

A Two Channel Spatio-Temporal Encoder

by

Lawrence N. Claman

Submitted to the Department of
Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Science in Electrical Engineering

at the

Massachusetts Institute of Technology

May 1988

© Massachusetts Institute of Technology, 1988

Signature of Author _____
Department of Electrical Engineering and Computer Science
16 May 1988

Certified by _____
Walter Bender
Principal Research Scientist, Media Laboratory
Thesis Supervisor

Accepted by _____
Leonard A. Gould
Chairman, Department Committee on Undergraduate Theses

RECEIVED

JUL 28 1988

LIBRARIES

ARCHIVE

A Two-Channel Spatio-Temporal Encoder

by

Lawrence N. Claman

Submitted to the
Department of Electrical Engineering and Computer Science
on 16 May 1988 in partial fulfillment of the requirements
for the degree of Bachelor of Science

Abstract

A new technique of interpolating spatial and temporal data using methods of two-channel encoding is discussed. This work is based on the assumption that for many applications it is possible to obtain two channels of data: one with high temporal resolution but low spatial resolution, and the other with low temporal resolution but high spatial resolution. Techniques for combining these two channels to create an output with both high spatial and temporal resolution are investigated. This included using methods of spectral analysis and statistical interpolation.

Thesis Supervisor: Walter Bender
Title: Principal Research Scientist, Media Laboratory

The work reported herein was funded in part by a contract from IBM.

Contents

1	Introduction	9
1.1	Background	10
2	Two-Channel Encoding by Frequency Separation	16
2.1	Overview	16
2.2	Process Used	18
2.3	Results	20
2.3.1	Qualitative Results	20
2.3.2	Signal to Noise Ratio Calculations	24
2.3.3	Animation Effects	26
2.4	Conclusions	28
3	Temporal Interpolation Based on Statistical Coding	30
3.1	Background	31
3.2	Description of Algorithm	33
3.3	Results for Creating a Single Frame	35
3.3.1	Qualitative Results	36
3.3.2	Animation Effects	39
3.3.3	Signal to Noise Ratio Calculations	39
3.4	Interpolating Two Frames	40
3.4.1	Procedure	41
3.4.2	Results	41
3.5	Conclusions	44
4	Spatial Interpolation by Statistical Coding	46
4.1	Background	47
4.2	Description of Algorithm	50

4.3	Results	52
4.4	Conclusions about Spatial Interpolation	53
5	Conclusions	55
5.1	Summary	55
5.1.1	Methods of Choice	56
5.2	Future Investigations	56
5.2.1	Determination of Maximum Interpolation Range . .	56
5.2.2	Image Flow and Motion Compensation	57

List of Figures

1.1	High Frequency Temporal, Low Spatial	11
1.2	Low Frequency Temporal, High Spatial	12
1.3	High Spatial, High Temporal Frequencies	13
2.1	Illustration of HTLS and LTHS channels	18
2.2	Approximate Error per Frame	19
2.3	Block Diagram of Frequency Separation	21
2.4	HTLS sequence created with $.5\pi$ cutoff filter	22
2.5	HTLS sequence created with $.25\pi$ cutoff filter	22
2.6	Results using half frequency HTLS channel	23
2.7	Results using quarter frequency HTLS channel	24
2.8	High Spatial Frequencies only Added Forward	29
2.9	High Spatial Frequencies Added Forward & Reverse	29
2.10	Error Function for Adding Highs both Forward & Reverse	29
3.1	Illustration of a Particular Vector	32
3.2	Flow of Temporal Interpolation Algorithm	34
3.3	Half Frequency Channel Results, Middle Element Averaged	37
3.4	Half Frequency Channel Results, Middle Element from HTLS Channel	37
3.5	Quarter Frequency Channel Results, Middle Element Averaged	38
3.6	Quarter Frequency Channel Results, Middle Elements from HTLS Channel	38
3.7	Duplicating Entries in Four Element Vector	42
3.8	Two Interpolated Frames using Half Frequency HTLS Channel, Averaging to produce Middle Search Elements	43

4.1	Correspondence between HTLS and LTHS pixels	48
4.2	Corner points from HTLS channel	49
4.3	Closeup of HTLS frame upsampled by statistical interpolation	52

List of Tables

2.1	SNR for unprocessed HTLS Channel	25
2.2	SNR of Reconstructed Sequence using Frequency Separation	26
3.1	SNR for Interpolating a Single Frame	39
3.2	Half Frequency HTLS Channel	43
3.3	Quarter Frequency HTLS Channel	44
4.1	SNR for Spatially Interpolated Images	53

Acknowledgments

There are many people I would like to thank for their continuing support. The entire Electronic Publishing group deserves credit for the countless tidbits of advice I picked up from everyone. In addition, certain people deserve mention. First, I would like to thank Walter Bender for giving me the idea as well as direction. I'd also like to thank Pascal Chesnais for his harassment, as well as **always** being at the lab to help out. The Brocks get credit for allowing me to blow off time whenever I got frustrated. Finally, I would like to thank my family, without whom I'd have never gotten where I am.

Chapter 1

Introduction

Without a doubt, one of the most important areas of image processing research is that of picture quality improvement. One only has to look at the multitude of high definition television (HDTV) systems being developed to see that an important objective is to be able to enhance an image with low resolution. This aim applies not only to increasing the spatial resolution (i.e. the detail) of a picture but also the temporal resolution (i.e. frame rate) of an image sequence. Together these two parameters are called the *spatio-temporal* resolution of an image sequence. Research has been conducted by the author on a method of creating a high spatial and temporal resolution sequence from two input sequences of different resolutions; one

of which has high spatial resolution but low temporal resolution, and the other which has low spatial but high temporal resolution. The result of this combination will be a new sequence with an overall higher resolution.

This paper will describe three methods for combining two channels. These are frequency separation, temporal interpolation by statistical coding, and spatial interpolation by statistical coding. The results will be analyzed both by their subjective appearance and by the calculation of their signal to noise ratio.

1.1 Background

There are many applications in which it will be useful to combine two channels of different resolutions. Most systems will have some upper limitation on their total resolution. This can be due to bandwidth limitations in a transmission channel (such as with television), or it can be due to actual physical limitations (such as a film's negative size). For all of these cases some sort of compromise must be made between having good spatial resolution versus having good temporal resolution.

This can be shown pictorially by looking at 2-dimensional frequency

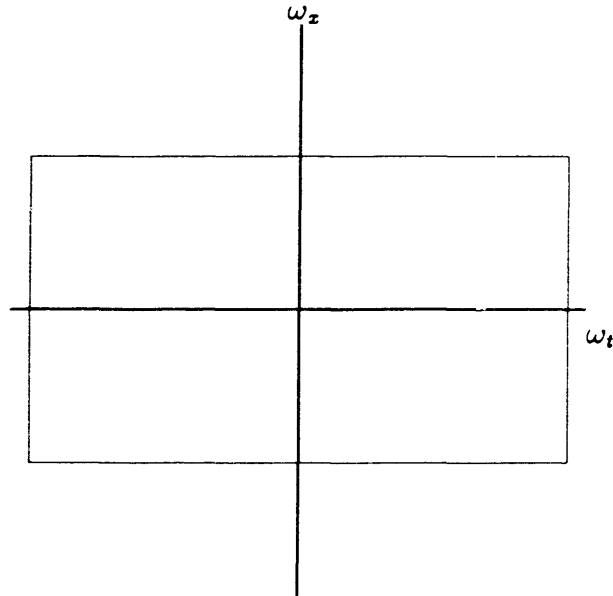


Figure 1.1: High Frequency Temporal, Low Spatial

space, which shows the parameters ω_x and ω_t ,¹ where ω_x represents the spatial frequencies and ω_t represents the temporal frequencies (see figures 1.1 and 1.2). In each figure, the area of the box, which represents the total resolution, is the same. However, the orientation of the boxes is different, which shows the difference between each channel's spatial and temporal resolution. Figure 1.1 would capture motion well, while figure 1.2 would capture detail well.

Research has been done by the author on a method of combining a high

¹Images are actually described by three-dimensional frequency space, with the parameters ω_x , ω_y , ω_t

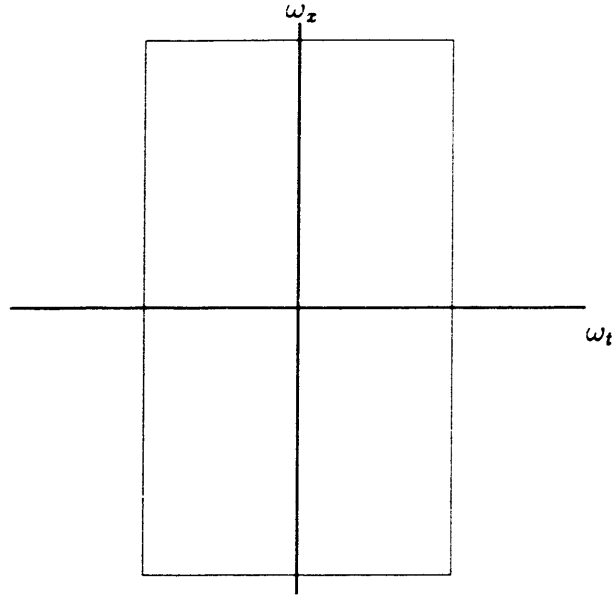


Figure 1.2: Low Frequency Temporal, High Spatial

spatial, low temporal sequence and a low spatial, high temporal sequence into a sequence with high spatial and high temporal information. These sequences will be referred to as *channels*, and the process that separates and processes the two input channels is called a *two-channel encoder*. Ideally, it will reconstruct the “missing corners” of the square in frequency space (see figure 1.3).

