Image Compression Technology and Techniques

Tsai-Hong Hong
Marilyn Nashman
Karen Chaconas

Sensory Intelligence Group
Robot Systems Division
National Bureau of Standards
Gaithersburg, MD 20899

April 1988

1. Introduction

Transmission of standard camera images using digital techniques requires a relatively high bandwidth channel. The camera image has 512 horizontal pixels per line, 512 lines per frame and 8 bits of resolution per pixel. Its output rate is 30 frames per second. Therefore, the bit rate for the channel is 60M bits/S. However, the low bandwidth of the transmission channel being considered between a remotely-operated High Mobility Multi-purpose Wheeled Vehicle (HMMWV) and the Remote Command Center (RCC) is about 16 Kbits/S. The compression ratio necessary to accomplish this goal is nearly 4000 : 1. An additional constraint is that the reconstructed images must retain essential information and reasonable fidelity for the mission. An operator must be able to remotely control a vehicle from the RCC, and must be able to pass mobility tests and survival tests using only the reconstructed image. Mobility tests include the ability to make trafficability and movement decisions based on the detection of local obstacles (craters, holes, depressions, etc.), the ability to classify local surfaces (swampy or soft soil, concrete, sand, etc.) , and the ability to determine local surface orientations. Moreover, he must be able to plan local paths and to follow landmarks. Survival tests include the ability to detect enemy threats such as ground vehicles, mines, air vehicles, etc. Weather and lighting conditions may affect the outcome of these tests, and therefore it is important to conduct experiments under varying environmental conditions. The objective of this paper is to report the state of image compression technology, to describe image compression techniques and to select algorithms which can be implemented on parallel image processing hardware for the purpose of remote driving.

Adequate resolution of video data requires vast amounts of data to represent a single image. In order to transmit video images in real-time using a low bandwidth channel, the quantity of this data must be compressed. The maintainance of the quality of the reconstructed images is especially important in considering the low bit-rate coding.

With this in mind, it becomes apparent that there is a need to establish criteria with which the quality of compressed and reconstructed data can be rated. Certain techniques may degrade the resolution of an image while others may decrease the contrast. A measure of the performance of a data compression method must express how closely a reconstucted image correlates to the original. Section 2 discusses the techniques commonly used to measure the fidelity of reconstructed images with both subjective and objective criteria.

Many methods of image data compression and reconstruction have been developed since 1960. The best compression ratio reached, 10:1 occurred in 1983. This class of methods is known as first generation data compression techniques. Section 3 overviews briefly first generation data compression schemes. It is not intended to replace excellent well-known reviews by Pratt[Pratt79], Netravali and Limb[Netra81], or Jain[Jain82], but rather summarizes the major techniques in this class.

Recent progress in the study of the properties of the human visual system has led to a new class of image compression schemes capable of achieving compression ratios as high as 70 : 1. This class of methods is known as second generation data compression. Section 4 describes these techniques. The details of the methods can be found in Kunt et. al. [Kunt85], and Musmann et. al. [Musma85].

During the past few years there has been great interest in the transmission of video images at very low data rates (64k bits/s). The main applications have been in the video telephone conference field [Robin84], [Pear85], [Kanek87], [Gerke87], [Kato87], [Moorh87], [Nam87], [Santa87], [Chen87], [Koga87], [Elnah87], [Chiar87], [Lee87], [Heiman87]. In addition, low data rate transmission is important in the fields of remotel; piloted vehicles (RPV) [Gonz77], and deaf communications[Pear82], [Abram82]. In Section 5 we present some examples of techniques which accomplish this goal.

The focus in Section 6 is oriented to advances in digital coding of real-time applications for low bandwith transmission. Several second generation algorithms are selected for implementation on the PIPE machine as preprocessing for HERDS. PIPE is a high performance parallel processor with an architecture specifically designed for processing video images at up to 60 frames per second. HERDS, developed by the Honeywell Corporation, is an image warping system which can reconstitute video/vehicle state data packets at 30 frames per second. It displays the warped pictures which approximate the current vehicle scene.

2. Overview of Techniques for Evaluating Reconstructed Images

The usefulness of a transmitted image is directly related to how well it approximates the original scene. It becomes important to maintain the integrity of surfaces, texture, and other features visible in the image. In order to determine how well the reconstructed image correlates to the original image, the fidelity must be measured.

Human observers can subjectively rate the quality of a reconstructed image. However, it is less time consuming and easier to generalize if an objective measure can be used. In addition, this objective measure must accurately predict the subjective measure of quality. An example of this is the visual fidelity criterion. Means of measuring fidelity using all approaches are discussed.

2.1 Objective Fidelity Criteria

In some low data rate image transmission schemes, some distortion in the reconstructed image can be tolerated. One type of fidelity criterion is a numerically-valued measure of distortion. Several commonly used objective fidelity criteria [Jain81] [Gonza77] are defined below. The following notation is used in the definitions of these fidelity measurements :

Suppose that the input image consists of an N by N array of pixels f(i,j), where i,j = 0,1,...N-1, and f(i,j) is the gray level value of pixel (i,j). The ouput image consists of an N by N array of pixels g(i,j), i, j = 0,1,...N-1, and g(i,j) is the gray level value of pixel (i,j).

The root-mean-square error (rms) is defined as :

$$e_{rms} = \sqrt{\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[g(i,j)-f(i,j)]^2} \quad .$$

The mean-square signal-to-noise ratio $(SNR)_{ms}$ is defined as :

$$(SNR)_{ms} = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} g^2(i,j)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[g(i,j)-f(i,j)]^2} \quad .$$

The rms value of SNR $(SNR)_{rms}$ is defined as :

$$(SNR)_{rms} = \left[\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} g^2(i,j)}{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[g(i,j)-f(i,j)]^2}\right]^{\frac{1}{2}} \quad .$$

The square root of peak value of g(i,j) squared and the rms noise $(SNR)_p$ is defined as :

$$(SNR)_p = \left[\frac{[peak\ value\ of\ g(i,j)]^2}{e_{rms}}\right]^{\frac{1}{2}} \quad .$$

The average distortion D_{ave} is defined as :

$$D_{ave} = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |g(i,j)-f(i,j)|}{N^2} \quad .$$

2.2 Subjective Fidelity Criteria

When the reconstructed images are viewed by operators, as is the case with the HMMWV, it is more appropriate to use subjective fidelity criteria to evaluate the images. As in Gonzalez[Gonza77] [Manno74], the same amount of rms error may appear to have drastically different visual qualities to a viewer. The subjective quality of an image can be evaluated by showing the image to a group of viewers and averaging their evaluations. Two subjective fidelity criteria have been used in [Frend60] and [Manno74]. The first method[Frend60] uses an absolute scale. Each viewer is asked to evaluate an image based on a scale ranging from Excellent to Unusable. The score of each image is the average value of its evaluations. The second method[Manno74] uses a pair-comparison (bubble sort) method. The viewer repeatedly compares pairs of images and arranges them in order of quality. Thus the best image bubbles to the top ranking and the worst one to the bottom. By averaging the results of many viewers, the first method results in an absolute scale for each image. However, some observers may find it difficult to evaluate images during the course of looking at a large sequence of images. The second method avoids this difficulty but yields only a rank order of images.

2.3 Visual Fidelity Criteria

A visual fidelity measurement is a measure of distortion corresponds to the subjective measurements of viewers. Several visual fidelity measurements have been developed in recent years. Mannos and Sakrison [Manno74] found an objective numerically-valued measure which is in reasonable correspondence with the subjective measure of viewers. The objective measure evaluates a weighted mean-square error of contrast in a image, e.g.,

$$C_{wms} = \sum_{l=0}^{N-1}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1}\sum_{i=0}^{N-1} |e(i,j)\, h(k-i,l-j))|^2$$

where

$$e(i,j) = \Psi(f(i,j)-g(i,j))$$

$$\Psi(x) = x^{\frac{1}{3}}$$

$$h(i,j) = A\,[a+(\frac{\omega}{\omega_0})e^{-[(\frac{\omega}{\omega_0})^{\beta}]}]$$

$$\omega = (i^2+j^2)^{\frac{1}{2}} \qquad A = 2.6 \qquad a = 0.0192$$

$$\omega_0 = \frac{1}{0.114} \qquad \alpha = 1 \qquad \beta = 1.1 \quad .$$

This measure is based on a spatial frequency response. Other measures based on spatial domain characteristics such as visibility can be found in [Netra77][Netra80].

3. First Generation Data Compression Techniques

The need for data compression methods became crucial with the large amounts of data needed for handling the transmision or storage of digital images. The significant problem of how to compress data in order to use available channels or to store it in an efficient manner gave rise to an entire class of compression techniques known as first generation. This class of algorithms is characterized by the fact that they are guided by information theory which does not take the human visual system into account. They reach compression ratios of up to 10 : 1. In this section, five types of first generation techniques are discussed.

