0.5 is better for better contrast and regional integrity (roads and rooftops). (c) Map: QTR0.5 is better. (d) Calibrated: QTR0.5 is worse because of false contours. (e) Photo: QTR0.5 is worse because of false contours. (f) Sleep: QTR0.5 is worse because of false contours. (g) X-ray: QTR0.5 is worse because of underquantization of bright gray levels.

6 Conclusions

We describe an automated quantization methodology that is histogram-based. To improve its robustness, we perform dynamic local thresholding to extract regional bisector val-

ues and subsequently apply multiresolution peak detection to identify significant peaks from the histogram of those bisector values as quantization decision points. As a result, the quantization is also dynamic and adaptive to a variety of images.

We have designed and implemented four quantizers Q, QT, QR, and QTR, and performed a set of comprehensive experiments to evaluate the performance of these quantizers. We have concluded that each of the multiresolution and dynamic local thresholding approaches improves the quantization robustness of the original Sezan's design. In addition, the synergy of the two facilitates the automation of our quantization technique and has resulted in the implementation of QTR0.5. The results of QTR0.5 are comparable to EPQ. QTR0.5 generates quantized images with more homogeneity and contrast, suitable for applications in image segmentation and classification and quick browsing and searching. But, QTR0.5 produces quantization results with generally worse PSNR values than a supervised EPQ. However, QTR0.5 is on par with the supervised EPQ on 10 out of 19 test images. Under subjective visual evaluation, QTR0.5 performs generally better on dark images and EPQ on bright images. QTR0.5 is also fairly good at reducing the number of gray levels at about 82.9% efficiency. In addition, from our experiments, we have also noticed that there are no simple intrinsic image attributes (dynamic range, intensity average, density, standard deviation, and number of levels) that can be used to predict the weight factor or the peak-detection parameter. Finally, we have concluded that our QTR0.5 design is a good automated quantizer.

Our QTR design incorporates dynamic local thresholding and multiresolution peak detection. The dynamic local thresholding technique benefits the quantization because it is adaptive, local, and noise resistant. It is especially beneficial when the image is large, noisy, or has range dropoff from one end of the image to another. In fact, we have applied QTR0.5 as a preprocessor extensively and successfully to satellite synthetic aperture radar (SAR) sea ice images,¹⁸ where images are noisy and suffer from range distortion. The multiresolution peak detection algorithm has several advantages. First, it incorporates multiple resolutions to investigate the validity of a peak, and that makes it noise resistant. Second, it is virtually automated. The weight factor that users need to specify does not influence the peak detection process. In addition, the weight factor's influence on quantization is significantly less than the peak-detection parameter in Sezan's original design. These two advantages give users more flexibility in choosing a satisfactory weight factor. Note also that we realize that our approach—the analysis on the number of decision points, tolerance, preservation rate, etc.—to determining a quantization parameter that is tolerant, general, and applicable to a variety of images, can be applied to other supervised quantization algorithms such as EPQ for automation. But we also believe that dynamic local thresholding and multiresolution peak detection play a major role in facilitating the design of such a quantization parameter, which is the weight factor in our case.

QTR0.5 has several weaknesses. First, it sometimes introduces false contouring, which can be countered by the IGS strategy of Ref. 2 that breaks up the false contour

Fig. 11 EPQ- (left) and QTR0.5- (right) quantized results of the contact image.

edges by adding to each pixel a pseudorandom number before quantization. Second, QTR0.5 is computationally intensive. The dynamic local thresholding's regional and pixel-wise interpolations are nontrivial processes, as well as the multiresolution peak detection. We ran QTR0.5 on a SunOS Sparc-5, 256 M RAM machine. It averaged 5.78 sec on the 16 128×128 images, and ran 87.16 sec for the 512×512 contact image, and averaged 96.85 sec on the two satellite images (1024×1024 and 896×980 , respectively). Table 10 summarizes the memory requirements and

Table 9 PSNR values of the results of EPQ and QTR0.5. In general, QTR0.5 has a smaller PSNR, except for the image map.