There is much basis for two channel encoding. It has been shown that, due to psychophysics, the motion of high spatial frequencies is perceived differently than low spatial frequencies [ADEL85], [GLENN84]. Therefore,

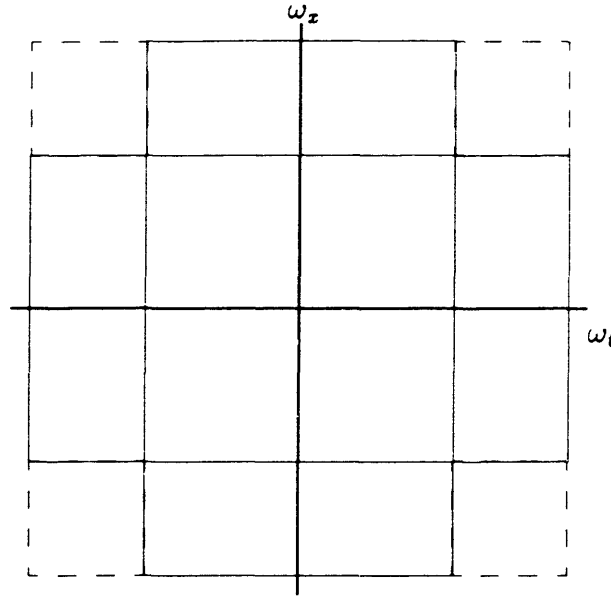


Figure 1.3: High Spatial, High Temporal Frequencies

a high spatial resolution channel has a different bandwidth requirement than a low spatial resolution channel.

A large body of research has been done on the problem of two channel encoding. Feldman has had success with the vector quantization of two channels [FELD87]. Schreiber and Troxel have developed real-time hardware for the separation into and subsequent reconstruction of a signal into two channels [TROX81], [SCHREIBER81]. Also, Kuo has developed a dual frame rate image coding system [KUU87]. The success of all these projects gives motivation for applying two channel encoding techniques to the problem of increasing both spatial and temporal resolution.

One application for this could be the encoding of movies. Given a fixed amount of light, one could choose between shooting 35mm frames or shooting 8mm frames at four times the frame rate of the 35mm frames. The encoder could take as an input to one channel a 35mm movie which had high spatial information but a low frame rate (temporally poor resolution). The input to the other channel could be an 8mm movie with low spatial information but a high frame rate. The output could then be reconstructed to contain high spatial and high temporal information.

Another use of two channel encoding is with high definition television (HDTV). One of the major efforts in HDTV research is to be able to transmit a high definition signal from which older, non-HDTV sets could receive non-enhanced information and HDTV sets could receive the high-definition information. A way to do this would be to transmit the normal, NTSC information over one channel, and to transmit the additional information needed for high definition over another. An HDTV receiver could then use two-channel encoding techniques to combine the two signals into one high-definition picture. These methods would be backwards compatible. This is desirable because older television sets are not rendered obsolete.

An additional way of looking at the work being done is to think of it

as a new method of interpolation. Standard methods of interpolation used with digital signal processing are sample and hold, linear interpolation, and band-limited interpolation.[OPPEN75] The two-channel work can be thought of as an extension to these methods, where additional information is being obtained from the second channel. This additional information, whether it be spatial or temporal, can be used to interpolate the missing data more accurately.

Chapter 2

Two-Channel Encoding by Frequency Separation

2.1 Overview

One way to approach the problem of two-channel encoding is by using frequency separation. For this, two channels are used with the characteristics described earlier: one has high temporal but low spatial resolution, while the other has low temporal but high spatial resolution. For convenience, these two channels will be referred to as the *HTLS* and *LTHS* channels, respectively. Evidence for using such a technique has been found by Schreiber

([TROX81], [SCHREIBER81]), Kuo ([KUO87]), and Glenn ([GLENN84]).

Given the two channels, the LTHS channel is high-pass filtered to obtain only the high spatial frequencies. This effectively removes everything but the detail from the frame. These high frequencies are then added to the next N frames of the HTLS sequence, where N represents the number of HTLS frames for every LTHS frame. Thus, the output sequence will have the positional information from the HTLS channel, and the detail from the LTHS channel.

This process is illustrated in figure 2.1. In this example, four frames of the HTLS channel are shown as small boxes to represent their low spatial resolution. One large box is shown for the LTHS channel to represent its high spatial resolution. Three additional frames are shown as dashed boxes which are the constructed frames made from the low spatial frequencies from the HTLS channel and the high spatial frequencies from the LTHS channel.

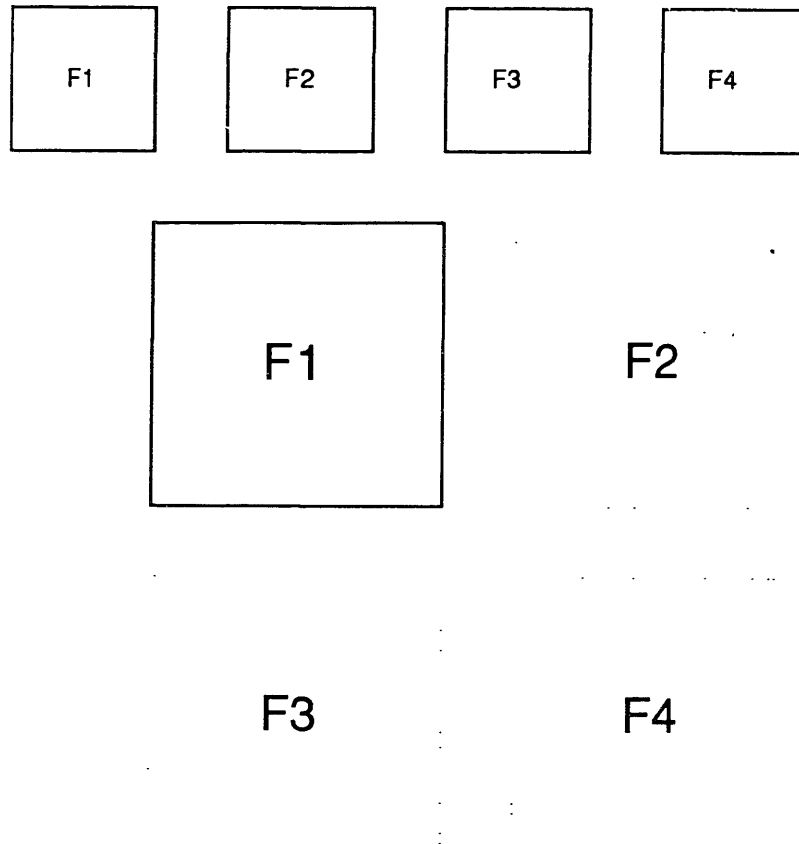


Figure 2.1: Illustration of HTLS and LTHS channels

2.2 Process Used

The first thing that needs to be done is to decide on a suitable model for research. Two channels of the same image sequence with the desired spatial and temporal characteristics need to be obtained. These were made by using one original sequence, with the LTHS channel created by simply removing certain frames, and the HTLS channel created by low-pass filtering the original sequence.

For this simulation it was decided to have the HTLS channel have four times the frame rate of the LTHS channel. Thus, the output would consist

Error Per Frame

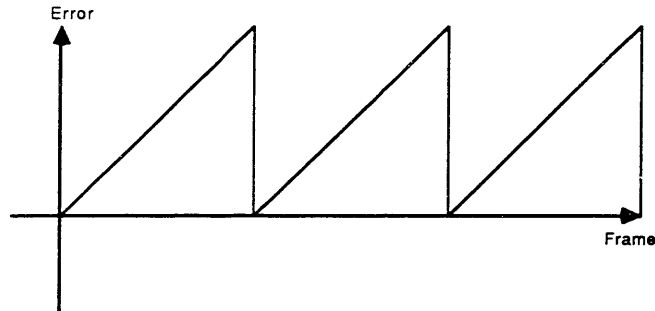


Figure 2.2: Approximate Error per Frame

of one perfect frame from the LTHS channel, and three frames created by the reconstructed algorithm. Furthermore, the error will be a function of $n \bmod 4$, where n is the frame number in the sequence. This is illustrated in figure error. In other words, the error for a particular frame is not a function of the number of frames being created. Thus, a 4:1 temporal ratio was chosen to experiment with.

