COMMITTEE T1 - TELECOMMUNICATIONS
STANDARDS CONTRIBUTION

DOCUMENT NUMBER: T1A1.5/95-123

DATE: April 21, 1995

STANDARDS PROJECT: VTC Performance

SUBJECT: Addition of Motion Correlated Impairment Artifacts to VIRIS

SOURCE: Bellcore

CONTACT(S): Neil Patel
Bellcore, Room 1F329
331 Newmann Springs Road, Red Bank, NJ 07701
Tel. (908) 758-5362
E-mail: patenr@cc.bellcore.com

ABSTRACT:
This contribution describes a modification to the block distortion impairment and the addition of a signal correlated noise impairment to VIRIS, a Video Reference Impairment System. Methods are described to simulate these two impairments by utilizing scene content and relationships are developed between the VIRIS impairment level and PSNR for two test sequences. We are seeking review and comment from T1A1.5 prior to submitting this proposed US contribution to ITU-T Study Group 12.

DISTRIBUTION:T1A1.5

The following is proposed as a US contribution to ITU-T Study Group 12:

INTERNATIONAL TELECOMMUNICATION UNION DOCUMENT NUMBER

Original: English

TELECOMMUNICATION
STANDARDIZATION SECTOR

STUDY PERIOD 1993 - 1996

Geneva, 6 - 14 September 1995

Question:22/12

STUDY GROUP 12 - CONTRIBUTION

Source[1]: Bellcore

Title: Addition of Motion Correlated Impairment Artifacts to VIRIS

Abstract: This contribution describes a modification to the block distortion impairment and the addition of a signal correlated noise impairment to VIRIS, a Video Reference Impairment System. Methods are described to simulate these two impairments by utilizing scene content and relationships are developed between the VIRIS impairment level and PSNR for two test sequences.

1. Introduction

In 1992 a software package was developed to simulate the impairment artifacts generated by video compression algorithms. This system, called VIRIS, a Video Reference Impairment System, injected block distortion, blurring, edge busyness, jerkiness, and random noise artifacts into a video sequence. VIRIS was intended for use in subjective tests to describe the performance of digital encoders in terms of an objective measure relating impairment level. Compression artifacts were initially simulated independently of the motion and detail in a sequence to simplify the impairment algorithm design. However, after further comparison between VIRIS generated impairments and those produced by compression algorithms, it became apparent that to improve VIRIS utility in subjective tests, selective impairments would need to be correlated to the scene's content. This contribution describes a method to simulate the block distortion artifact and signal correlated noise impairment seen in compressed video.

A long term goal for the development of VIRIS is to contribute information useful for the creation of the draft recommendation P.RISV, a reference impairment system for video. This work supports Question 22 of Working Party 2/12 which addresses audio/video quality in multi-media services.

1. **CONTACT PERSON** Mr. Neil Patel Tel.: 1 908-758-5362
Fax : 1 908-758-4545

The remainder of this contribution is outlined as follows:

Section 2 discusses the theory to correlate discrete compression artifacts to the motion and detail within a scene. A method is described to segment an image into edges and planes and to determine regions of an image in motion.

Section 3 describes an algorithm to correlate VIRIS's block distortion impairment to a scene's motion and detail. Building on this motion correlation technique, an algorithm is also described to simulate signal correlated noise, an impairment appearing as a combination of mosquito and edge busyness impairments. The impairment level of the modified block distortion impairment and signal correlated noise simulation algorithms are characterized in terms of an objective measure, Peak Signal to Noise Ratio (PSNR). Two picture sequences are processed through VIRIS for both impairment types and calculations are performed to relate VIRIS impairment level to PSNR.

Section 4 presents the conclusion.

2. Simulation of Motion Correlated Impairment Artifacts

This contribution describes a set of techniques to simulate two compression impairments, block distortion and signal correlated noise. To optimally simulate the block distortion and signal correlated noise impairments, the appearance of such compression artifacts must be correlated to the spatial information and temporal information of a scene.

Block Distortion Impairment

Block distortion is an impairment characterized by the appearance of the underlying block encoding structure. This impairment is commonly seen in H.261 compressed sequences and MPEG1 compressed scenes. One cause for the visibility of block distortion in compressed video is inadequate bandwidth to code spatial information. During the compression process, block-based transform coding algorithms segment an image into 8x8 pixel blocks prior to transforming the spatial information into the frequency domain. If, during the compression process, there is insufficient bandwidth to code the frequency components, certain blocks could become visible.