3.1 The Coder Method

A simple error-free data compression method is the coder method. The coder assigns each element of its input data to a unique code word. The coder input-output relationship is one-to-one, hence the process is reversible. In order to achieve the compression, the coder must use as few bits as possible. In theory, a lower bound on the average number of bits required to code a set of input data cannot be less than the first order entropy (which is defined below) if successive input elements are coded independently. Compact code algorithms such as the Huffman coder, the B-coder, and the Shift coder can be bounded by the first order entropy[Gonza77]. The concept of the first order entropy is defined as follows :

Suppose we have a set of input grey level values $v_0, v_1, v_2, \ldots, v_{N-1}$, with probabilities p(0), p(1), p(2), ...,p(N-1). Then the first order entropy is defined as :

$$H = -\sum_{i=0}^{N-1} p(i)\times\log_2 p(i) \quad .$$

Thus, the maximum entropy is obtained when the input grey levels have equal probabilities. On the other hand, if p(0) = 1, and p(1) = p(2) = ... = p(N-1) = 0 then the entropy will be zero. This situation results in an image where all pixels have the same grey level. In general, the entropy ranges from 0 to $\log_2 N-1$. As a result, the maximum bound on the compression ratios for the coder methods is $\log_2 N-1$: 1.

3.2 Quantization

From the previous section, the compression ratio for a coder is a function of N, where N is the number of grey levels in an image. In order to achieve higher compression, we may want to decrease N by quantizing the grey levels. A quantizer is a device which reduces output to a lower number of possible grey levels.

Since quantization is irreversible, the distortion due to quantization must be minimized. Several quantizer designs available offer various tradeoffs between simplicity and performance. These include the Lloyd-Max quantizer, the compressor-expander, the optimum uniform quantizer, and the Shannon quantizer. Gish and Pierce [Gish68] and Berger [Berge71] have found that the uniform quantizer is quite close to the optimum based on entropy versus mean-square distortion ($(SNR)_{ms}$) criterion. Combining a Huffman coder and the uniform quantizer may improve the compression ratio of the coder and the quality of the output images. However, the maximum bound on the compression ratios for the combining method is $\log_2 M-1$: 1, where M is the number of grey levels in the quantized images.

3.3 Predictive Compression

The predictive compression methods make use of the property that values of adjacent gray levels are highly correlated for most images. The major predictive method is called the Differential Pulse Code Modulation (DPCM). The DPCM method is used in the following manner. Based on grey level v_{i-1} of pixel i-1 and the correlation between adjacent grey levels, the grey level v_i of pixel i-1 can be predicted. Let $\hat{v}_i$ be the predicted value of v_i; the difference $d_i = v_i - \hat{v}_i$ can be obtained. Assuming that the predictions are reasonably accurate, the magnitude of the difference d_i is usually significantly smaller than the magnitude of grey level v_i. Hence compression is achieved since fewer bits are required to code the differences. The block diagram in Figure 1 describes the DPCM method. A DPCM system contains a predictor, a quantizer, and a coder. The maximum compression ratio achieved by this method is about $\log_2 N$: 1, where N is the number of grey levels in the input images.

Another predictive method is the interpolative method. Most commonly used interpolators are zero-order and first-order interpolators. However higher order polynomials or splines can also be used. The disadvantage of these methods is their computational complexities. The maximum compression ratio for the interpolative methods is the same as those of the DPCM methods.

3.4 Transformation Compression

Transformation compression is shown in Figure 2 and can be divided into three modules. The first module, the transform, reduces the correlation between pixels in an image. By processing the transformed coefficients independently, fewer bits are needed to code the coefficients image than the original input image. The output of the second module, the quantizer, is an image with fewer grey levels. Thus fewer bits are required. The third module is a coder, such as Huffman coder, which may be used to assign a code word to each quantized output.

The general form of the transform module can be illustrated as follows. The input image is subdivided into a number of subimages. Each subimage is coded as a unit. Suppose that each subimage consists of an M by M array of pixels f(i,j), where i,j = 0,1,...M-1, and f(i,j) is the gray level value of pixel (i,j). The tranformed subimage consists of an M by M array of pixels g(k,l), k, l = 0,1,...M-1, such that

$$g(k,l) = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} f(i,j)\, t_{ijkl}$$

The inverse transformation gives each original pixel as a linear combination of the coefficients, i.e.

$$f(k,l) = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} g(i,j)\, T_{ijkl}$$

where the kernels, t and T, depend on the types of transformations used. For example, when using a Discrete Fourier Fransform, t is defined as :

$$t_{ijkl} = e^{[-\iota 2\pi(\frac{ik}{M}+\frac{jl}{M})]} \qquad where \quad \iota = \sqrt{-1} \quad .$$

Other transform kernels used, such as the Karhunen-Loeve Transform, the Hadamard Transform, the Haar Transform, the Discrete Sine Transform, the Discrete Cosine Transform, and the Slant Transform, can be found in [Pratt79], [Netra80] and [Jain81].

The performance of a transform method depends on the transformation itself, the dimension of the transformation, the quantization strategies and the subwindow size. The Hotelling transformaton performs best from both a mean-square error and a subjective quality viewpoint. However, the combination of a fast algorithm and the Discrete Cosine transform is most commonly used. In general, two dimensional transformations yield slightly better compression ratios than one dimensional transformations. Quantization strategies include rounding coefficients and quantizing coefficients. For good quality reconstructed images, approximately half the coefficents are obtained and quantized uniformly. Based on mean-square error measurements, the performance increases as the subwindow size increases. However, the distance between correlated pixels is about 20 pixels in most images. Hence, a subwindow size of 16 or 32 is most commonly used in the literature.

If a subwindow size of n is used, m coefficients are retained, and k bits used to code each coefficient, then there are a total of mk/n bits per pixel. The compression ratio will be $\frac{n}{mk}$: 1 (approximately 10 : 1).

3.5 Hybrid Coding Methods

Hybrid coding refers to methods which combine predictive techniques and transform techniques. Habibi [Habib74] developed a hybrid coding method which combines DPCM and one dimensional transformations such as the Hotelling, the Hadamard, or the Discrete Cosine Transform. The hybrid coding method has the same performance as two-dimensional transform coding methods but has the advantage of being simple to implement.

4. Second Generation Data Compression Techniques

Second generation data compression techniques differ from first generation techniques in two ways. Second generation techniques use the properties of the human visual system such as the spatial impulse response of retinal cells and hierarchical data structures resembling the human nervous system. Secondly, images are segmented into physical entities such as contours and textures.

4.1 Pyramid Coding

The pyramid coding method is a hybrid method which also combines features of predictive and transform methods. Burt and Adelson [Burt83] propose a fast algorithm for this hybrid method. The algorithm first obtains the predicted value for each pixel in the image by convolving a local weighted average, a 5 by 5 Gaussian-like kernel, with the image. The result is a low-pass filtered image which can be represented by fewer samples than the original. The first predicted error image is then the difference between the filtered image and the original image. The predicted error image can be quantized and coded with many fewer bits than the original. Recursively obtaining the next predicted error image by using the same algorithm applied to the sampled low-pass filtered image achieves further data compression. The original image can be reconstructed by adding all of the expanded predicted error images. This algorithm can achieve a compression ratio of 10 : 1. The algorithm has been implemented in the PIPE and will be described in Section 6.3.

4.2 Anisotropic Non-Stationary Predictive Coding

The anisotropic non-stationary predictive coding method [Wilso83] [Knuts83] can be classified as a hybrid method which combines the predictive and transform methods. Lines and edges in an image provide key information. A gray scale image is first transformed into two bias images by measuring these nonstationary linear features. One bias image is a measure of edge magnitude and direction. The other bias image is binary and indicates whether an anisotropic prediction should be used because edge directionality varies rapidly from point to point or whether an isotropic prediction is sufficient.

These two bias images control the anisotropic filter, which is an estimator defined in terms of position (i,j) and angular frequency (ρ,θ). The anisotropic filter is a non-stationary linear combination of three estimations :

$$H(\beta_{ij},\gamma_{ij},\theta_{ij}) = H_i(\rho)+\beta(i,j)[1-\gamma(i,j)]H_a(\rho,\theta,\theta(i,j))+\beta(i,j)\gamma(i,j)H_i'(\rho)$$

where

$H_i(.)$ = stationary isotropic low-pass filter

$H_a(.,.,.)$ = nonstationary anisotropic high-pass filter

$H_i'(.)$ = nonstationary isotropic high-pass filter

$\beta(.,.)$ = edge magnitude

$\theta(.,.)$ = edge direction

$\gamma(.)$ = binary function which indicates whether directional or isotropic estimation should be used.

The predicted image can be obtained by using the estimator described above. Let $\hat{f}(i,j)$ be the predicted gray value in the predicted image :

$$\hat{f}(i,j) = \sum_{k=l}^{M-1} \sum_{l=-n}^{n} H(\beta_{kl},\theta_{kl},\gamma_{kl})\hat{f}(i-k,j-l) \quad .$$

Then, the reconstructed image g(i,j) of size M by M pixels can be written as a function of the predicted image and the difference between the original and the predicted image described above as :

$$g(i,j) = \hat{f}_{ij} + D_{ij}$$

where D_{ij}, prediction error , is the difference between the original and the predicted image for pixel i,j. Quantization of the control parameters β,θ,γ provides for typical compression ratio of 80 : 1 using a variable bit rate encoding scheme. By adding the number of bits used to code the prediction error (D_{ij}), a 35 : 1 compression ratio can be obtained. This technique is optimal for images which contain significant line and edge features. It does not perform as well on images with fine texture or low signal-to-noise ratio.