A second parameter that needs to be selected for the model is the spatial frequency difference between the two channels. For experimentation purposes two different cutoff frequencies of $.5\pi$ and $.25\pi$ were used in cre-

ating the HTLS channel. A 7×7 gaussian filter was used to create the filtered image. The high-pass filtered image was obtained by subtracting the filtered image from the original image, leaving only the high spatial frequencies.

An illustration of the process can be seen in figure 2.3. Four frames are shown representing the original sequence. These are then low-pass filtered to obtain four frames of the HTLS sequence (figures 2.4 and 2.5 show how the low-pass filtered images look). The first filtered frame is subtracted from the first original frame to obtain the high spatial frequencies from that frame. These are then added to the four low-passed frames to get four output frames.

2.3 Results

2.3.1 Qualitative Results

Half Frequency HTLS Channel

The first test model used created the HTLS channel using a 7×7 gaussian filter with a cutoff of $.5\pi$. Adding in the high spatial frequencies from the

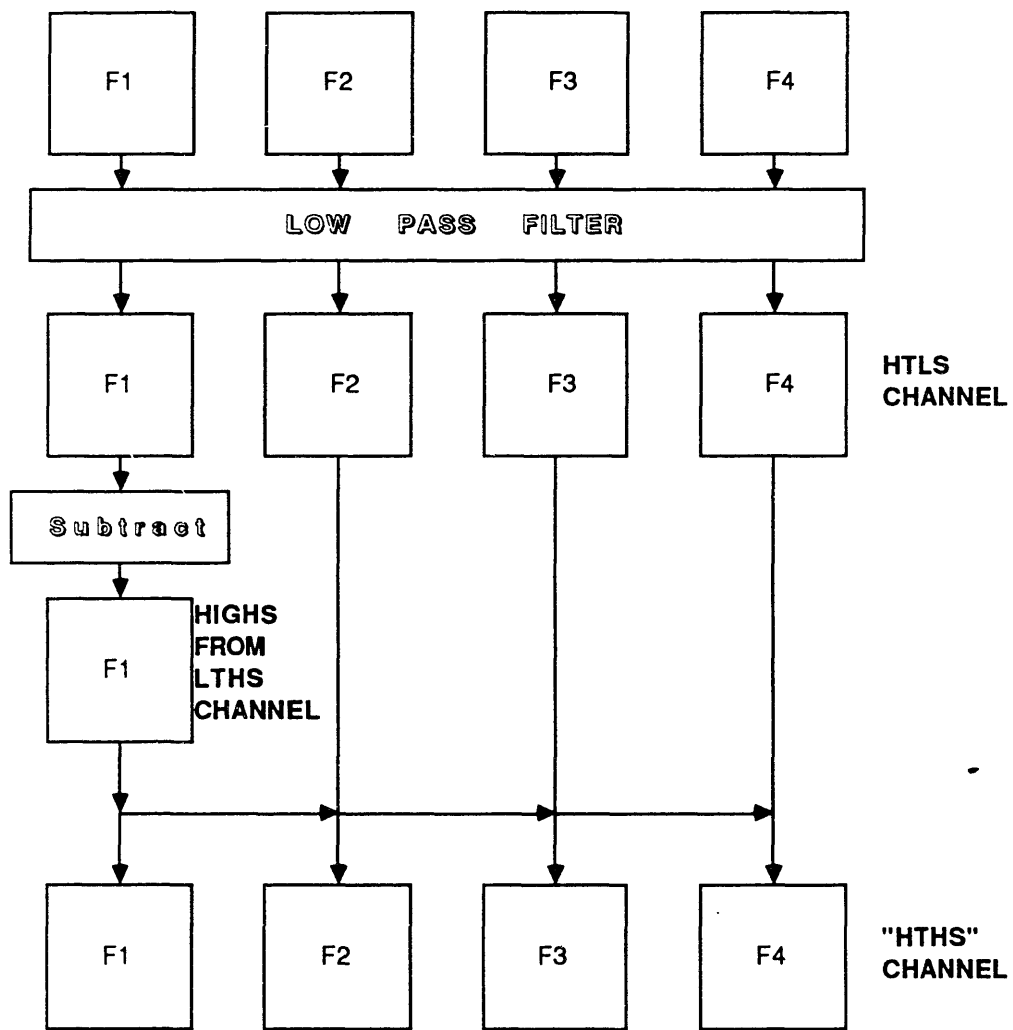


Figure 2.3: Block Diagram of Frequency Separation

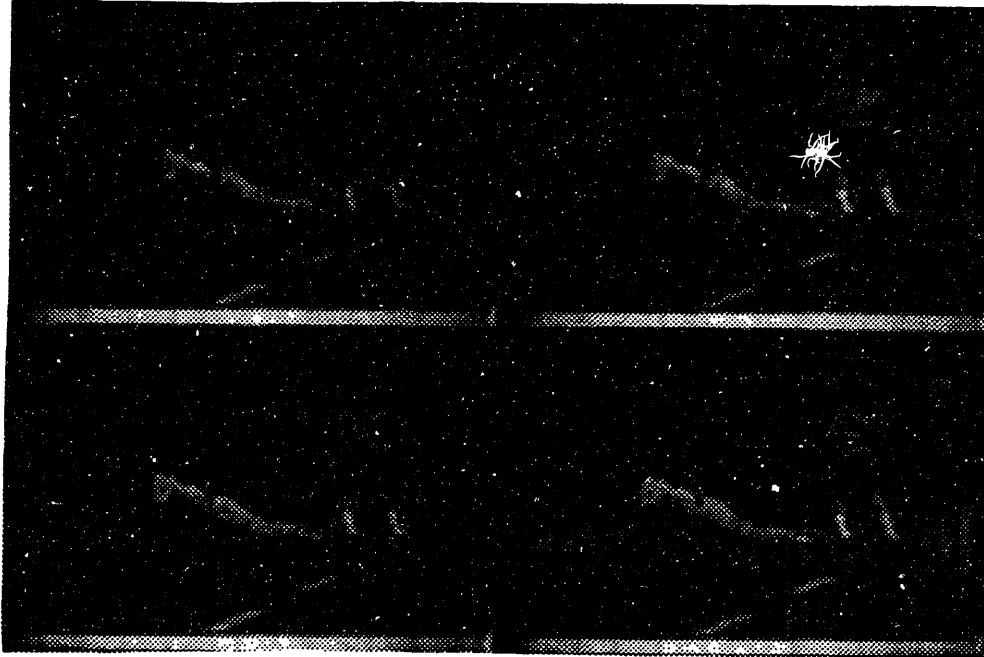


Figure 2.4: HTLS sequence created with $.5\pi$ cutoff filter

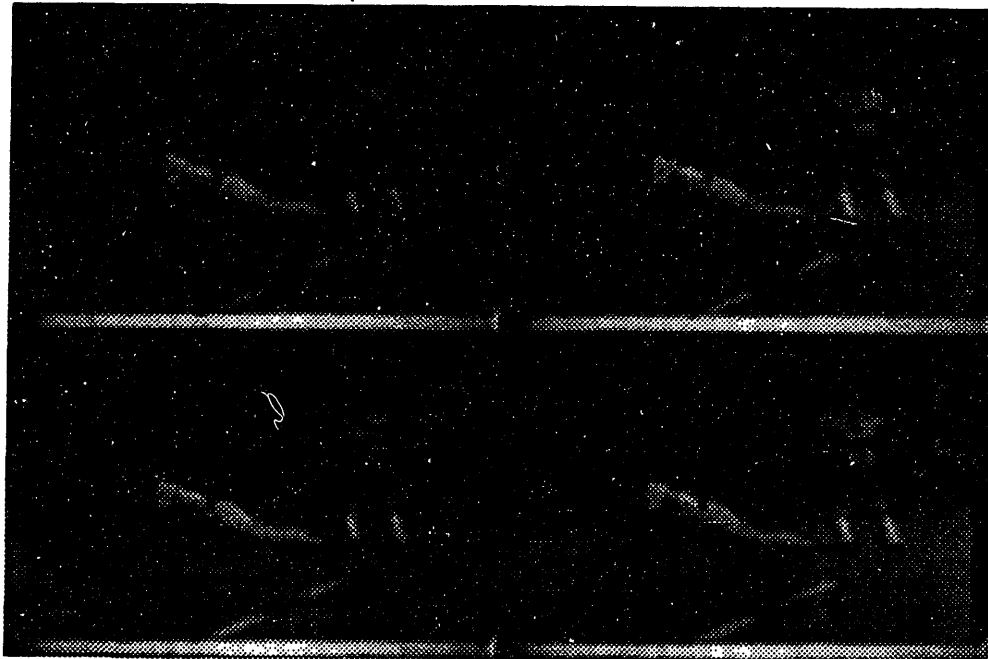


Figure 2.5: HTLS sequence created with $.25\pi$ cutoff filter

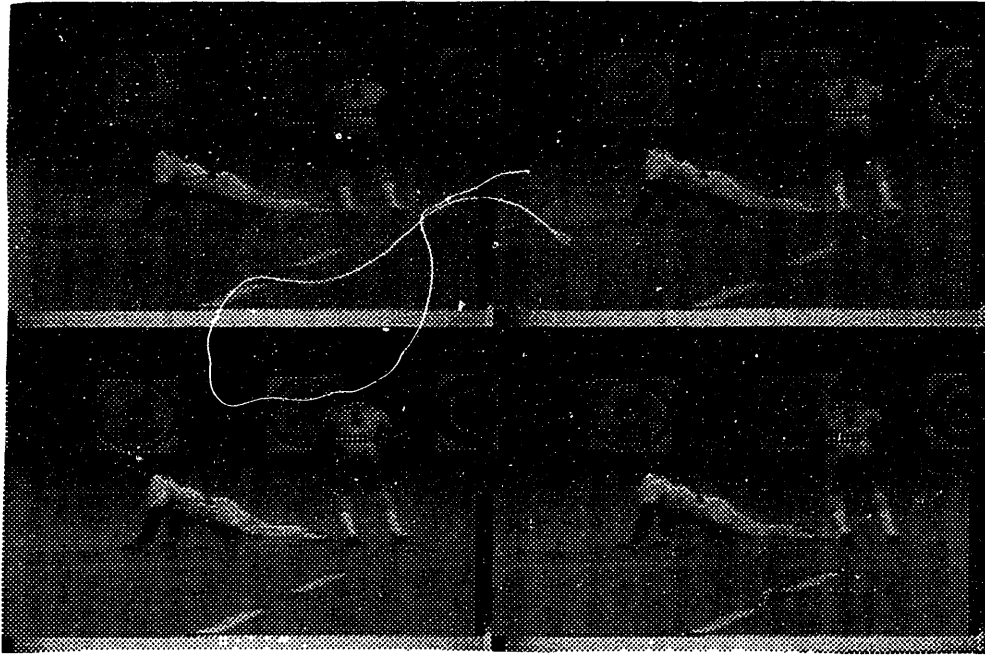


Figure 2.6: Results using half frequency HTLS channel

HTLS channel gave an output sequence in which the high-frequency detail appeared as a “ghost” superimposed on top of the low-frequency, blurred objects which moved from frame to frame. (figure 2.6)

Quarter Frequency HTLS Channel

A second desirable test would be to create an HTLS channel using a filter with a cutoff frequency of $.25\pi$. The results of this were very similar to the previous test: successive frames exhibited a “ghosting” effect between the detail and position.(figure 2.7)

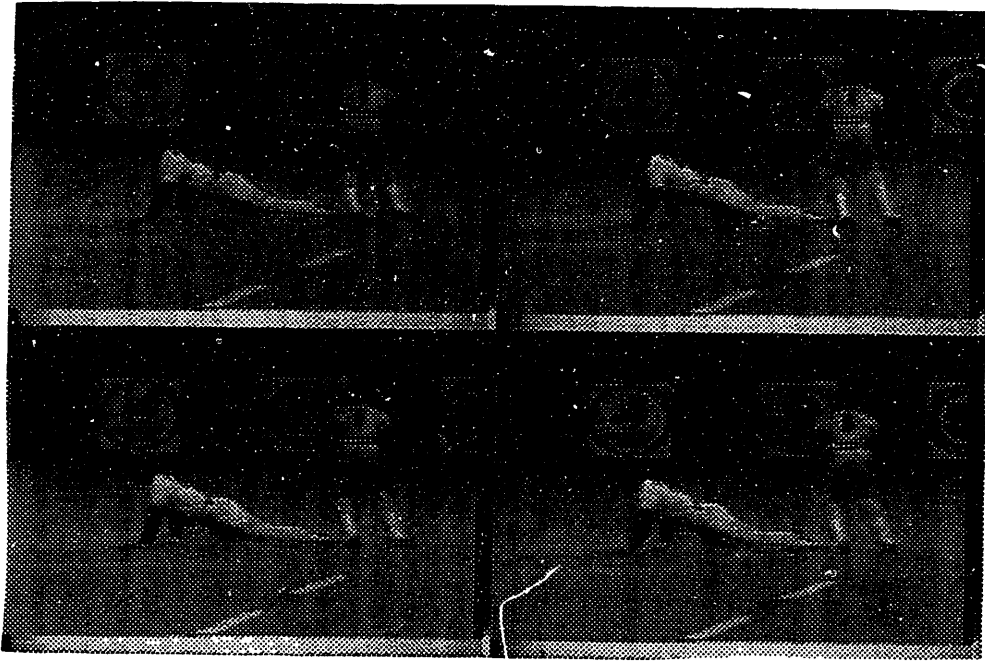


Figure 2.7: Results using quarter frequency HTLS channel

2.3.2 Signal to Noise Ratio Calculations

Description

One way to analyze these results quantitatively is by calculating the signal to noise ratio (SNR) of the image. This is defined mathematically as:

$$SNR = 20 \log \left[\frac{255}{\sqrt{\frac{1}{N} \sum_{i=0}^N (x'_i - x_i)^2}} \right]$$

which gives the SNR in decibels (dB's). In this formula, N is the number of elements in the picture, x' is the original pixel value, and x is the noisy

frame	half	quarter
zero	33.41	29.42
one	32.99	28.97
two	32.74	28.74
three	32.67	28.68

Table 2.1: SNR for unprocessed HTLS Channel

pixel value. The value 255 is used because it is the maximum pixel intensity value. The higher the value of the SNR, the more accurately the second picture represents the original. Thus, the SNR can be used as a rough estimate for quantitatively comparing the results of different algorithms.

SNR Results