Images	EPQ	QTR0.5
airplanes	18.32	12.79
airport	10.04	9.32
buildings	18.39	9.32
moon	14.22	12.60
city	14.79	11.44
couple	7.86	7.49
crowd	10.21	9.92
girl	11.52	11.02
mandrill	13.57	13.32
map	7.06	10.14
calibrated	15.47	12.92
photo	28.59	16.90
port	15.93	11.78
sleep	8.59	8.30
street	9.18	8.44
xray	10.62	10.54
contact	21.95	21.49
roll	8.22	6.43
ice	13.97	12.49
average	13.61	11.59

complexity of EPQ and QTR0.5. Even though QTR0.5's complexity is linear, it does require more memory and complexity than EPQ. We are now optimizing our source code to improve the speed of QTR0.5. We also plan to investigate the automated determination of the optimal weight factor in the QTR design. As we have observed, the determination cannot be based on simple, intrinsic image attributes. We plan to investigate more complicated image attributes, such that the future version of QTR is able to compute the optimal weight factor for each image. We also plan to apply QTR0.5 as a preprocessor for color image quantization techniques to automatically obtain the number of quantization levels.

Finally, we acknowledge that in radar images where multiplicative noise is present, due to the different mean intensities, the corresponding histogram peaks may have different amplitudes and widths. In that case, our peak detection algorithm would extract a different set of peaks from the multiplicative noisy image, compared to the corresponding image without such noise. Despite that our approach is noise resistant due to its multiresolution strategy in peak localization, since it is adaptive to the histogram of the image and if the histogram of an image were vastly different as a result of multiplicative noise, it would not be able to extract the original, noise-free peaks.

Table 10 Memory requirements and complexity of EPQ and QTR0.5. N is the number of image pixels, L is the number of gray levels, K is the number of local regions, M is the maximum number of peaks, and R is the number of resolution scales.

	EPQ	QTR0.5
Memory requirement	Roughly $N+2L$	Roughly $N+L+5KM+M$
Complexity	Roughly $2N+2L$	Roughly $3N+4K+5L+3KR$

Acknowledgments

The research was sponsored in part by Naval Research Laboratory contract N00014-95-C-6038. The author would like to thank Costas Tsatsoulis for improving the paper and the anonymous reviewers for their insightful and useful comments.

