Correction of Scan-Line Shifts of Digitized Video Images

Xiao-Song Du, Warren C. Couvillion Jr., Ming-Hoe Kiu

Digital Image Group
Department of Electrical and Computer Engineering
Engineering Research Center
Mississippi State University
Box 6176
Mississippi State, MS 39762

{xsdu, billy, kiu} @ erc.msstate.edu

Abstract

The United States Forestry Service (USFS) wishes to use satellite photos to determine the amount of land covered by forests. The USFS digitized footage shot from a moving airplane using a hand-held VHS camcorder. These images were to be used as test data to test the accuracy of the satellite. The digitization process shifted each scan-line by a random amount. Image processing techniques originally designed for detecting moving objects were modified to determine the amount each scan-line was shifted, so that a corrected image could be created by shifting the scan-lines back to their original position. These techniques approximately doubled the signal to noise ratios (SNRs) (in dB) for the special case of images with only every other scan-line shifted. Attempts to modify these techniques further to work for images where all scan-lines were shifted raised the SNR a few tenths of a dB. This was not enough to visually detect an improvement.

1. Introduction

We attempted to remove errors in digitized images of forested areas shot with a VHS camcorder from a moving airplane. The images were to be used by the USFS to determine the amount of forest cover. The calculated forest cover would be compared to that calculated from images taken using a satellite recording various frequencies. The USFS felt that the digitized camcorder images should be as accurate as possible to serve as good test data for comparison with the later, satellite images.

Most of the digitized images contained scan-lines shifted by an indeterminate amount. The shift was introduced in the digitization process, possibly caused by camera motion between successive interleaved frames, the digitizer being unable to correctly identify the VHS sync signal, or a combination of the two. The digitizer may not have been able to pick up the sync signal due to the tape stretching. The tape may have been sampled at regular, unvarying intervals, despite the tape having stretched so that it actually took different amounts of time for a single frame to traverse the player's head. The tape player camcorder motors may also have run at slightly different speeds, which would cause a similar effect.

Since the images to be corrected were produced using inexpensive, widely available video and digitizing equipment, a method of correcting errors in the images would have widespread applications. Improving the quality of digitized images shot from moving camcorders would have applications in forestry, surveillance, agriculture and rock-concert and/or political rally estimation.

Previous attempts to correct the USFS images met with little success [1]. We attempted to modify methods for determining moving objects from digitized images to determine the amount each scan-line was shifted relative to its neighbors.

2. Error Detection Overview

Our approach was to extract the major features of the image to be corrected. Since the shifts were all horizontal, we believed that discontinuities in vertical edges of the image would indicate the presences of a shift, and distance between the two discontinuous portions of the edge would indicate the amount of shift undergone by one of the lines.

2.1 Conversion to Gray ("Brightness") Levels)

The first step was to convert the color images we were given to gray scale images [1]. This was done by applying the following matrix multiplication to each pixel [2]:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

The "Y" value of each pixel corresponds to it "luminance" or brightness ("I" and "Q" values are not necessary for the black and white image). The "RGB" triple represents the red, green and blue intensity of each pixel for the color image.

2.2 Extracting Vertical Edges

After converting the images to black and white, vertical edges were extracted from the image. Since the scan-line shifts were horizontal, vertical edges in the image would be the features of the image most sensitive to the shifts, i.e., a discontinuity in a vertical edge would indicate a shift had occurred.

For each pixel, a response or gradient value was computed by summing the brightness of pixel's neighbors weighted by a three-by-three "mask" matrix, i.e.:

$$F = \left| \sum_{i=1}^{9} I_i W_i \right|$$

where "F" is the response for the center point I_5, and "I" and "W" values are given by the matrices:

$$\begin{bmatrix} I1 & I2 & I3 \\ I4 & I5 & I6 \\ I7 & I8 & I9 \end{bmatrix} \qquad \begin{bmatrix} W1 & W2 & W3 \\ W4 & W5 & W6 \\ W7 & W8 & W9 \end{bmatrix}$$

Image Gray Level *Mask Matrix*

The mask matrix for detecting vertical lines is [10]:

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Vertical Edge Detection
Mask Matrix

The resulting image is known as the "response" or "gradient" image.