To use the SNR to make comparisons, a reference needs to be calculated. This is found by calculating the SNR for each of the filtered images as compared to the original image. These were found to be as shown in table 2.1. In this table, the *half* column shows the results from finding the SNR for the images filtered by a cutoff frequency of $.5\pi$, and the *quarter* column shows the results for the images filtered by a cutoff frequency of $.25\pi$. Ideally, the algorithms developed by this research should increase these numbers.

The results of calculating the SNR showed that the algorithms indeed

frame	half	quarter
zero	∞	∞
one	39.42	35.49
two	36.19	32.13
three	34.19	30.19

Table 2.2: SNR of Reconstructed Sequence using Frequency Separation

increased picture resolution. The numbers computed are shown in table 2.2. These numbers show that, as per visual observation, the algorithm did in fact increase spatial resolution. It also agrees with the observation that the farther away in the sequence that the algorithm is applied, the less exact the results.

2.3.3 Animation Effects

Since the aim of the work is to improve the overall quality of image sequences, it is important to analyze results when the images are being animated. First, the half-frequency results were studied. Although these images had many noticeable artifacts while stationary, these disappeared once the image upon animation. The human visual system (HVS) is less sensitive to defects when the images are moving [GLENN84], [SCHRIEBER84]. The HVS seems to move the “detail” (high spatial frequencies) to corre-

spond to the positional information given by the low spatial frequencies. This is apparently due to a psycho-physical masking effect taking place in the HVS. It seems that the eye responds faster to positional changes in low spatial frequency images than in high spatial frequency images. To compensate for this problem it appears that the HVS re-positions the non-moving high-frequency components of the image to correspond to position of the low frequency components of the image. Thus, the animated sequence looks realistic.

The quarter frequency results were then looked at. They too had artifacts which disappeared upon animation. This is very interesting for a number of reasons.

In this case, because the HTLS channel only contains very low spatial frequencies, the subtraction of this channel from the LTHS channel would leave middle as well as high spatial frequencies. The fact that the HVS appears to reposition these as well is curious. More research is needed to determine the exact boundary between where the HVS will notice motion and where it will simulate motion.

2.4 Conclusions

The use of frequency separation to combine two channels gave adequate results. The output is a definite improvement compared to either using just the HTLS or just the the LTHS channel.

Improvements

An improvement to the reconstruction process can be noted by looking at figure 2.2. The error per frame rises until the next set of data is computed. An improvement is possible if, instead of just applying the high spatial frequencies forward (as in figure refforward), they are also added to the previous frame (as seen in figure refbackward). This produces the error function shown in figure refbettererror. The main disadvantage of this process is that it turns the algorithm into a non-causal process. Thus, while the algorithm would no longer work for real time data, such as video. It still, however, would work for data previously acquired, such as movies.