References

1. R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Info. Theory* **44**(6), 1–63 (1998).
2. W. T. Bisignani, G. P. Richards, and J. W. Whelan, "The improved grey scale and coarse fine PCM systems: two new digital TV bandwidth reduction techniques," *Proc. IEEE* **54**(3), 376–390 (1966).
3. J. O. Limb and C. B. Rubinstein, "On the design of quantizers for DPCM coders: a functional relationship between visibility, probability, and masking," *IEEE Trans. Commun.* **26**, 573–578 (1978).
4. D. C. Stanford and A. E. Raftery, "Determining the number of colors or gray levels in an image using approximate Bayes factors: the pseudolikelihood information criterion (PLIC)," Technical report no. 390, Dept. Statistics, Univ. Washington (Feb. 2001).
5. R. M. Haralick, K. Shanmugan, and I. Dinstein, "Texture features for image classification," *IEEE Trans. Syst. Man Cybern.* **3**, 510–521 (1973).
6. J. Max, "Quantizing for minimum distortion," *IRE Trans. Inf. Theory* **6**, 7–12 (1960).
7. M. D. Paez and T. H. Glisson, "Minimum mean-square-error quantization in speech PCM and DPCM systems," *IEEE Trans. Commun.* **20**, 225–230 (1972).
8. A. H. Dekker, "Kohonen neural networks for optimal color quantization," *Network Comput. Neural Syst.* **5**, 351–367 (1994).
9. O. Verevka and J. Buchanan, "Local k-means algorithm for color image quantization," *Proc. 1995 Graphics Interface*, pp. 128–135, Quebec City, Canada (1995).
10. X. Wu, "Color quantization by dynamic programming and principal analysis," *ACM Trans. Graphics* **11**(4), 348–372 (1992).
11. F. Fethauer and O. Bryngdahl, "Texture control in image quantization by iterative wavelet transform algorithms," *J. Opt. Soc. Am.* **13**, 12–17 (1996).
12. F. Fethauer and O. Bryngdahl, "Image quantization by a windowed iterative Fourier transform algorithm with local object dependent control," *Opt. Commun.* **138**, 187–192 (1997).
13. T. Zeggel, S. Weissbach, and O. Bryngdahl, "Noise modification in iterative multi-level image quantization," *Opt. Commun.* **100**, 67–71 (1993).
14. C. K. Chow and T. Kaneko, "Automatic boundary detection of the left ventricle from cineangiograms," *Comput. Biomed. Res.* **5**, 388–410 (1972).
15. S. Boukharouba, S. J. M. Rebordao, and P. L. Wendel, "An amplitude segmentation method based on the distribution function of an image," *CVGIP: Image Understand.* **29**, 47–59 (1985).
16. M. I. Sezan, "A peak detection algorithm and its application to histogram-based image data reduction," *CVGIP: Image Understand.* **49**(4), 36–51 (1990).
17. D. Haverkamp, L.-K. Soh, and C. Tsatsoulis, "A comprehensive, automated approach to determining sea ice thickness from SAR data," *IEEE Trans. Geosci. Remote Sens.* **33**(1), 46–57 (1995).
18. L.-K. Soh and C. Tsatsoulis, "Unsupervised segmentation of ERS and RADARSAT sea ice images using multiresolution peak detection and aggregated population equalization," *Int. J. Remote Sens.* **20**, (15 and 16), 3087–3109 (1999).
19. L.-K. Soh, "Automated image segmentation: A data investigation model and SAR sea ice applications," PhD dissertation, Dept. of EECS, Univ. of Kansas, Lawrence, KS (1998).
20. *Multiresolution Image Processing and Analysis*, A. Rosenfeld, Ed., Springer-Verlag, Berlin (1984).
21. F. Bergholm, "Edge focusing," *IEEE Trans. Pattern Anal. Mach. Intell.* **9**(6), 726–741 (1987).
22. M. Antiquazzaman, "Multiresolution Hough transform—an efficient method of detecting patterns in images," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(11), 1090–1095 (1992).
23. F. Goudail, E. Lange, T. Iwamoto, K. Kyoma, and N. Otsu, "Face recognition system using local autocorrelations and multiscale integration," *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(10), 1024–1028 (1996).
24. H. Greenspan, R. Goodman, R. Chellapa, and C. H. Anderson, "Learning texture discrimination rules in a multiresolution system," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(9), 894–901 (1994).
25. R. A. Hummel, B. Kimia, and S. W. Zucker, "Deblurring Gaussian blur," *CVGIP: Image Understand.* **38**, 66–80 (1987).
26. F. Sjöberg and F. Bergholm, "Extraction of diffuse edge focusing," *Pattern Recogn. Lett.* **7**(3), 181–190 (1988).
27. R. Kakarala and A. O. Hero, "On achievable accuracy in edge localization," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(7), 777–781 (1992).
28. Y. Lu and R. C. Jain, "Reasoning about edges in scale space," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(4), 450–468 (1992).
29. S. G. Mallat and S. Zhong, "Characterization of signals for multiscale edges," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(7), 710–732 (1992).
30. P. T. Jackway and M. Deriche, "Scale-space projection of the multi-scale morphological dilation-erosion," *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(1), 38–51 (1996).
31. L. M. Reisell, "Wavelet multiresolution representation of curves and surfaces," *Graph. Models Image Process.* **58**(3), 198–217 (1996).
32. M. H. Yaou and W. T. Chang, "Fast surface interpolation using multiresolution wavelet transform," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(7), 673–688 (1994).
33. *Handbook of Image and Video Processing*, A. Bovik Ed., Academic Press San Diego, CA (2000).

Leen-Kiat Soh received his BS, MS, and PhD degrees in electrical engineering from the University of Kansas, Lawrence, in 1991, 1993, and 1998, respectively. He has been an assistant professor at the Computer Science and Engineering Department at the University of Nebraska since August 2001. Previously, he was a Research Associate at the Information and Telecommunication Technology Center at the University of Kansas. His research interests are in multiagent systems, computer-aided education, intelligent image analysis, and machine learning. He is a member of the ACM, IEEE, AAAI, Phi Kappa Phi, Tau Beta Pi, and Eta Kappa Nu.