4.3 Contour and Texture Coding

Contour and texture coding methods first segment the image into textured regions surrounded by contours such that each contour represents one object in the image. Kunt, et. al. [Kunt85] overviews several approaches for contour and texture segmentation algorithms. After the segmentation, the contour and texture in an image are coded separately. Using this method, the compression ratio can be as high as 50 : 1.

4.4 Fractal Compression

A new method for data compression is the fractal compression technique [Barns88]. An image is segmented into object elements such as ferns, leaves, clouds, etc. Each object element in the image is then assigned a corresponding fractal in the fractal database which contains relatively compact sets of numbers, called Iterated Function System (IFS) codes. Hence the object elements can be reconstructed by using their IFS codes. This method can achieve a compression ratio of 10,000 : 1.

5. Time Varying Image Compression Techniques

The transmission of data for remotely piloted vehicles, video conference images, and video telephone images requires data compression of a sequence of images taken over a period of time. Data integrity must be maintained at an acceptable level while achieving the best compression ratio possible. The compression techniques which are discussed in this section include frame-rate-reduction, motion-adaptive frame interpolation and motion compensated prediction. When these methods are combined with transformation methods or quantization, relatively high compression ratios can be attained.

5.1 Remotely Piloted Vehicle

An important application of time varying image compression techniques is the Remotely Piloted Vehicle (RPV). In military surveillance applications, RPVs are more effective than piloted aircraft because they are smaller and cheaper. Each RPV may contain a television camera which transmits images to a pilot in a Remote Control Center (RCC) which is located some distance away. The pilot controls the RPV by remote control signals. Because most RPVs are operated in a hostile (jamming) environment, a significant data compression is required to transmit images over a relatively low-band width (0.5M bits/second) transmitter. A compression ration of 60 : 1 is required.

One solution is a frame rate reduction; i.e. 9 out of every 10 frames are dropped. This is equivalent to transmitting 3 frames per second, the minimum number of frames per second required by the pilot to maintain and control the RPV. This solution achieves a compression ratio of 10 : 1.

The remaining 6 : 1 compression ratio can be achieved by encoding each transmitted image using transformation methods (see Section 3.4) or hybrid methods (see Section 3.5).

5.2 Airborn Television Camera

Lippmann [Lippm80] proposed a motion-adaptive frame interpolation scheme for video transmission of airborn television images at reduced frame rates. He reduces the frame rate at the coder from 25 frames per second to 1 frame per second achieving a bit rate reduction factor of 25. The images between the first and the twenty-fifth frame are approximated by linear interpolation. However, flat ground and the absence of moving objects in the scene are assumed. In addition, two corresponding points have to be measured. For a more accurate interpolation which includes the rotational components of motion, five points of correspondence are needed[Roach80].

5.3 Video Conference and Video Telephone Images

In the cases of video conference and video telephone images, a bit rate of 64-384 Kbits/S has been established as a worldwide standard specification for the transmission of moving images. In order to achieve the required compression ratio of 3000-600 : 1, several researchers [IEEEJ87] have proposed a hybrid coding scheme which uses motion-compensated prediction, followed by a Discrete Cosine Transformation (DCT) and/or quantization. One constraint in this proposed method is that special hardware may be required for real time applications.

6. Preprocessing System for HERDS

The Human Engineered Remote Driving System (HERDS) is a unique system developed by the Defense Systems Group at Honeywell. It facilitates interactive driving of a vehicle from a remote location where the data link between the operator and the vehicle is non-line of sight and low bandwidth. The narrow band radio frequency link that is part of an exisiting communication system has the limitation of not being able to provide full rate video. HERDS overcomes this limitation by transmitting a compressed image frame at 3 seconds per frame along with more frequent vehicle position information. Full rate video images can be provided to an operator at a remote location by extrapolating images between transmitted frames. The dynamics of the true scene are incorporated by using the vehicle position information to "warp" the scene to reflect these changes.

Improvements can be made to the quality of the extrapolated results by decreasing the transmission time latency between image frames. With less latency, the "warped" images deviate less from the actual scene because the information they are derived from is more current. A preprocessing system to HERDS that further compresses image data so that it can be transmitted more frequently than 3 seconds per frame would improve the quality of the information received.

6.1 PIPE machine

Parallel processing is especially applicable to low level image processing. The data structure used at this level is relatively simple; it is the spatially indexed image of points. All parts of the image are treated in the same way and, in general, no effort is made to distinguish between different parts of it. Local operations depend only on corresponding elements or on a neighborhood of elements of the input and output images (Figure 3) Computations tend to be straight forward arithmetic, algebraic or logical operations, and typically a low number of computations per pixel is required. Parallel processors are also suited to multi-resolution representations and processing techniques.

Many local and neighborhood operation data enhancement techniques can be implemented on the Pipelined Image Processing Engine (PIPE) developed at the National Bureau of Standards and manufactured by Aspex, Inc. Some features of PIPE are discussed here, but the reader is referred to [Kent 84][Lumia84] for a more detailed description of the system. PIPE acquires its images in real-time from analog sources such as television cameras, video tapes, and ranging devices, as well as digital data sources. Its output can be directed to television monitors, symbolic mapping devices, and higher level processing systems. All inputs and outputs are synchronous with the video rate of sixty fields per second.

The PIPE system is composed of up to eight identical modular processing stages, each of which contains two image buffers, look-up tables, three arithmetic logic units, and two neighborhood operators (Figure 4). A forward path from one stage to the next allows pipelined and sequential processing. A recursive path from a stage output back to its input allows feedback and relaxation processing. A backward path from one stage to the previous stage allows for temporal operations (Figure 5) . The images in the three paths can be combined in arbitrary ways on each cycle of a PIPE program, and the chosen configuration can change on different cycles. In addition, six video buses allow images to be sent from any stage to any one or more stages.

Images can be processed in any combination of four ways on PIPE: point processing, spatial neighborhood processing, sequence processing or Boolean processing (Figure 6). Different processing can occur at individual pixels in the image by using a region-of-interest operator. All methods can be considered local operations.

Point processing can be either a function of one or two input images and includes simple arithmetic and logical operations such as scaling, thresholding, converting number systems, etc. Look-up tables resident on each PIPE stage allow the user to perform more complex arithmetic operations, trigonometric operations, comparisons, rotations, etc.

PIPE can perform up to two 3 x 3 neighborhood convolutions on each stage in parallel. Both neighborhood operators operate on the same image input, but can perform different neighborhood operations. Larger neighborhood convolutions can be achieved by decomposing an odd-sized neighborhood mask into a sequence of 3 x 3 convolutions. The neighborhood operators can be either arithmetic or Boolean and are performed identically on all locations in the image unless a region-of-interest is specified. Special features are provided to prevent inaccurate computations on the image borders.

Multi-resolution pyramids can be constructed by selecting the "squeeze" or "expand" options as an image is stored or written from a buffer. In the former case, each 2 x 2 neighborhood of the input image is sampled and written to the output image resulting in an image half the resolution of the original. This process can be repeated to generate successively smaller resolution images. Expanding an image involves the opposite operation by pixel replication and generates successively larger resolution images.

Sequential processing works on a set of multiple images, e.g. sequences of images over time, a stereo pair of right and left images, or multi-resolution images. By taking advantage of the inter-stage paths, images can be combined, compared, sampled or differenced to extract the desired application dependent information.

When performing Boolean processing, each pixel of information is considered to be composed of eight independent bit planes, which are operated upon simultaneously. The neighborhood operators can be applied in a Boolean mode, where the output is the combination of the 3 x 3 neighborhood using local operations on each of the eight bit planes.

PIPE programs are written on a host computer using a software package which is an iconic representation of the hardware to generate microcode. Programs are executed by downloading the microcode instructions to PIPE. The software development tool, ASPIPE, allows the user to code the spatial and temporal flow of the data through the hardware to define the look-up tables and PIPE resources to be used. Programs can be edited, saved, compiled, executed, and debugged in this environment. In addition, ASPIPE generates a sequencer file that specifies which micro-operation is executing at each time-cycle. This sequencer also controls branching and looping during microcode execution.

A hardware interface between PIPE and a high level processor (HLP) has been developed and software has been written to support this interface. In this manner, the results of low level vision tasks are transferred to a sequential computer which can perform high level vision tasks of image analysis, recognition, and general decision making which require global information. Since the interface is bi-directional, the HLP can download images or look-up tables directly to any buffer or table on any selected PIPE hardware. In addition, the HLP can select PIPE algorithms by manipulating the PIPE sequencer.

6.2 Image Compression on PIPE

A number of data compression algorithms have been developed and demonstrated on PIPE. These include grey-level quantization, non-maxima suppression, foveal-peripheral simulation, image differencing, histogram slicing, binning, and Laplacian pyramids. The following section briefly describe each of these methods.