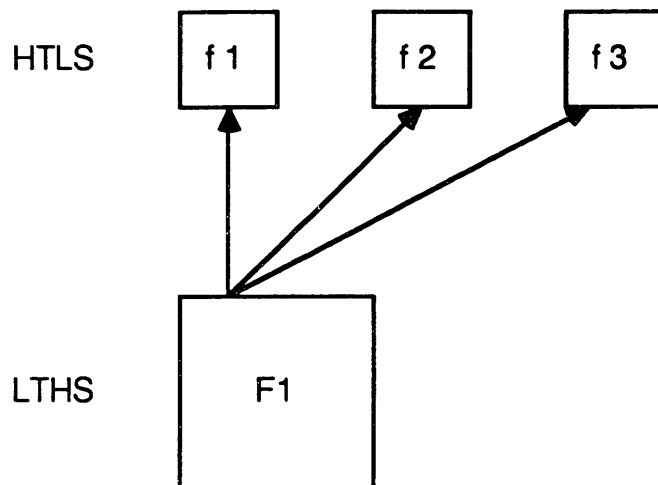


Figure 2.8: High Spatial Frequencies only Added Forward

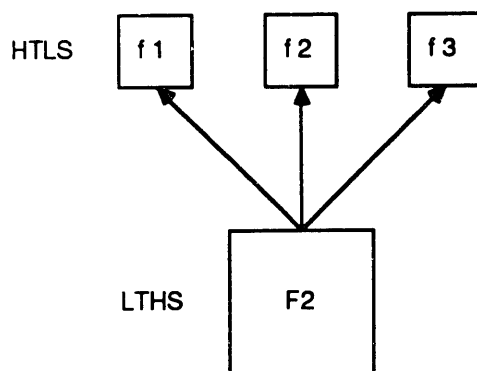


Figure 2.9: High Spatial Frequencies Added Forward & Reverse

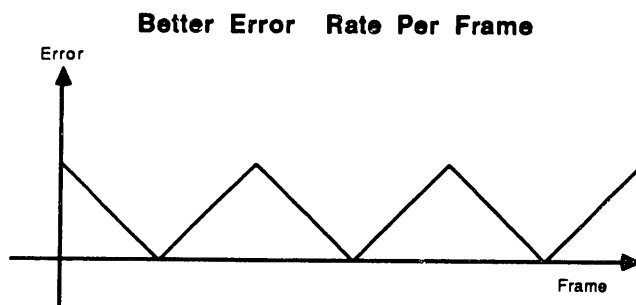


Figure 2.10: Error Function for Adding Highs both Forward & Reverse

Chapter 3

Temporal Interpolation Based on Statistical Coding

While the two-channel encoding by frequency separation described in the previous section gave reasonable results, there are a number of problems with the process. A major problem is that the high spatial frequencies being added to the HTLS channel only have positional information from the first frame. This causes the detail to be slightly offset in position, thus causing the “ghosts” that were described in the previous chapter. Ideally, the positional information from the HTLS channel should be used to determine where to place the detail information. Another property that

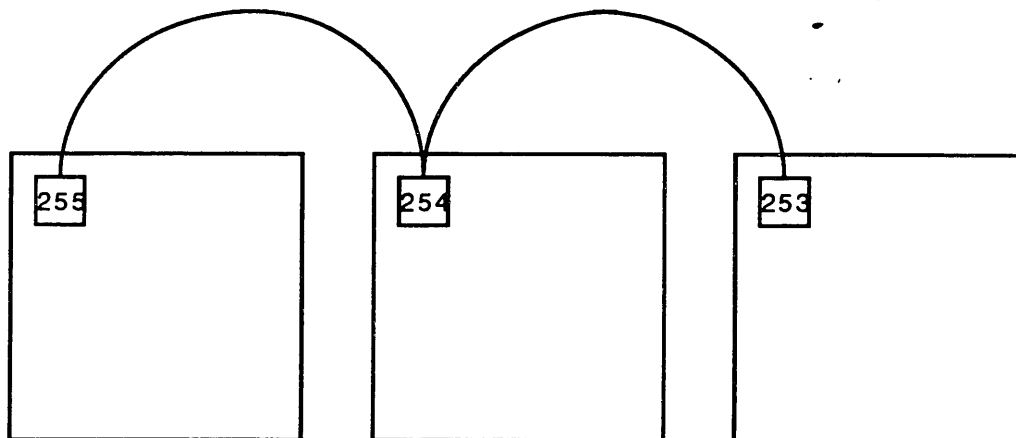
is not being exploited is the correlation between the images. Because the images are successive frames a large amount of the information is duplicated between them.

It seems that a more effective method frame interpolation would be to exploit this correlation. Methods of statistical coding can be used to determine information common between the frames of the the HTLS channel, and this information can then be used to interpolate a missing frame from the LTHS channel.

3.1 Background

To extract the statistical information from the image sequences, vector quantization (VQ) tools developed at the MIT Media Laboratory were used. These are general purpose tools used mainly for color image quantization. However, it is very simple to apply these programs to the problem of obtaining a statistical coding of an image sequence. A good discussion of vector quantization can be found in Gray [GRAY84], and Heckbert [HECKBERT80].

Before the use of these tools is discussed, some explanations and def-



Vector : [255, 254, 253]

Figure 3.1: Illustration of a Particular Vector

initializations must be made. A *vector* is a set of pixel values for a particular pixel location in successive frames. This is illustrated in figure 3.1. Here, three successive frames are shown, with a particular location emphasized. If the successive values found in this location were 255, 254, and 253, the vector for that location would be [255,254,253], and the 255 and 253 will be referred to as the *endpoints* of the sequence. What the VQ tools essentially do is sample a set of images and come up with a table of the most popular vectors in that sequence.

To apply this to the problem of two channel reconstruction, a vector table is created for a set of frames from the HTLS channel. Two frames of

the LTHS channel are then analyzed, using the pixel values at each location as endpoints to a vector. The middle value of the vector corresponds to the interpolated frame that will be created. This is done by taking each vector from the LTHS channel and searching through the vector table from the HTLS channel to find the closest match. When the vector is found with the closest match, the middle value is used as the pixel value in the frame that is being created. This process is illustrated in figure 3.2.

3.2 Description of Algorithm

As described earlier, for this model the HTLS is assumed to have twice the frame rate of the LTHS channel. Also, as in earlier tests of two channel encoding, two different filters were used to create the HTLS channel: one with a cutoff of $.5\pi$, and the other with a cutoff of $.25\pi$. Three frames of the sequence are input to the vector-maker program, which creates a vector table of 1024 entries. The vector table is then sorted by popularity, with the least popular of the 1024 vectors first and the most popular vector last. This new list is scanned, and the endpoints of the vectors are looked at. If a vector is found that has the same endpoints as a more popular vector,

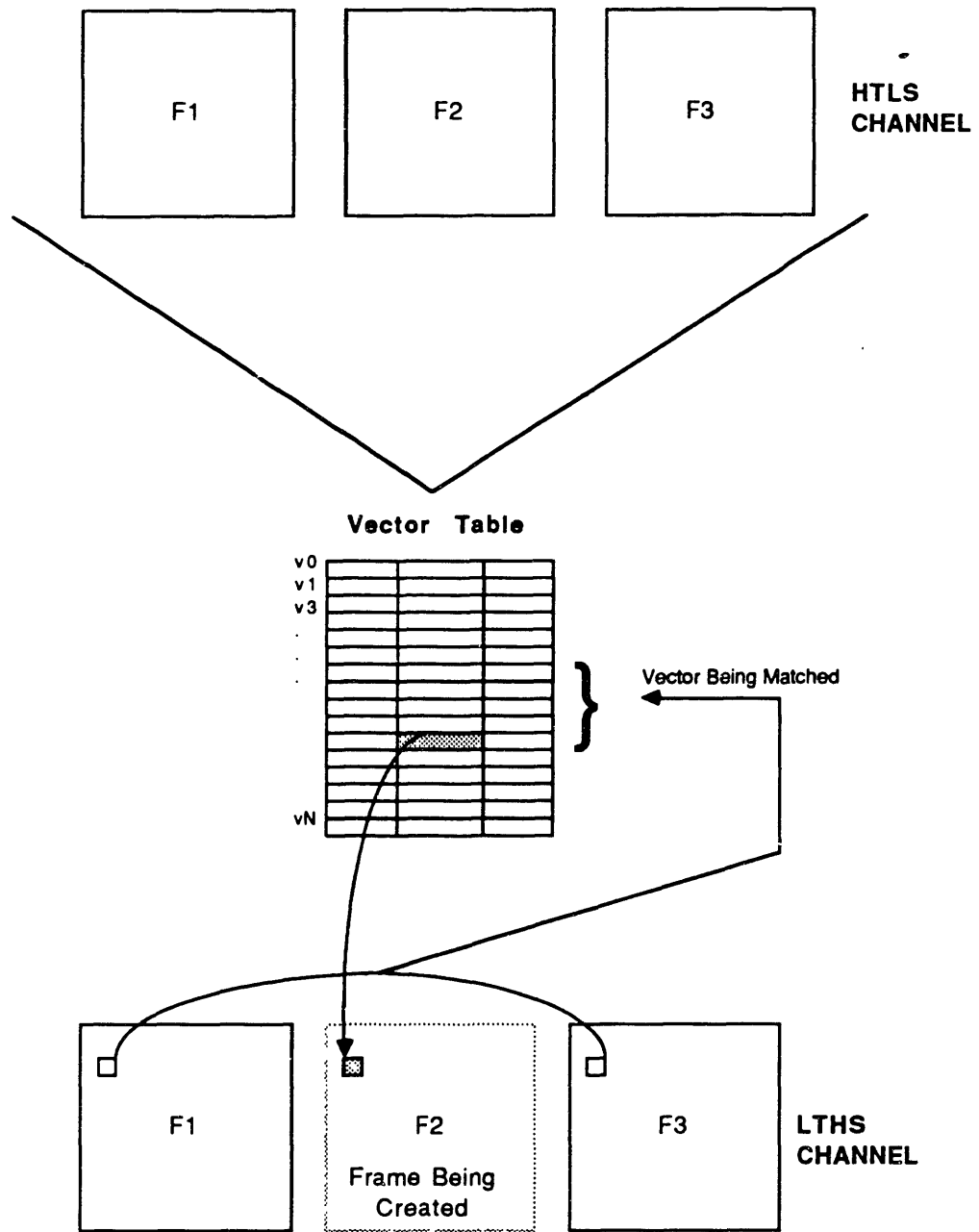


Figure 3.2: Flow of Temporal Interpolation Algorithm

the vector with less popularity is removed from the table. This is done so that when the searching is done on the new table, a match is found based on popularity of the endpoints.