Based on viewings of H.261 and MPEG1 encoded sequences, block distortion is visible in smooth areas of an object in motion. An algorithm to simulate block distortion within a reference scene requires positioning the blocks similarly to where they would be seen in compressed video. The current version of VIRIS positions the block distortion impairment randomly throughout the scene. A more accurate simulation of block distortion impairment is possible if the blocks are positioned within smooth areas of a scene in motion. The proposed algorithm (which will be discussed subsequently) positions the blocks within a given video sequence by initially locating smooth areas of an image, determining whether these areas are in motion, and finally, altering the pixel luminance value to produce the block distortion impairment. Under this algorithm, VIRIS is able to more realistically simulate the block distortion impairment as seen in actual encoders, therefore, increasing its utility as both a mechanism to generate subjective test reference conditions and as means to characterize coder performance.

Signal Correlated Noise Impairment

Signal correlated noise is a distortion typically seen around the edges of moving objects. This impairment is commonly characterized by a fine grained, "halo" noise pattern superimposed near and on edges of objects in motion and has a tendency to blur detailed surfaces in motion. This impairment is seen in H.261 compressed sequences and MPEG1 compressed sequences and appears as a combination of edge busyness and mosquito impairments.

2.1 Detection of Edges within a Scene

The block distortion impairment and signal correlated noise impairment simulation share a similar attribute. Prior to the placement of both of these impairment artifacts within a video sequence, edges must be identified. The block distortion impairment is placed in relatively smooth regions while the signal correlated noise impairment is placed near edges of objects. Therefore, prior to injecting the actual impairment artifact, the scene must be segmented into regions of edges and planes (smooth regions).

The basis for the edge detection algorithm is that smooth areas and edges of an image are distinguished by a trend in luminance intensity over successive pixels. Smooth areas are characterized by a relatively consistent regional luminance intensity while edges are identified by adjacent pixels with a large differential luminance.

From vector analysis, the gradient points in the direction of maximum change of *f* at *(x,y)*. The gradient of an image *f(x,y,t),* where *f(x,y)* is the luminance intensity of a frame at time *t*, is described by a differentiation in the horizontal, h, direction and vertical, v, direction

$$\nabla f = \begin{bmatrix} G_h \\ G_v \end{bmatrix} = \begin{bmatrix} \frac{df}{dh} \\ \frac{df}{dv} \end{bmatrix}$$

In the edge detection process, the quantity of interest is the magnitude of this vector

$$\nabla f = \text{mag}\,(\nabla f) = \left[G_h + G_v\right]^{1/2}$$

The gradient indicates the rate of increase of *f(x,y,t)* per unit distance or the slope of the image at pixel *(x,y)*. Depending on the slope of a region, an area can be characterized as an edge or plane. For instance, areas of high slope imply sharp changes in luminance level (i.e. edges) and areas of small slope imply consistent luminance level (i.e. smooth regions).

The gradient calculation, a derivative operation in the horizontal and vertical direction, is modelled as a subtraction of adjacent pixels. But because differentiation simultaneously enhances noise, edge detection is more reliable if an averaging operation is performed along with the differentiation operation. The Sobel operators have the advantage of providing both a differentiating and smoothing effect. From Figure 2-1, the derivatives based on the Sobel operator kernel for pixel coordinate *(x,y)* are

$$G_h = -f(x-1,y+1,t)-2f(x,y+1,t)-f(x+1,y+1,t)+f(x-1,y-1,t)+2f(x,y-1,t)+f(x+1,y-1,t)$$

$$G_v = -f(x-1,y+1,t)-2f(x-1,y,t)-f(x-1,y-1,t)+f(x+1,y-1,t)+2f(x+1,y,t)+f(x+1,y+1,t)$$

where *f* is the zero-padded image at time t.