Grey scale quantization involves reducing the resolution of each pixel in the image. As represented on PIPE, an image pixel contains 8 bits. However, image resolution and contrast remain acceptable when three or even four low-order bits are dropped. Thus the number of bits required to transmit an image can be reduced by 37.5% or 50% respectively (Figure 7).

Histogram slicing and binning involve operations that are excuted both on PIPE and a high level processor (HLP) since these methods require global knowledge.

Histogram slicing involves transmitting an image from PIPE to the HLP. The HLP computes a histogram of the image and selects the eight most significant peaks (Figure 8). These values are used to create a look-up table consisting of only 3 bits per pixel (0 - 7) which is then downloaded to PIPE. This method can be considered to be a special case of grey scale quantization.

Histogram binning is another grey scale quantization method. A histogram is computed on the HLP and can be segmented either by an integral method in which each bin contains an equal number of pixels or an uniform method where each bin contains an equal number of grey levels (Figure 9). The results of the binning operation can be converted into a look-up table and downloaded to PIPE.

Non-maxima suppression is an image processing method which results in a binary edge image in which all edges are one pixel wide. Figure 10 describes the computations. The compression ratio is very high for this technique since only one bit of information is required to represent each pixel.

Image differencing is an effective compression technique when there is relatively little motion between successive scenes in a sequence of images. In a difference image, only moving objects are visible. Thus, very little information need be transmitted to reconstruct the next image (Figure 11) . In practice, good results have been achieved by transmitting a full image every 8 seconds and difference images every $\frac{1}{30}$th of a second.

The foveal-peripheral simulation is based on the biology of the human vision. In humans, there is a very small area of acute vision, surrounded by areas of degraded vision. Two methods developed demonstrate this phenomenon. In the first (Figure 12), a square window of the image is displayed with full 8-bit resolution. This square is surrounded by concentric windows containing 6 bits, 4 bits, 2 bits, and finally 1 bit of resolution. The full resolution window can be repositioned and resized to meet the users' requirements.

The second foveal-peripheral algorithm utilizes the concept of multi-resolution image processing (Figure 13). In multi-resolution processing, a full sized image is successively sampled and reduced in resolution by a factor of two . Thus a 256 X 256 image is reduced to a 128 X 128 image which is reduced to a 64 X 64 image, etc. Using this technique, an arbitrarily sized and positioned square is displayed at its full resolution. It is surrounded by information obtained from its next level of resolution, (Figure 14) which in turn is surrounded by information at the next level in the pyramid. Each level of the pyramid is successively fuzzier, but requires fewer bits of information.

The last method implemented an PIPE, briefly described in Section 4.1, involves creating a pyramid of multi-resolution images. Each level of the pyramid represents a difference of Gaussian filtered images at two successive levels of resolution. Each level of the pyramid then contains unique frequency bands of information. Figure 15 and Figure 16 summarize the algorithm and the reconstruction of the original image fro the filtered pyramid. Data compression ratios can be increased by using integral binning on each level of pyramid. The theoretical basis for this method is described in

[Burt 83].

7. Conclusion

In order to accurately maneuver a remotely-operated vehicle, visual data must be supplied to the operator as quickly and as accurately as possible. Due to the amount of image information and the delay time in transmitting over a low bandwith channel, data compression techniques must be incorporated. The methods used should provide a compression ratio of 4000 : 1 as well as a reconstructed image that can be used to operate a vehicle under variable lighting conditions and over varied terrain.

We have briefly overviewed major classes of data compression schemes and have selected several algorithms which have been implemented on PIPE. These algorithms can achieve the necessary data compression in real-time, and, when used as a preprocessor for HERDS, can aid in achieving the goal of the remoting driving project.