To generate the missing frame, the two frames from the LTHS frames are iterated through by location, with the pixel values at each location providing the endpoints to a vector. This vector is then compared to the vector table that has been created from the HTLS channel, and a closest match is found. The middle value from this match is then used as the value for the location in the frame being created.

3.3 Results for Creating a Single Frame

The process of searching through the vector table to find the closest match was not entirely straightforward. Because the two frames from the LTHS channel only give two out of three elements of the vector, the search that is being done has a complete degree of freedom in the middle element. This cannot be ignored when making a search. Because of this, two different methods were used to perform the search. These were as follows:

1. Average the first and last elements to make a middle element, and then search on all three values.
2. Take the value from the corresponding location of the middle frame of the HTLS sequence as the middle element, and search on all three values.

In addition, the original vector table (where duplicate entries were not removed) was also searched for comparison.

3.3.1 Qualitative Results

The results of this process can be seen in figures 3.3 and 3.4. These pictures show the results for working with the LTHS channel created with the half frequency filter. Figures 3.5 and 3.6 show the results when the quarter frequency filter is used to create the HTLS. The middle frames of the sequences (created by the algorithms) clearly show the effectiveness of the process: high frequency detail is easily seen to have been added to the image, and there is none of the “ghosting” exhibited by the earlier process of frequency separation.

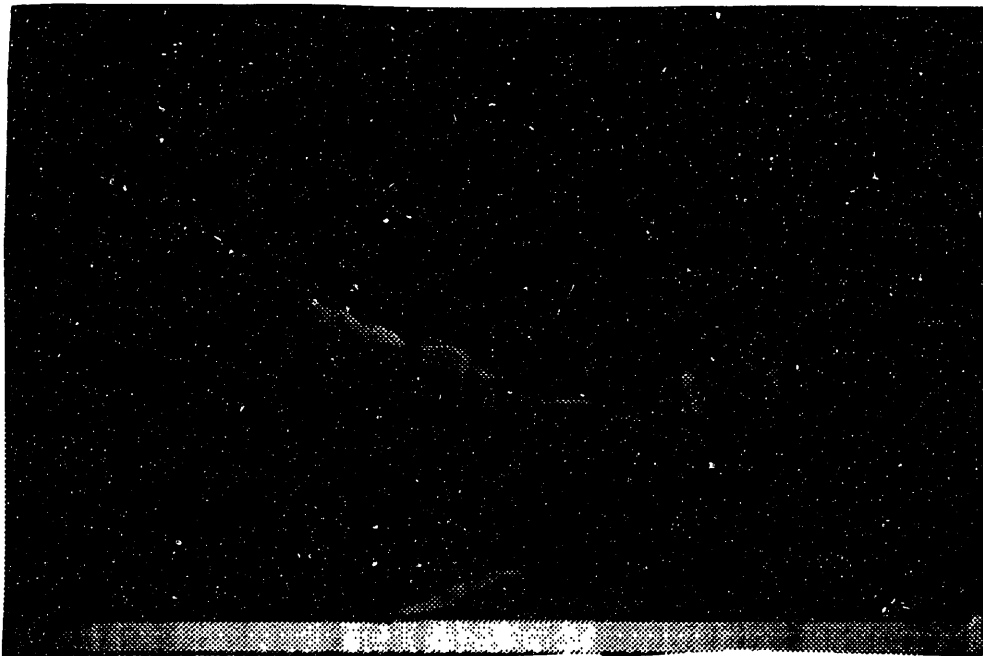


Figure 3.3: Half Frequency Channel Results, Middle Element Averaged



Figure 3.4: Half Frequency Channel Results, Middle Element from HTLS Channel

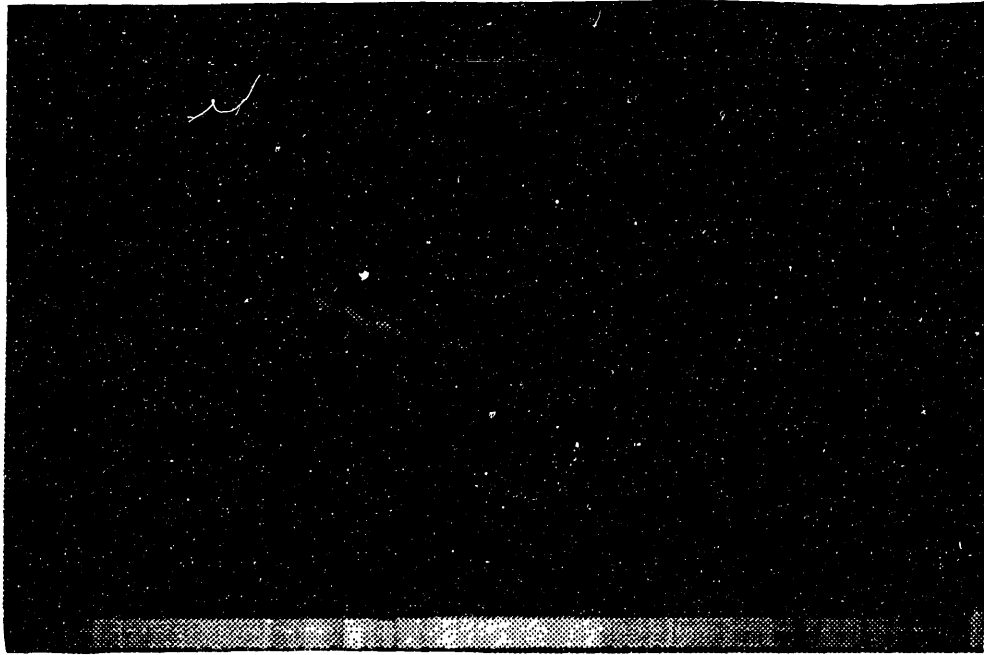


Figure 3.5: Quarter Frequency Channel Results, Middle Element Averaged



Figure 3.6: Quarter Frequency Channel Results, Middle Elements from HTLS Channel

Method	Half	Quarter
Full, av	40.16	40.21
Full, md	37.39	35.05
Reduced, av	40.21	40.27
Reduced, md	37.67	35.08

Table 3.1: SNR for Interpolating a Single Frame

3.3.2 Animation Effects

Upon animating the three frame sequences, an improvement over the frequency separation process is again noticed. When viewing the sequence, it is seen that the high frequency detail is now changing position in the middle frame to correspond to the positional information gained from the HTLS channel. This is a significant result.

3.3.3 Signal to Noise Ratio Calculations

The SNR calculations on the generated frames also show an improvement. This is seen in table 3.1. The entries in this table need a few explanations. *Full* means the search was conducted on the full vector table, while *Reduced* indicates that the search was done on the vector table which had been stripped of entries with duplicate endpoints. *Av* means the search was

done on all three points of the table, and that the middle element of the test vector was created by averaging the endpoints. Finally, *md* means the same as *av*, except that the middle element in the test vector was obtained by using the value from the corresponding pixel location in the HTLS channel.

Comparing this data with table 2.1 shows that every method produces some improvement, as every entry in this table is greater than the corresponding entry in table 2.1. This data also indicates that the best method seems to be the search on the vector table which has been stripped of duplicate endpoint entries, and that the search should be done on all three entries, creating the middle point by averaging the endpoints.

3.4 Interpolating Two Frames

The success of interpolating a single frame leads to an additional question: how many frames can be effectively interpolated? As a start, the algorithms were enhanced to be able to interpolate two missing frame. For this model, it is assumed that the HTLS channel has three times the frame rate of the LTHS channel.