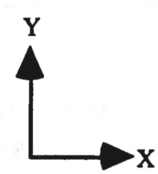
To reduce computational complexity, the gradient can be approximated as

$$\nabla f \cong |G_h| + |G_v| > E$$

The Sobel operators, G_h and G_v, differentiate the image *f(x,y,t)* in the horizontal and vertical directions. The gradient computed at the center of the mask produces the gradient value for that pixel. To obtain the gradient for the next pixel, the mask is translated and the calculation is repeated. Those pixel locations with a gradient sum in the *x* and *y* directions greater than *E* (edges) are considered edges (where *E* is determined by experimentation based on the slope necessary to characterize a spatial detail as an edge). Conversely, those pixel locations with spatial slope less than *E* are considered smooth regions of the image.

-1	-2	-1
0	0 (x,y)	0
1	2	1

Horizontal Edge Detection



-1	0	1
-2	0 (x,y)	2
-1	0	1

Vertical Edge Detection

Figure 2-1 Sobel Filters

2.2 Detection of Temporal Motion

Not only are impairment artifacts correlated to the detail within a scene, but they are also correlated to the motion in a scene. A codec stressed with high motion, high detail scenes, under a fixed data rate must allocate bandwidth between coding motion and detail such that the resultant video is of adequate quality. Unfortunately in making such a trade-off, there is the potential for the visibility of compression artifacts. Block distortion and signal correlated noise are two examples of such motion/detail related compression artifacts. In order to better simulate these impairments, motion detection is necessary. One simple method to detect regions of an image in motion is to perform a pixel-by-pixel subtraction across successive frames,.i.e. at time t and t+1. If across successive frames the difference in luminance value for pixel *f(x,y)* (or change in temporal perceptual information, or TI) is greater than a threshold *C* (where *C* (correlation) is determined by experimentation based on the desired impairment correlation to motion), the object is in motion.

$$TI(x,y,t+1)=f(x,y,t+1)-f(x,y,t) > C$$

Such is not a robust method for identifying regions in motion, as changes in lighting or camera panning can also imply motion. However, its simplicity, a desirable attribute for VIRIS algorithms, as compared to techniques such as object segmentation, justifies its use.

3. Simulation of Impairment Artifacts

The edge and motion detection schemes form the foundation for accurately simulating the block distortion and signal correlated noise impairments. In the case of the block distortion, once edges of an image are identified, smooth areas are inherently known. These smooth areas are tested for motion and classified as possible candidates for the addition of block distortion. In the case of the signal correlated noise impairment, instead of adding impairments to smooth areas in motion, noise is added to edges in motion.

The specific algorithm to simulate the two impairments is described in **Sections 3.1** and **3.2**. Along with a specific implementation in VIRIS, the relationships between PSNR and impairment level are provided for two sequences impaired by VIRIS.

3.1 Block Distortion Impairment Simulation

The procedure for placing block distortion in an scene is to first locate smooth regions of an image and then to determine whether such areas are in motion. Using the Sobel filtering techniques provided in **Section 2.0**, the smooth areas of an image in motion can be identified. These locations are prioritized in terms of the most probable location where block distortion would occur.

As block distortion is the visibility of 8x8 pixel blocks, the impairment prioritizing process begins by subdividing the entire image into 8x8 blocks. For a SIF image (Source Intermediate Format) such an operation results in a total of 1320 blocks. The SIF image is segmented into 8x8 blocks by grouping the spatial pixel positions into block locations (indexed by (r,s)) through the following assignment:

$$b(r,s) = b(\text{truncate}((x-1)/8)+1, \text{truncate}((y-1)/8)+1) \text{ for all } x \text{ and } y$$
$$x = 1 \text{ to } 352 \quad y = 1 \text{ to } 240 \text{ (pixel range)}$$
$$r = 1 \text{ to } 44 \quad s = 1 \text{ to } 30 \text{ (block range)}$$

where $b(r,s)$ identifies a specific block enclosing a set of 64 pixels of the image. Those (x,y) which map into the same $b(r,s)$ are pixels enclosed by the block.

The specific blocks to impair are selected based on satisfying two criteria:

- This block must enclose no more than a fixed number of pixels identified as edges in the present and previous frame (this condition is imposed as it is otherwise very difficult to locate a block containing absolutely no edge detail).