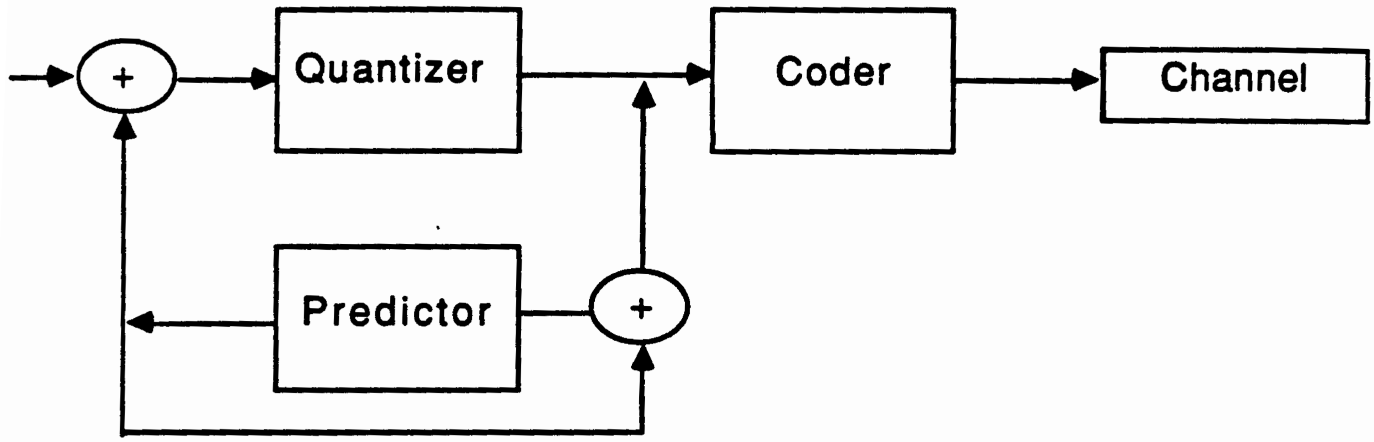


Figure 1. Block diagram of DPCM system

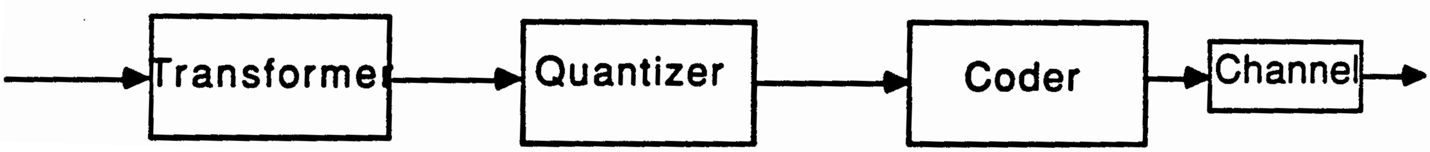


Figure 2. A tranformation model

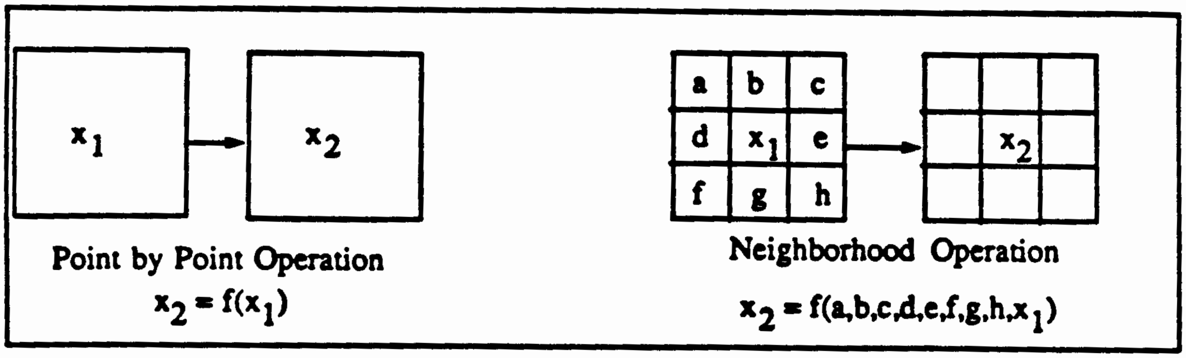


Figure 3. Local operations

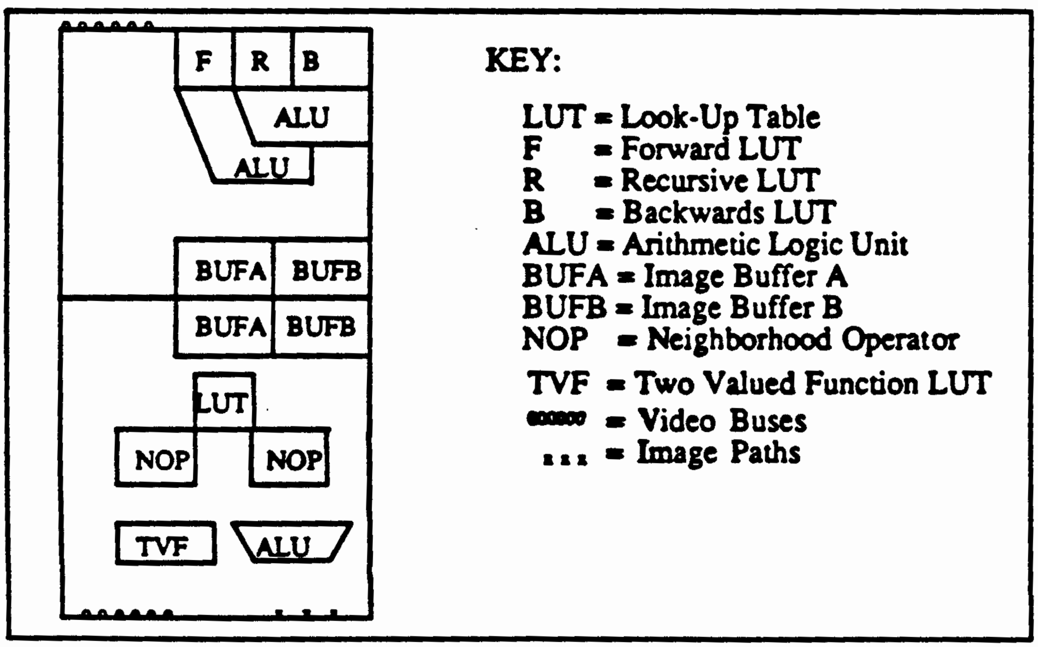


Figure 4. PIPE modular processing stage

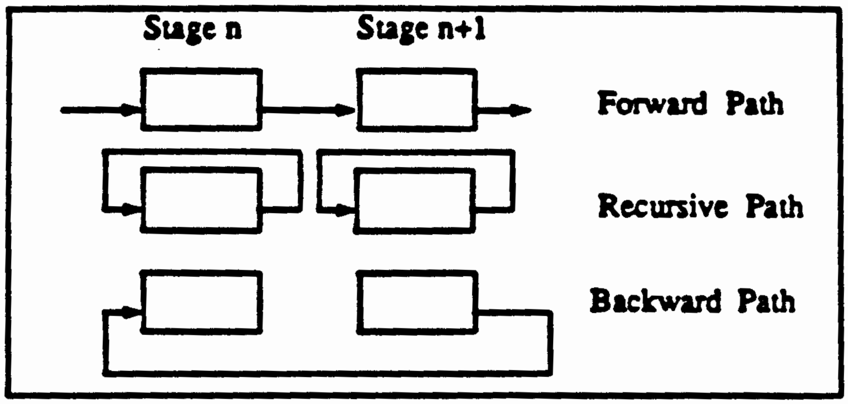


Figure 5. Data flow path between PIPE stages

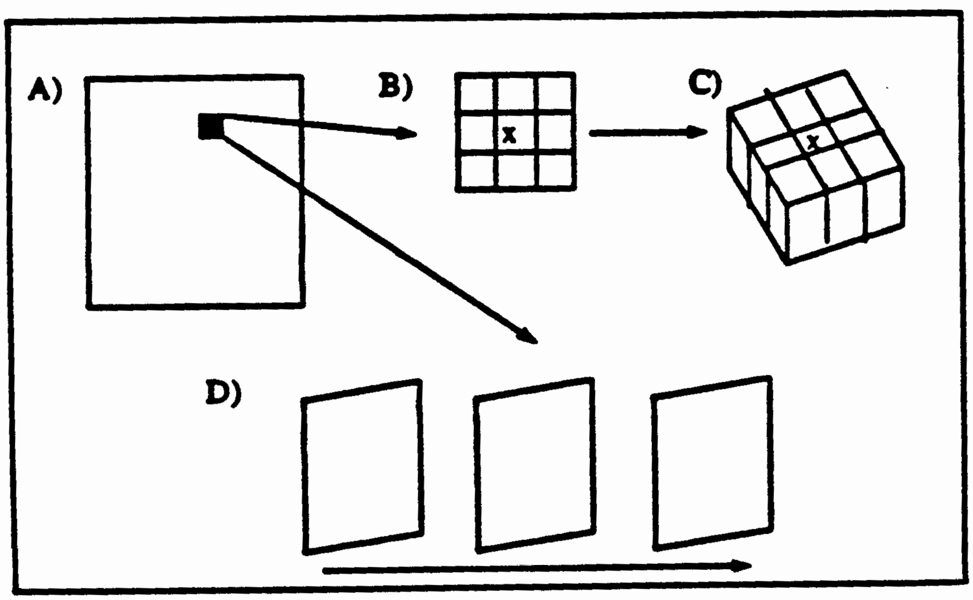


Figure 6. Processing on PIPE: (a) Point (B) Spatial (c) Boolean (D) Sequence

Grey Scale Quantization

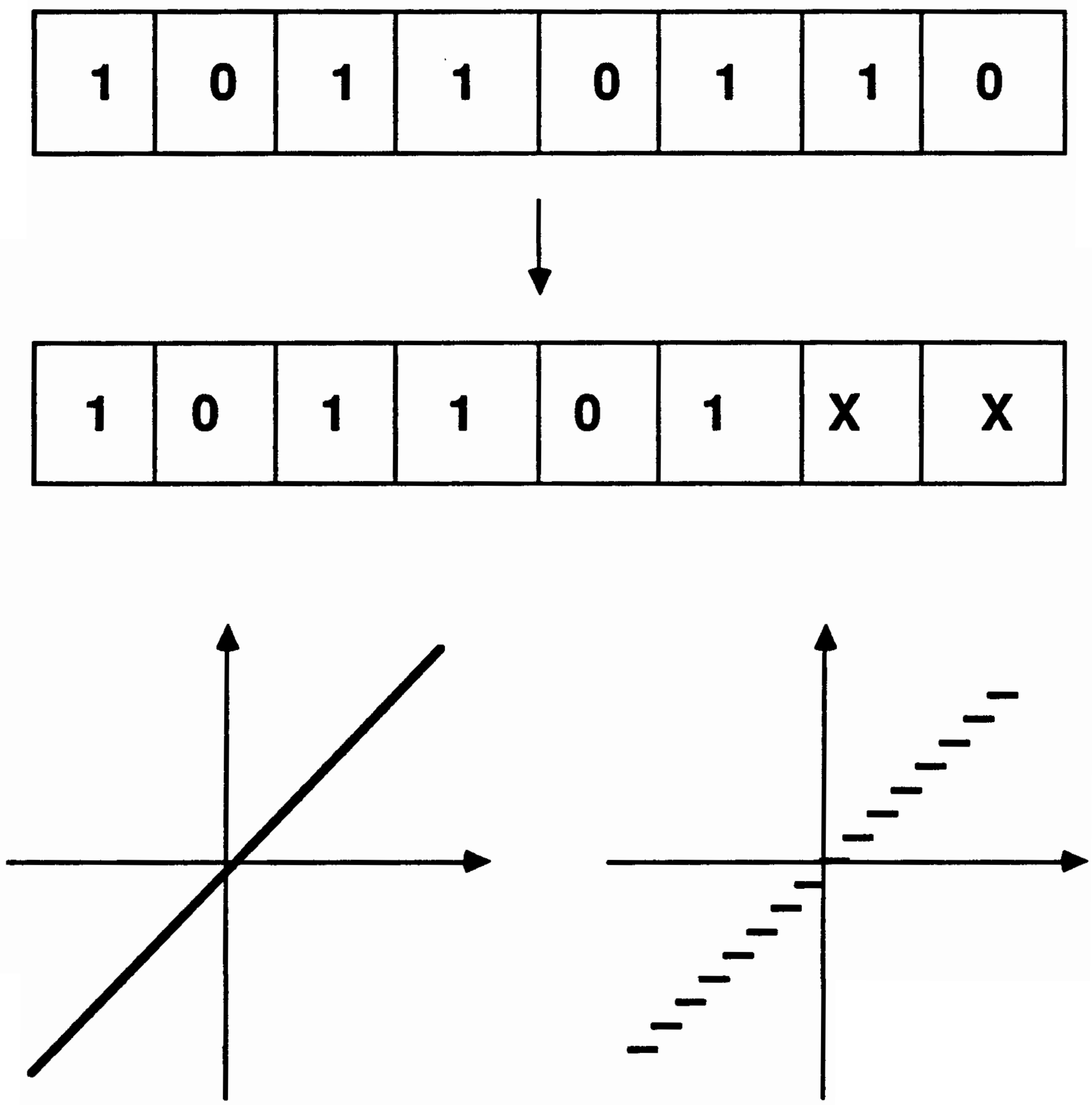
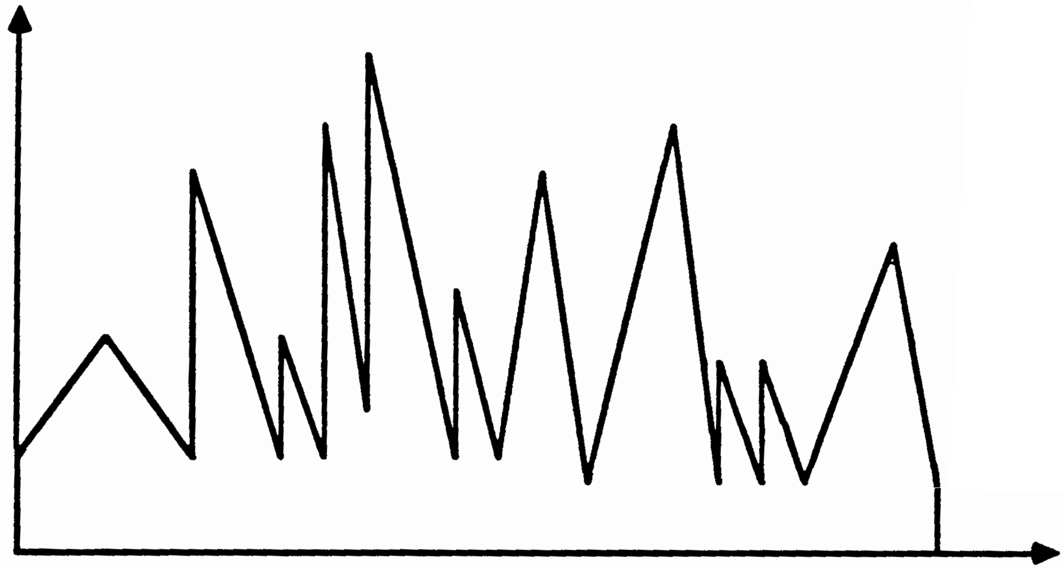