3. Correction

A brief overview of our method for correcting shifts is shown in figure 3.1. A detailed explanation of our assumptions and the methods tried to determine the amount of shifts follows.

Detection of the amount of shift for each scan-lines were based on these assumptions:

1. The changes between the two successive frames are small

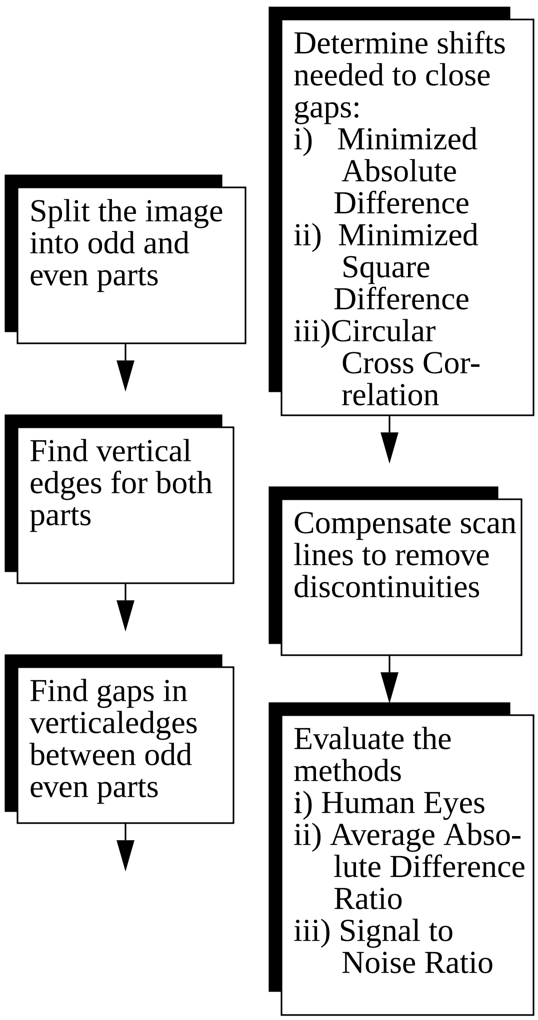


Figure 3.1: Shift Correction Diagram

2. In one frame, the changes between the adjacent odd lines and even lines are small

Given these two assumptions, any large gaps in the vertical edges would be due to shifts in the scan-lines, rather than being actual features of the image.

We tried to compute the amount of shift in the gaps by several methods:

3.1 Minimizing Absolute Differences

In this method, the value k, is chosen to minimize the function

$$D(k) = \sum_{i=0}^{n} \left| I((i+k))mod_n, j) - I(i, j+1) \right| \quad (1)$$

where I(i, j) is the pixel intensity at pixel i on line j, and the number of pixels per line is n. The value k is then assumed to be the amount that line j was shifted.

3.2 Minimizing Squared Differences

In this method, the value k is chosen to minimize the function

$$D(k) = \sum_{i=0}^{n} [I((i+k))mod_n, j) - I(i, j+1)]^2$$

where all terms are defined as for equation (1). Again , the value of k that yields the minimum D(k) is assumed to be the amount that line j was shifted.

3.3 Cross Correlation

For this method, the cross correlation function for lines j and j+1 is computed using the formula

$$Corr(k) = \sum_{i=0}^{n} I((i+k))mod_n, j) \cdot I(i, j+1) \quad (2)$$

where all terms are again defined as for equation (1). The value of k that yields the maximum cross correlation value is taken to be the amount that line j is shifted.