- The block must contain the largest temporal perceptual information of all blocks within the image. In the temporal information calculation, pixel positions identified as edges in the past and present frame are excluded from the summation.

$$\sum_{i=-7}^{0} \sum_{j=-7}^{0} \mathrm{abs}(TI(8r+i, 8s+j, t+1)) = \text{maximum}$$

where if *(8r+i,8s+j)* is an edge at t or t+1, TI(.)=0, and r=1 to 44, s=1 to 30, the subsampled block position.

The algorithm iterates over the entire image by locating the 8x8 block containing the set of pixels with the greatest change in luminance value and fewer than *E* edge pixels. Excluding already impaired blocks of the same frame, this process is repeated until the desired number of blocks is impaired. The set $\{(r,s)\}$, the block positions impaired in the present frame, is retained and applied as impairment locations for the next 14 frames. At the 15th frame, new block distortion positions are recalculated and assigned into $\{(r,s)\}$. Block distortion is injected into the present frame, and again the position of the blocks specified in $\{(r,s)\}$ are used to impair the next 14 frames. This process is repeated for all frames in the sequence.

Block position is fixed at multiples of 15 frames to allow adequate time for the eye to perceive the impairment. If a higher or lower block correlation to motion is desired, this parameter can be adjusted appropriately.

The selected block is actually impaired by altering the luminance value of its 64 pixels. Each pixel value within that block is assigned a luminance level based on the average luminance of the pixels in the undistorted block, the original luminance value of the pixel, and a random number between -2 to 2. Mathematically, block distortion is added to a block *(r,s)* by altering the luminance value of all enclosed pixels:

$$\frac{\sum_{i=-7}^{0} \sum_{j=-7}^{0} f(r \cdot 8 + i, s \cdot 8 + j)}{2} + \frac{f(x, y)}{2} + \mathrm{random}(-2, 2) = \tilde{f}(x, y)$$

for all *(x,y)* in *b(r,s)* such that:

$$r \cdot 8 - 7 \le x \le r \cdot 8$$
$$s \cdot 8 - 7 \le y \le s \cdot 8$$

VIRIS Implementation of Block Distortion Impairment

Combining the block placement and impairment techniques, VIRIS was modified to better simulate the block distortion impairment. Similar to the original VIRIS, a level parameter is established to indicate the number of blocks to impair. This impairment level is based on the fraction of the 1320 blocks of a SIF image. For instance, a block distortion level of 100 implies that a maximum of 100*1.320 or 132 blocks within the image will be impaired based on scene content. The specific blocks to impair are determined through the conditions described in **Section 3.1**.

The threshold for edge detection was determined experimentally to be E=500. Gradients less than this threshold are considered smooth regions and are candidates for the positioning of the block distortion impairment. The maximum number of edge pixels within an 8x8 block for a region to be considered a block distortion candidate has been experimentally determined to be M=5.

Two sequences were run under the modified block distortion impairment algorithm. Football is a sequence with a high degree of motion and detail (football players in a game) while Salesman is a sequence with less detail but more surface area in motion (speaker holding object in hand). The PSNR ratio was used in the simulations as a means to objectively characterize the level of impairment.

$$\mathrm{Nrms}_k = \sqrt{\sum_{i=1}^{i=N}\sum_{j=1}^{j=M}\frac{(u(i,j,t)-p(i,j,t))^2}{(N\cdot M)}}$$

Where:

Nrms_k = average rms noise for frame k
$u(i, j, t)$ = luminance value of unimpaired pel (reference image) at row i, column j at time t
$p(i, j, t)$ = luminance value of impaired pel (impaired image) at row i, column j at time t
and for a SIF image, N=240, M=352.

The PSNR across all frames in a sequence is calculated by first determining the average Nrms across all frames in the sequence and then converting that quantity into a noise ratio.

$$\mathrm{Nrms} = \frac{1}{F}\cdot\sum_{i=1}^{F}\mathrm{Nrms}_k$$

Where:
F = number of frames over which to calculate PSNR
Nrms = average rms noise over frames of interest

$$\mathrm{PSNR} = 20\log\frac{255}{\mathrm{Nrms}}$$

Figure 3-1 and 3-2 illustrate the relationship between PSNR and VIRIS input for the Football and Salesman sequences, respectively. The impairment levels range from 5 to 300, which corresponds to a maximum of 6 to 396 impaired blocks. As the impairment level is increases, the PSNR and resulting visual quality of the sequence decrease. In the Salesman sequence, the block positioning alternates between the speaker's face and the hand held object, depending on which has more motion. The blocks in the Football sequence are centralized on the players.