Figure 7

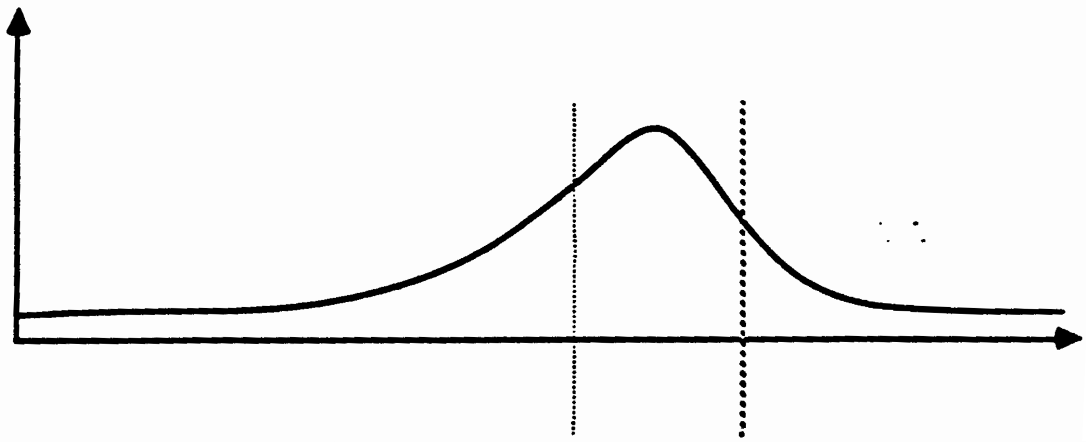
Histogram Slicing



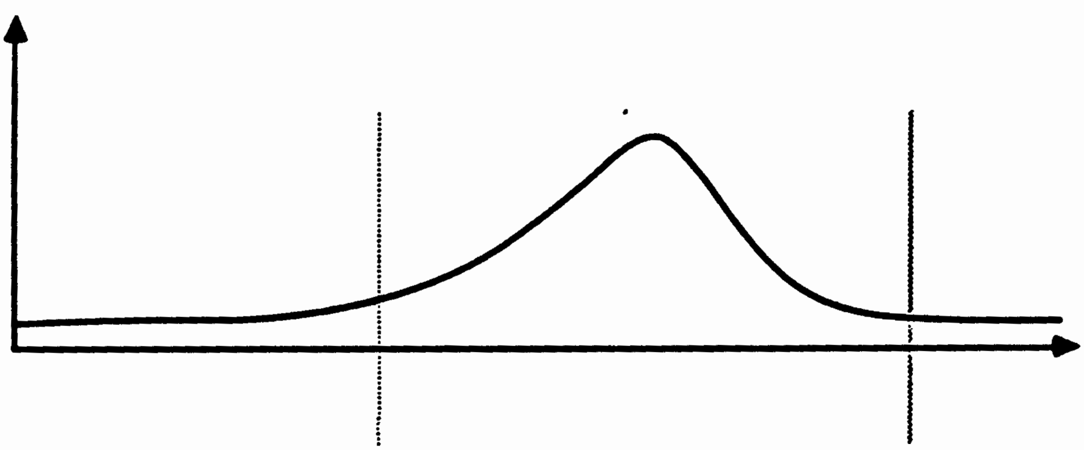
- Compute histogram
- Find 8 peaks in histogram
- Quantize grey levels into 1 of 8 values

Figure 8

Histogram Binning



Integral method-
each bin contains equal number of pixels



Uniform method-
each bin contains an equal number of gray levels

Figure 9

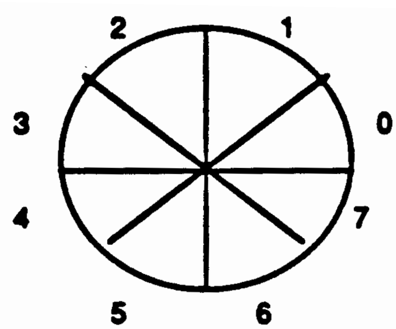
Non-Maxima Suppression

- Apply Sobel Edge Detector

  Compute Edge Magnitude and Direction

  Result is thick edges (3 pixels wide)

- Quantize edges into 8 directions

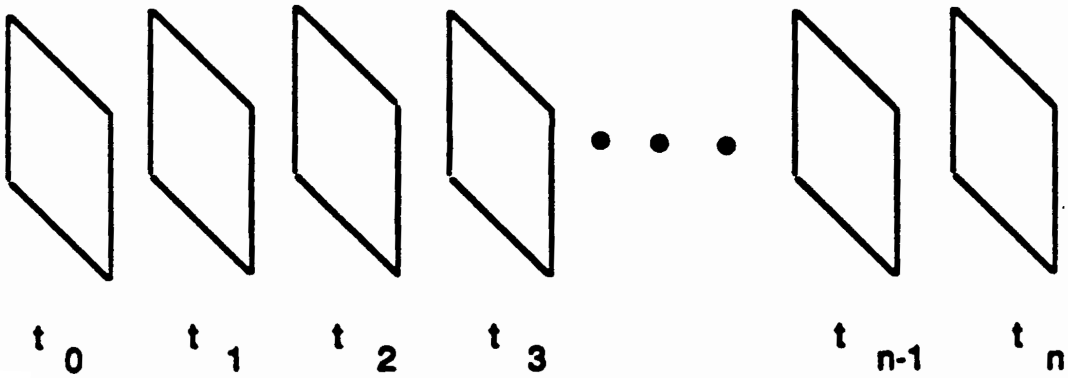


- Eliminate all edge points that are not maximum in the gradient direction

  Result is thin edges (1 pixel wide)