3.4.1 Procedure

Basically, the two middle frames are interpolated in the same manner as for just one single frame. There are now two degrees of freedom when searching through the vector table (due to the two missing frames rather than one), so it is expected that results will not be as good. However, due to the symmetry created by having an even number of frames, there is now an additional method that can be used to do a four element search through the vector table. This is akin to a sample and hold procedure. When doing the search, the second element of the vector to be matched is obtained by simply duplicating the first element, and the third element is obtained by duplicating the fourth element (figure 3.7).

3.4.2 Results

As before, experiments were done with two different methods of creating the HTLS channel: one by filtering with a filter that has a cutoff frequency of $.5\pi$, and the other by filtering with a filter that has a cutoff frequency of $.25\pi$. A vector table was generated from this channel, and searches were done as described earlier.

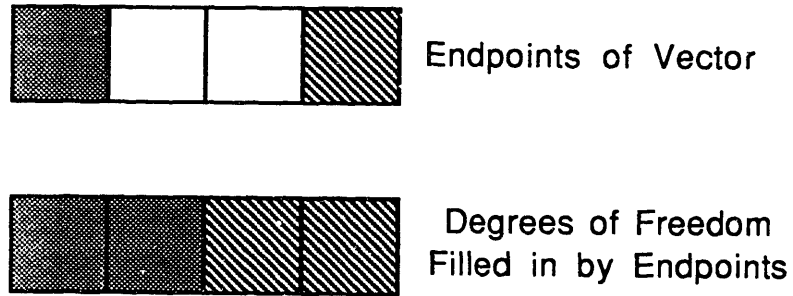


Figure 3.7: Duplicating Entries in Four Element Vector

As expected, interpolating two frames did not produce results that were as good as only interpolating a single frame. However, the generated images themselves were a reasonable success. A typical resultant sequence is shown in figure 3.8. Animating these frames produced satisfactory motion.

SNR Calculations

Calculating the SNR for this algorithm produces the data shown in table 3.2 and 3.4.2.

The terms used in this table are for the most part the same as seen in the table in section 3.3.3. The one new term is *hold*, which refers to

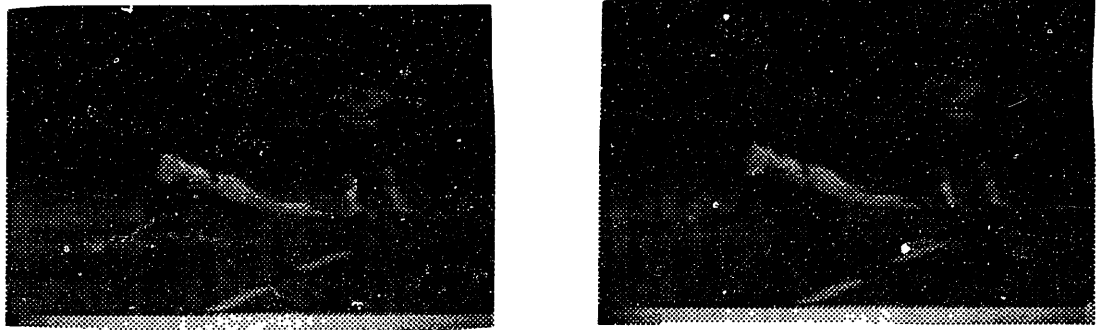


Figure 3.8: Two Interpolated Frames using Half Frequency HTLS Channel, Averaging to produce Middle Search Elements

Method	Frame 1	Frame 2
Full, av	36.03	36.84
Full, md	34.67	35.20
Full, hold	35.29	36.90
Reduced, av	36.05	36.83
Reduced, md	34.71	35.24
Reduced, hold	35.29	36.88

Table 3.2: Half Frequency HTLS Channel

Method	Frame 1	Frame 2
Full, av	36.03	36.84
Full, md	31.92	32.28
Full, hold	35.67	36.97
Reduced, av	36.14	36.91
Reduced, md	32.02	32.34
Reduced, hold	35.68	36.97

Table 3.3: Quarter Frequency HTLS Channel

the “sample and hold” mentioned earlier that is used to fill in the missing elements of the vector to be matched.

The results here show that stripping out duplicate endpoint entries from the vector table helps improve the image in most cases. The best overall method for filling in the degrees of freedom seems to be the “sample and hold” procedure, although for this case it doesn’t seem to matter if the vector table is stripped of duplicate endpoint entries.

3.5 Conclusions

Two-channel encoding by statistical encoding has been shown to be an effective method of improving image quality. The major uncertainty in its use is caused by the degrees of freedom encountered when a search is

performed through the vector tables. Out of the methods described here, averaging the known elements to obtain the missing elements seems to provide the best results.

A topic for future investigation should be to investigate more methods of dealing with the extra degrees of freedom. Only a few methods were used for in the initial research. It is likely that the ideal method has not yet been tested.

Chapter 4

Spatial Interpolation by Statistical Coding

The last chapter dealt with temporal interpolation by statistical coding. A high temporal sequence was coded statistically, and these statistics were used to help interpolate missing frames of a low temporal sequence. This chapter will deal with the dual of that procedure: spatial interpolation by statistical coding. Here, a high spatial channel is analyzed statistically, and this information is used to help increase the resolution of a low spatial channel. The motivation for exploring this method is that there should be more correlation statistically between pixels than there was temporally

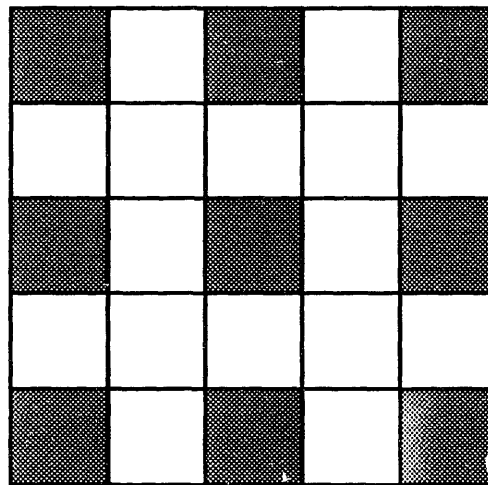
between frames.


4.1 Background

The model of the HTLS and LTHS channel used is similar to the one used earlier. The HTLS channel is taken to have twice the frame rate of the LTHS channel, and the LTHS channel is assumed to have twice the spatial resolution of the HTLS channel. Figure 4.1 shows the correspondence between the pixels of the two channels.

The coding to be done on the LTHS channel is based on groups of 3×3 pixels. Thus when the vectors to be matched are obtained from the HTLS channel, the corner pixels of the vector are from the HTLS channel, and the rest are interpolated. An example is shown in figure 4.2. This gives four points which can be matched, and five degrees of freedom from the other pixels.

In addition, the creation of the vectormap from the LTHS channel is not entirely straightforward. The vectormap to be used with the HTLS channel is for use with the middle frame. However, it is not possible to code the middle frame of the LTHS channel since by definition it doesn't