3.2 Signal Correlated Noise Simulation

In order for signal correlated noise to be simulated at a pixel location, the pixel must have the fol-

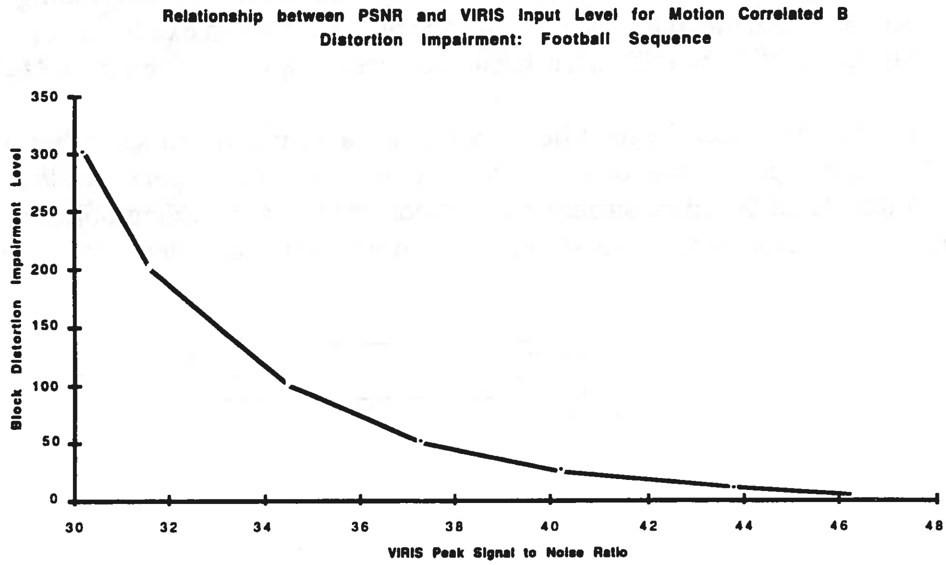


Figure 3-1 Football Sequence - Block Distortion

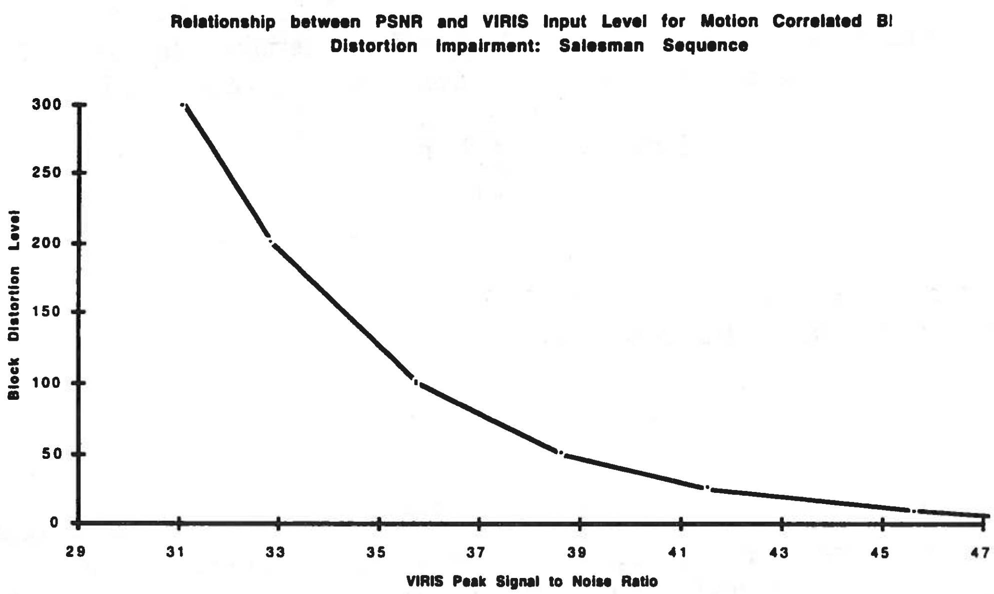


Figure 3-2 Salesman Sequence - Block Distortion

lowing characteristics:

- The pixel location $f(x,y)$ must have a gradient greater than E, indicating an edge.
- The pixel location must be in motion, as specified by a change in luminance intensity over successive frames.