Figure 10

Difference of Images



- Transmit image at t_0

- Transmit difference images

  $t_1 - t_0$

  $t_2 - t_1$

  $\vdots$

  $t_n - t_{n-1}$

- Reconstruct original image

  $$t_0 + (t_1 - t_0) + (t_2 - t_1) + \ldots + (t_n - t_{n-1})$$

  $$\sim t_n$$

Figure 11

Foveal-Peripheral Vision

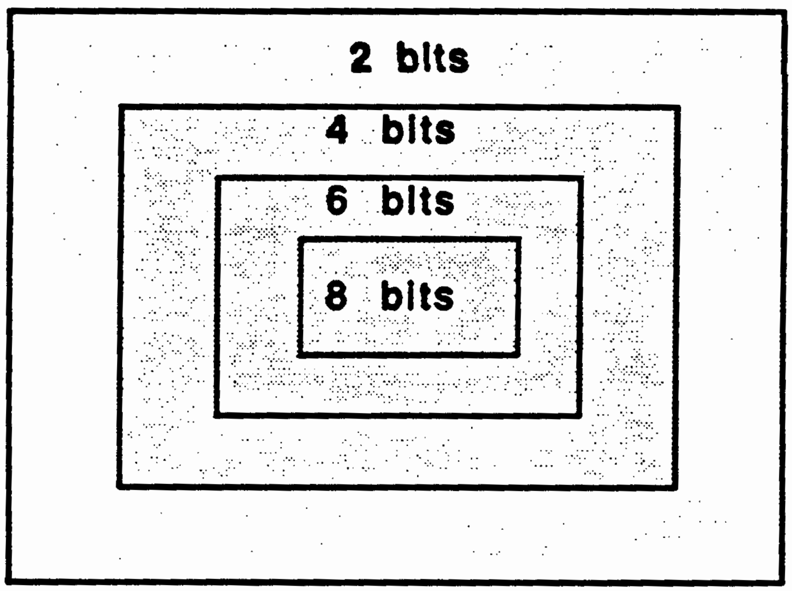


Figure 12

Multi-Resolution Pyramids

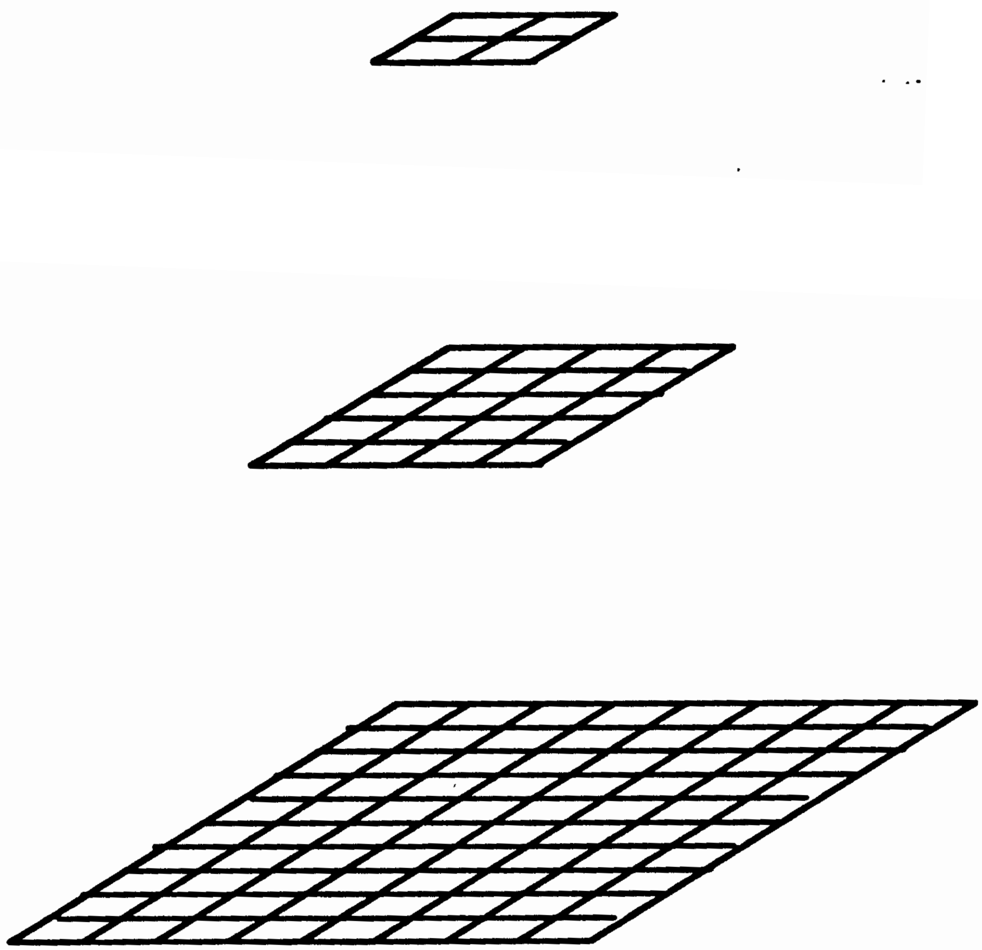


Figure 13

Foveal-Peripheral Vision

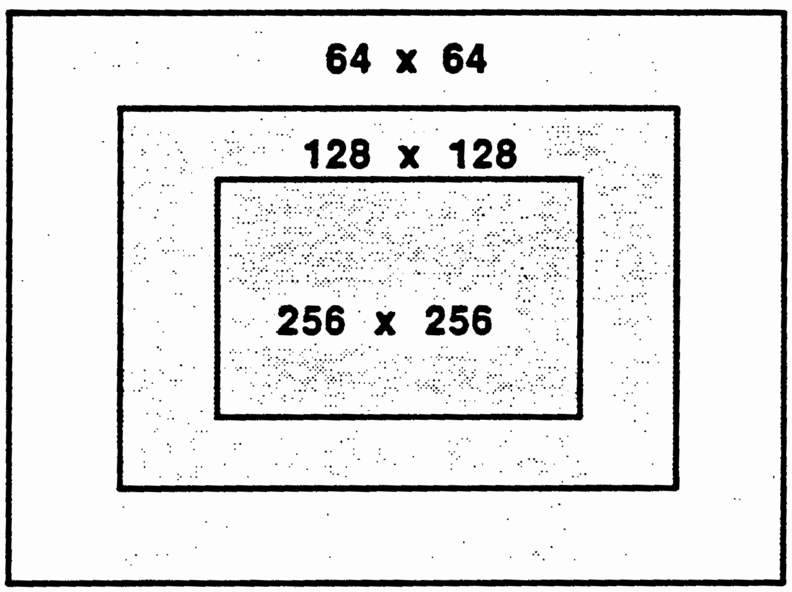


Figure 14

Burt's Laplacian Pyramids

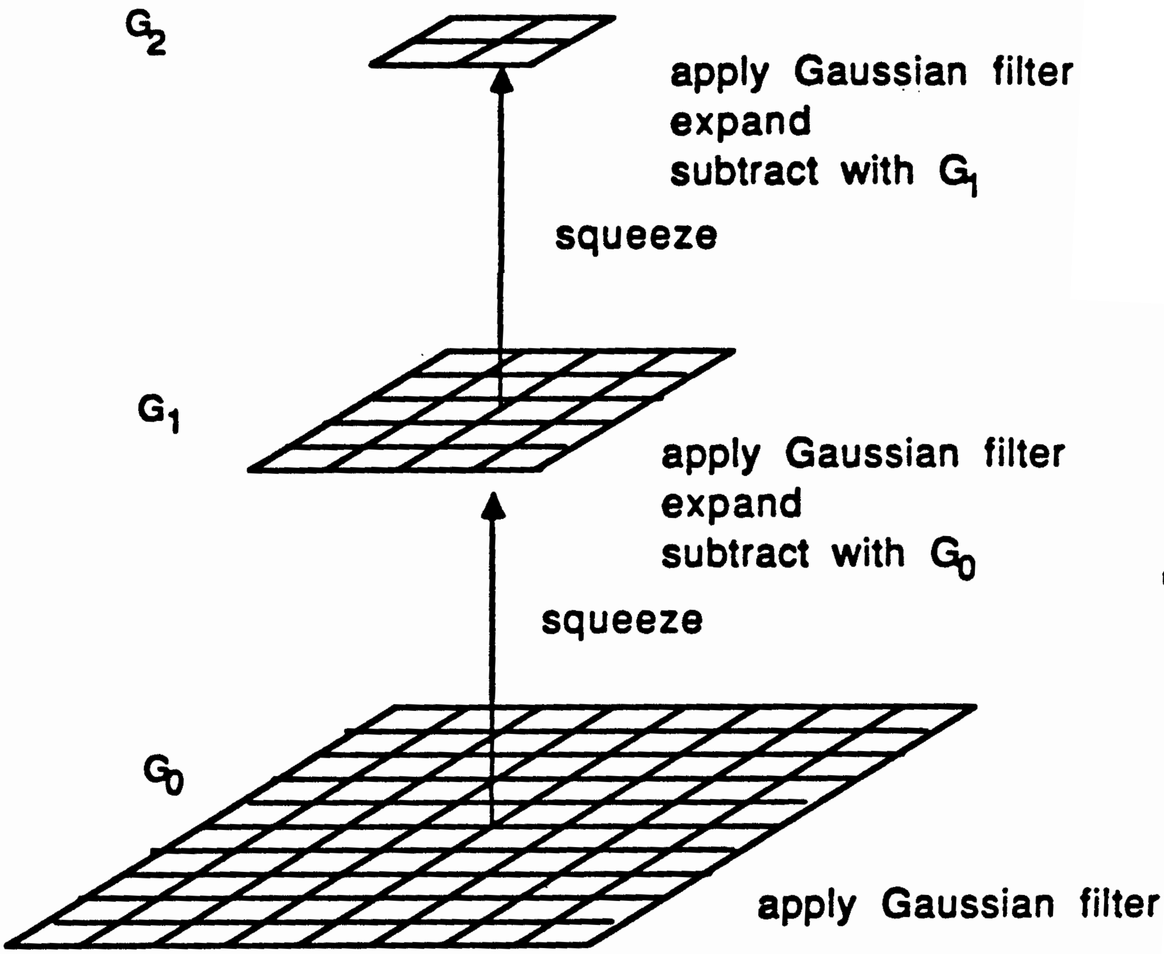


Figure 15

Reconstruction of Burt's Laplacian Pyramids

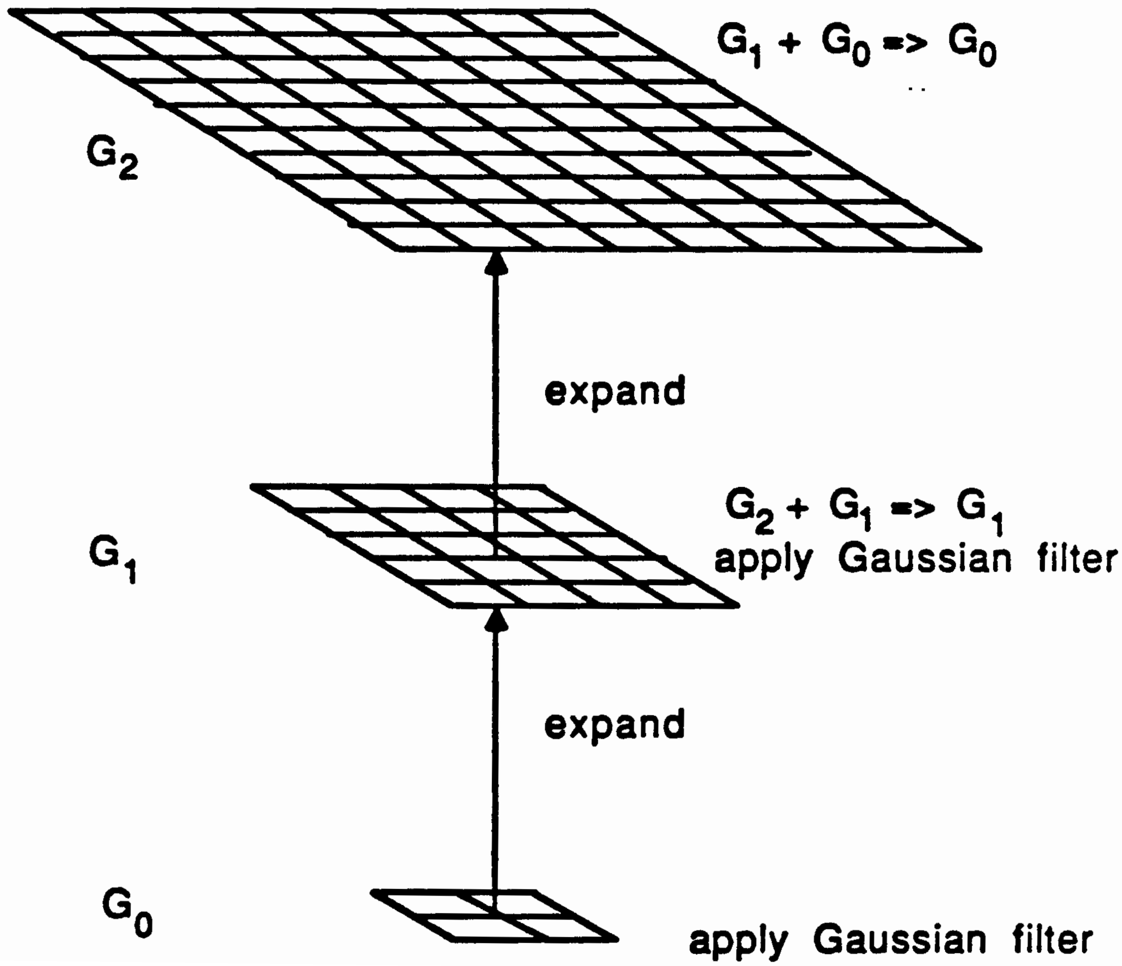


Figure 16

References :

1. [Barns88] M. F. Barnsley and A. D. Sloan, "A better way to compress images," Managing Megabytes, pp. 215-223, 1988.

2. [Berge71] T. Berger, Rate Distortion Theory, Englewood Cliffs, NJ, Prentice-Hall, 1971.

3. [Burt 83] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as compact image code," IEEE Transactions on Communications, vol. COMM-31, pp. 532-540, 1983.

4. [Chen 87] W. H. Chen and D. Hein, "Recursive temporal filtering and frame rate reduction for image coding," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1155-1165, 1987.

5. [Chiar87] L. Chiariglione, S. Fontolan, M. Guglielmo, and F. Tommasi, "A variable resolution video codec for low bit rate applications," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1184-1189, 1987.

6. [Elnah87] S. E. Elnahas and J. G. Dunham, "Entropy coding for low-bit-rate visual telecommunications," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1175-1183, 1987.

7. [Frend60] G. L. Frendendall and W. L. Behrend, "Picture quality - Procedures for evaluating subjective effects of interference," Proceeding of the IRE, vol 48, pp. 1030-1034, 1960.

8. [Gerke87] P. Gerken and H. Schiller, "A low bit-rate image sequence coder combining a progressive DPCM on interleaved rasters with hybrid DCT technique" IEEE Journal on Selected Areas in Communications, vol 5, pp. 1079-1089, 1987.

9. [Girod87] B. Girod, "The efficiency of motion compensating prediction for hybrid coding of video sequences," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1140-1154, 1987.

10. [Gish68] H. Gish and J. N. Pierce, "Asympotitically efficient quantization," IEEE Transactions on Information Theory, vol. 14, pp. 676-681, 1968.

11. [Gonz77] R. C. Gonzalez and P. Wintz, Digital Image Processing, Addison-Wesley Publishing Company, Inc., 1977.

12. [Habib74] A. Habibi, "Hybrid coding of pictorial data," IEEE Transactions on Communications, vol. COMM-22, pp. 614-624, may 1974.

13. [Heima87] A. Heiman and K. Rose, "A look up based universal real time transformer for image coding," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1207-1213, 1987.

14. [IEEEJ87] Special issue on low bit-rate coding of moving images, IEEE Journal on Selected Areas on Communications, vol. 5, 1987.

15. [Jain 81] A. K. Jain, "Image data compression : A review," Proceeding of the IEEE, vol. 69, pp. 349-384, Mar. 1981.

16. [Kanek87] M. Kaneko, Y. Hatori, and A. Koike, "Emprovements of transform coding algorithm for motion-compensated interframe prediction errors," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1068-1078, 1987.

17. [Kato 87] Y. Kato, N. Mukawa, and S. Okubo, "A motion picture coding algorithm using adaptive DCT encoding based on coefficient power distribution classification," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1090-1099, 1987.