 = HTLS Pixel

 +  = LTHS Pixel

Figure 4.1: Correspondence between HTLS and LTHS pixels

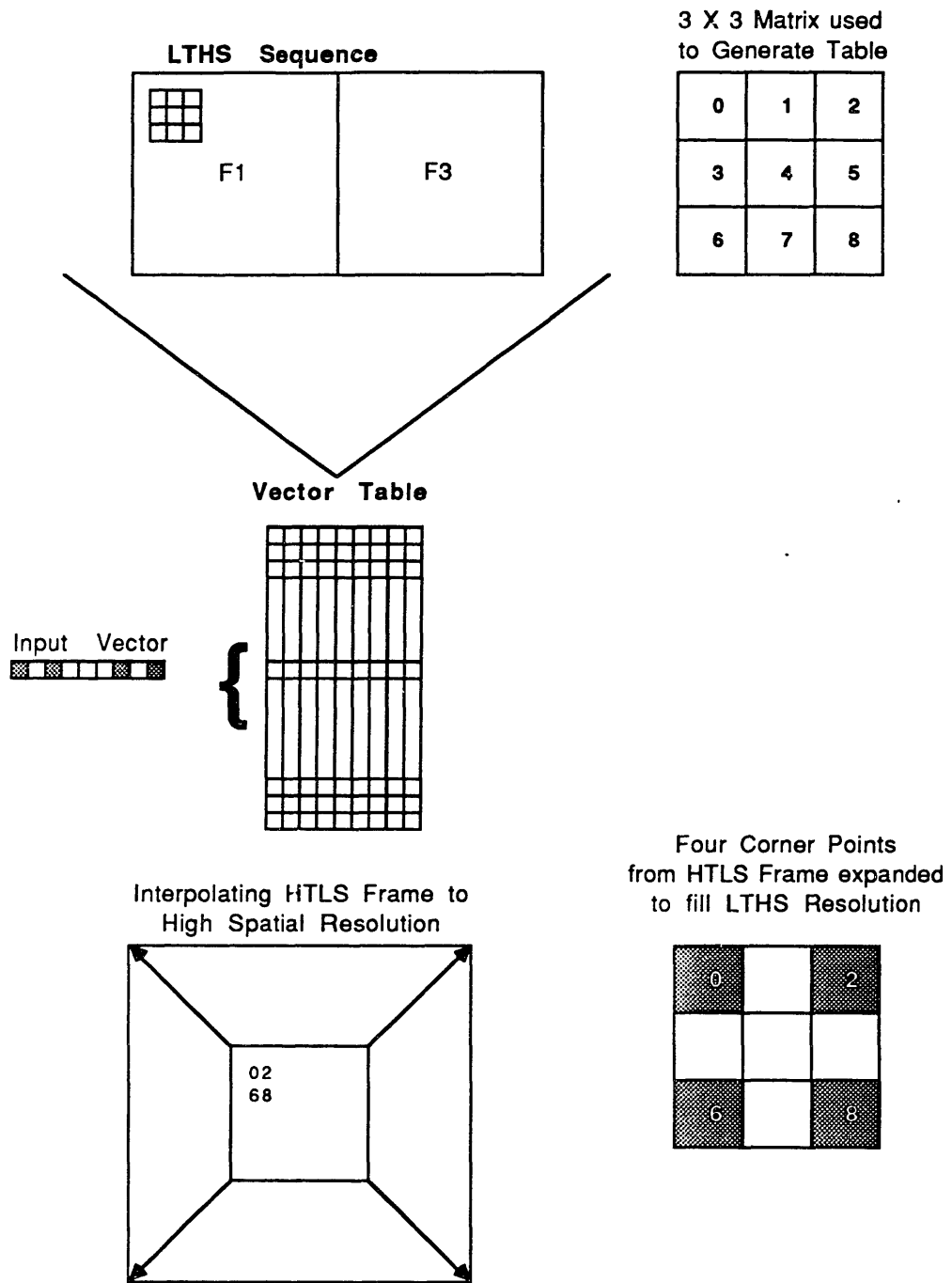


Figure 4.2: Corner points from HTLS channel

exist! Because it is not clear whether to code on the first or last frame, a compromise had to be made. Two methods were used to create a vectormap. The first method tried coded on both the frames. This was done by concatenating the first and third frames, and then generating a vector table from this. Thus, pertinent information from each frame would be incorporated in the vectortable.

The second method that was used to create a vector table was much more complicated. First, a separate vector table was created for both the first and third frames. These two tables were then concatenated and run through the vector-making algorithms. This produced a vector table of the most popular vectors between the two tables. This new table can then be used within the algorithm for spatial interpolation of the middle frame.

4.2 Description of Algorithm

The algorithm for this method is very similar to that used for temporal interpolation. In this case, each vector map entry will have nine elements, instead of the three or four used in temporal interpolation. As stated earlier, this gives five degrees of freedom. These can be filled in using the methods

described in section 3.3. The only major difference is that now there are four points to match, instead of only the two endpoints from before. Also, the case where the degrees of freedom are filled by using the middle frame of the sequence is slightly different here, as there is no middle frame. In this case, the first and third frames are averaged, and the values obtained are used to fill in the missing degrees of freedom.

A second difference in the algorithm results from the length of the table entries. In the case of temporal interpolation, the vector table output from the vector making program was searched through, and entries with duplicate endpoints were discarded (as described in section 3.2). At first a duplicate procedure was tried on the spatial vectormap. However, it was found that the procedure was not discarding any entries, or only discarded a couple. This is not unlikely, as it was trying to match four endpoints (the four corners in figure 4.2) in a 1024 point table. Thus, it was decided not to use this procedure, as it would not make much difference.



Figure 4.3: Closeup of HTLS frame upsampled by statistical interpolation

4.3 Results

Using spatial interpolation to create a missing frame produced a reasonable result (figure 4.3). However, the frame appeared to be slightly noisier than that produced by the temporal interpolation. This observation was quantified by calculating SNR for the images.

In table 4.3, *Full* means the search was done on a full vectormap created from concatenating the first and third frames. *Coded* indicates the search was done on the vectormap created by coding vectormaps from the first and third frames. The *av* entries means the degrees of freedom were filled

Method	Half	Quarter
Full, av	29.55	27.67
Full, md	30.99	29.39
Coded av	29.42	27.63
Coded, md	30.98	29.40

Table 4.1: SNR for Spatially Interpolated Images

in by averaging neighboring pixel values, while *md* means they were filled in by using values obtained from an averaged first and third frame.

This table clearly shows that the best method is obtained by using the middle values obtained from the first and third frames. It also shows that coding the vectormaps to obtain a new vectormap doesn't make much of a difference.

4.4 Conclusions about Spatial Interpolation

Overall, two channel encoding by spatial interpolation was not very successful. The SNR of the generated images was *less* than the SNR of the HTLS channel. Clearly this is not acceptable; the quality of the images has been reduced. There are a number of reasons why this method was not as successful as it was hoped it would be. The main reason is the large

amount of unknown information.

To start, the codebook could not be created on the frame that it was to be used with. The codebook that was used did not have any correct positional information for the second frame. Results might improve if a better method were found to create a frame to code on rather than concatenating the first and third frames.

A second source of error was the large number of degrees of freedom in the matching algorithm which searched through the vectormap. Only four out of nine points were known, as compared to the temporal interpolation process where two out of three points were available. The fact that only 44 percent of the points were correct in the search probably accounts for why the match wasn't very good.

Chapter 5

Conclusions

5.1 Summary

Two-channel reconstruction has been shown to be a feasible process. The examples presented earlier have shown it possible to combine a HTLS channel and a LTHS channel to create a HTHS channel. Three different methods of doing this were investigated. These were frequency separation, temporal interpolation by statistical coding, and spatial interpolation by statistical coding.

Spatial interpolation by statistical coding was shown to produce inadequate results. Temporal interpolation by statistical coding produced the

best results, but the results from frequency separation were also reasonable.

5.1.1 Methods of Choice

If this process needed to be run for an application, there are a number of criteria which would determine which method to use. If all the images were available beforehand (such as with a movie), then temporal interpolation by statistical coding would be the method to choose, as it produced the best results. However, this method cannot be used for certain applications. If the signal was being transmitted real-time (such as a TV signal) then the statistical coding methods could not be done due to their non-causal nature. Furthermore, the coding methods took a much longer amount of processing time. If real-time was an issue, then the frequency separation method should be used.

5.2 Future Investigations

5.2.1 Determination of Maximum Interpolation Range

In the investigations of temporal interpolation, attempts were only made to create one and two middle frames. An interesting experiment would

be to expand upon this, and determine the limit as to how many frames could be interpolated before the method produces inadequate pictures. It is hypothesized that this number wouldn't be more than three frames, as results has already started to diminish at two frames.

5.2.2 Image Flow and Motion Compensation

Sophisticated mathematical methods can be used to analyze an image to determine which parts of the image are in motion, and what the direction of the motion is. These use techniques of image flow and motion compensation. Worthwhile results could be obtained by analyzing the HTLS channel, and applying the motion vectors to the LTHS channel to actually "move" what is in motion. The main drawback to these methods is that they do not work very well for objects that deform. In most natural image sequences, moving objects are always changing shape. Nevertheless, the results of an investigation could prove informative.

Bibliography

- [ADEL85] Adelson, Edward H., Bergen, James R., *Spatiotemporal energy models for the perception of motion*, Journal of the Optical Society of America, Feb. 1985.
- [FELD87] Feldman, Uri, *Two-channel Vector Quantization of Moving Image Sequences*, MIT Media Lab Memorandum, Dec. 1987.
- [GLENN84] Glenn, W. E., et al, *Reduced Bandwidth Requirements for Compatible High Definition Television Transmission*, IGC HDTV conference paper, 1984.
- [GRAY84] Gray, Robert M., *Vector Quantization*, IEEE ASSP Magazine, April 1984.
- [HECKBERT80] Heckbert, Paul S., *Color Image Quantization for Frame Buffer Display*, MIT Bachelor's Thesis, May 1980.
- [KUO87] Kuo, David D., *Design of Dual Frame Rate Image Coding Systems*, MIT Master's Thesis, Feb. 1987.
- [OPPEN75] Oppenheim, Alan V., Schafer, Ronald W., *Digital Signal Processing*, Prentice-Hall, New Jersey, 1975
- [NETRAVALI] Netravali, A., and Haskell, B., *Digital Pictures: Representation and Compression*, AT&T Bell Laboratories.
- [SCHRIEBER86] Schreiber, W. F., *Fundamentals of Electronic Imaging Systems*, Springer-Verlag, New York, 1986.
- [SCHRIEBER84] Schreiber, William F., *Psychophysics and the Improvement of Television Image Quality*, SMPTE Journal, Aug. 1984.

- [SCHREIBER81] Schreiber, William F., Buckley, Robert R., *A Two-Channel Picture Coding System: II - Adaptive Companding and Color Coding*, IEEE Transactions on Communications, Dec. 1981
- [TROX81] Troxel, Donald E., Schreiber, William F., et al, *A Two-Channel Picture Coding System: I - Real-Time Implementation*, IEEE Transactions on Communications, Dec. 1981