Once these locations are identified within an image, signal correlated noise is added by selectively varying the pixel luminance level of an unimpaired image. This is accomplished by randomly altering the luminance level of the pixels identified as edges in motion. The noise impairment is simulated by randomly selecting an integer luminance level between $-\beta$ and $+\beta$ and adding that quantity to the luminance level of the unimpaired image:

$$\tilde{f}(x, y) = f(x, y) + \mathrm{random}(\pm\beta)$$

For each pixel location satisfying the conditions bulleted above, the luminance value of the unimpaired image is altered by an absolute amount β. The range of β controls the intensity of the impairment. Increasing β increases the visibility of the signal correlated noise impairment. To better correlate the impairment to the temporal variation, the pixels to impair in the present frame are the union of pixels identified as edges in motion in the previous and present frame.

VIRIS Implementation of Signal Correlated Noise Impairment

VIRIS was also modified to simulate signal correlated noise. Within VIRIS, the noise impairment level is described by the range of variation of the unimpaired image pixel intensity. For instance a level of $\beta = 10$ indicates that the luminance level of those pixel to impair can vary by at most 10. Furthermore, experimentation and informal laboratory viewing has shown that the pixel locations $f(x,y)$ with a gradient magnitude, ∇f, greater than E=50 and a temporal change in perceptual information between successive frames greater than C=2 achieves optimum placement and simulation of the signal correlated noise.

Again, the two sequences, Football and Salesman, were run though VIRIS with varying impairment levels. Figures 3-3 and 3-4 illustrate the relationship between PSNR and VIRIS impairment level for each of the two sequences. As the impairment level increases from a level of β=2 to β=25, the PSNR decreases indicating a reduction in image quality in terms of the impairment level. In the Salesman sequence the signal correlated noise is concentrated on the speaker's face and the object he is holding. In the Football sequence the noise is concentrated on the edges outlining the players.

4. Conclusion

This contribution describes a method for simulating impairments commonly seen in images produced by video compression algorithms. Specifically, algorithms are provided to simulate the block distortion and signal correlated noise impairments by accounting for the spatial and temporal detail during impairment placement. The original block distortion impairment algorithm in VIRIS has been modified to more realistically correlate the block distortion impairment to the content of a scene. An algorithm is also proposed to simulate an additional impairment commonly seen in encoded sequences, signal correlated noise. Further studies for these two impairments will require performing subjective tests to develop relationships between mean opinion score and impairment level.