18. [Kent 84] E. Kent, M. Shneier, and R. Lumia, "PIPE - Pipelined Image Processing Engine," Journal of Parallel and Distributed Computing, 1984.

19. [Knuts83] H. E. Knutsson, R. Wilson, and G. H. Granlund, "Anisotropic nonstationary image estimate and its applications : Part I - Restoration of noisy images," IEEE Transactions on Communications, vol. COMM-31, pp. 388-397, 1983.

20. [Koga 87] T. Koga and M. Ohta, "Entropy coding for a hybrid scheme with motion compensation in subprimary rate video transmission," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1166-1183, 1987.

21. [Kunt 85] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second generation image coding techniques," Proceeding of the IEEE, vol. 4, pp. 549-574, 1985.

22. [Lee 87] C. Lee and M. Nadler, "Predictive image coding with pseudo Laplacian edge detector," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1190-1206, 1987.

23. [Lippm80] R. Lippmann, "Continuous movement regeneration in low-frame-rate aerial images," Proceeding of the IEEE, International conference on Electronic Image processing, pp. 194-198, 1980.

24. [Lumia84] R. Lumia, M. Shneier, E. Kent, "A real time iconic image processing," NBS-IR, 1984.

25. [Manno74] J. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," IEEE Transactions on Information Theory, vol 20, pp. 525-540, 1974.

26. [Moorh87] R. J. Moorhead II, S. A. Rajala, and L. W. Cook, "Image sequence compression using a pel-recursive motion compensated technique," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1100-1114, 1987.

27. [Musma85] H. G. Musmann, P. Pirsch, and H. Grallert, "Advances in picture coding," Proceeding of the IEEE, vol. 73, pp. 523-549, 1985.

28. [Nam 87] M. K. Nam and W. D. O'Neill, "Adaptive linear predictive coding of time-varying images using multidimensional recursive least squares ladder filters," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1115-1126, 1987.

29. [Netra77] A. N. Netravali and B. Prasada, "Adaptive quantization of picture signals using spatial masking," Proceeding of the IEEE, vol. 65, pp. 536-548, 1977.

30. [Netra80] A. N. Netravali and J. O. Limb, "Picture coding : A review," Proceeding of the IEEE, vol. 68, pp. 366-406, Mar. 1980.

31. [Pear 82] D. E. Pearson and H. Six, "Low-data-rate moving image transmission for deaf communication," in Proceeding of the IEEE Conference on Electronic Image Processing, pp. 204-208, 1982.

32. [Pear 85] D. E. Pearson and J. A. Robinson, "Visual communication at very low data rates," Proceeding of the of IEEE, vol. 73, pp. 795-811, 1985.

33. [Pratt79] W. K. Pratt, Image Transmission Techniques, New York Academic Press, 1979.

34. [Roach80] J. W. Roach and J. K. Agarwal, "Determining the movement of objects from a sequence of images," IEEE Transactions on Pattern Anal. Machine Intell., vol. PAMI-2, no. 6, pp.554-562, 1980.

35. [Robin84] J. A. Robinson and D. E. Pearson, "Visual teleconferencing at telephone data rates," Proceeding of the International Teleconference Symposium, pp. 386-393, 1984.

36. [Santa87] P. Santago and S. A. Rajala, "Using convex set techniques for combined pixel and frequency domain of time-varying images," IEEE Journal on Selected Areas in Communications, vol 5, pp. 1127-1140, 1987.

37. [Wilso83] R. Wilson, H. E. Knutsson, and G. H. Granlund, "Anisotropic nonstationary image estimation and its applications : Part II - Predictive image coding," IEEE Transactions on Communications, vol. COMM-31, pp. 398-404, 1983.