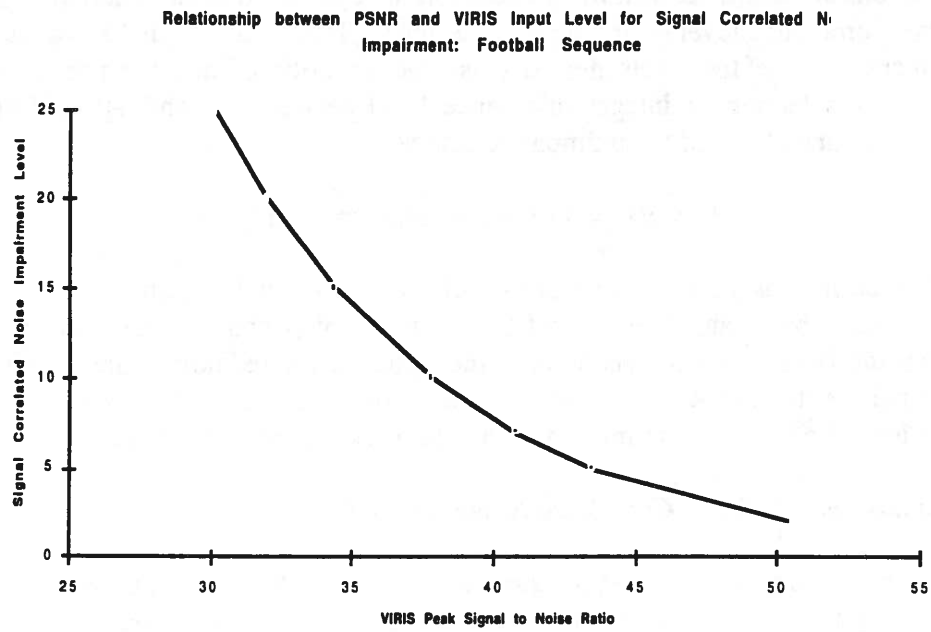


Figure 3-3 Football Sequence - Signal Correlated Noise

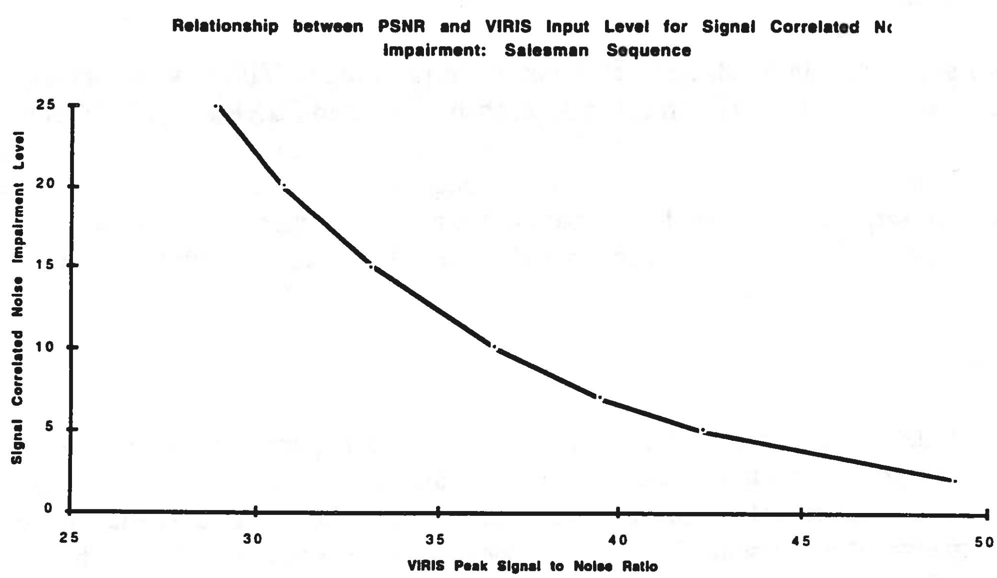


Figure 3-4 Salesman Sequence - Signal Correlated Noise

DRAFT

FIGURE 7
BLURRING PSNR VS MOS

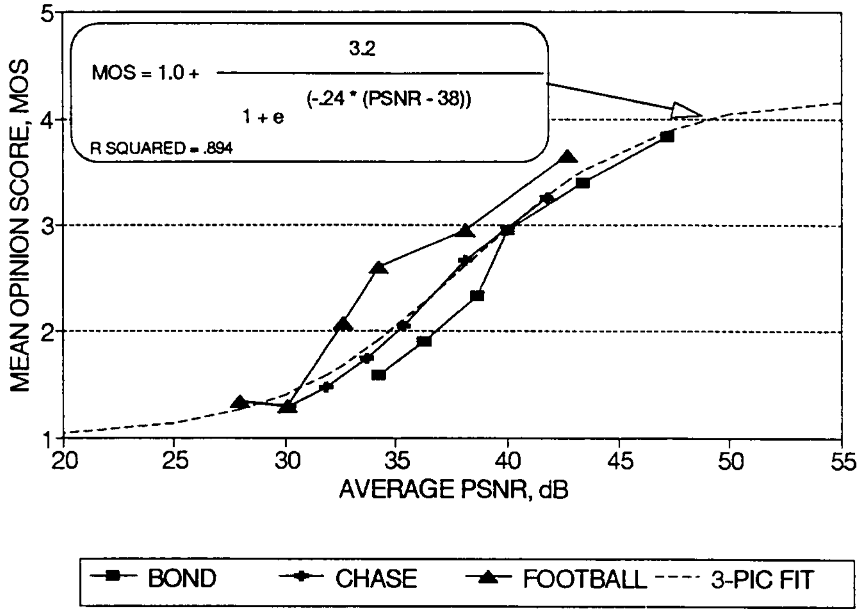


FIGURE 8
MOSQUITOES PSNR VS MOS