Given our assumptions about the image, it is obvious why one should shift the line to minimize the differences between the two lines. However, why the maximum value of the cross correlation requires some justification.

Recall the Cauchy-Schwarz inequality: for positive, real number series $a_1, a_2, ..., a_n$ and $b_1, b_2, ..., b_n$,

$$\sum_{i=0}^{n} a_i b_i \le \sqrt{\sum_{i=0}^{n} a_i^2} \cdot \sqrt{\sum_{i=0}^{n} b_i^2} \quad (3)$$

From inequality (3), it follows that if there exist two series, one of which is the other rotated,

i.e.:

$$c_i = a_{(i+k)mod_n}$$

/for some integer k, then

$$\sum_{i=0}^{n} a_i b_i \leq \sum_{i=0}^{n} a_i^2 \qquad (4)$$

Obviously, the left-hand term of equation (4) will have its maximum value when the series c_i is the same as a_i.

The correlation function given in equation (2) can be thought of as the left-hand term of inequality (4). This tells us that when Corr(k) is at its maximum, the pixel values of the lines j and j+1 are closest, which means that line j, shifted by k pixels, is maximally close to line j+1. Given our assumption that there are small changes in pixel values between lines, this means that line j shifted by k pixels is closest to being correct.

3.4 Pel Recursion

Similar to the methods described above is pel (or pixel) recursion [10]. Since the images we have are subsequent frames of a video, the amount of shift occurring for each scan-line between adjacent frames can be computed. (One frame would contain the even scan-lines, and the subsequent frames would contain the odd scan-lines, i.e., the two frames are actually part of a single, interlaced image.) If I(i, j, n) is the brightness (or luminance) of a pixel at location (i, j) at time n, then assuming that the pixel was shifted by a translation vector, $\mathbf{D}=(d_x, d_y)$, then the brightness of corresponding pixel in the next frame is given by:

$$I(i, j, n+1)=I(i+d_x, j+d_y, n)$$

The frame difference at location (i, j), FD(i, j), is the differences in the brightness values between the current and previous frames, i. e.:

$$FD(i, j) = I(i, j, n) - I(i, j, n-1)$$

which is approximately the difference between $I(i, j, n)$ and $I(i+d_x, j+d_y, n)$. Thus:

$$FD(i, j) = I(i, j, n) - I(i+d_x, j+d_y, n)$$

This equation can be written in terms of the brightness gradient, and then in pixel-to-pixel and line-to-line difference:

$$\mathbf{D'}_i = \mathbf{D'}_{i-1} + \mathbf{U}_i$$

where:

i = stage of iteration,

$\mathbf{D'}_i$, $\mathbf{D'}_{i-1}$ = new and previous estimates,

$\mathbf{U}_i$ = adjustment.

The displaced frame difference (DFD) for step "i" is defined as:

$$DFD(i, j, \mathbf{D}_i) = I(i, j, n) - I(i+d_{xi-1}, j+d_{yi-1}, n)^2$$

If G is a positive constant controlling the rate of convergence of the displacement vector, **D**, then a (hopefully) converging series of displacement vectors can be computed using the formula:

$$\mathbf{D}_i = \mathbf{D}_{i-1} - G\, DFD(i, j, \mathbf{D'}_{i-1})\ \nabla I\,(i+d_{xi-1}, j+d_{yi-1}, n-1)$$

Since there are only horizontal shifts, oI can be calculated by:

$$\nabla I = \frac{(I(i, j) - Ii, j+1))}{2}$$

If the displacement vector for a pixel in a frame converges, it can be determined where that pixel is in the subsequent frame should be located, and thus the subsequent frame can be corrected.

Pel recursion was developed as a technique of motion compensation of objects in a sequence of two dimensional images, such as the television images. It was assumed that the image contained an object in motion against an otherwise unchanging background. However, in our case, we treated each scan-line as an object. This effectively reduced the problem from two dimensionsal to one dimensional, as the “object” pixel values had to be compared only to those “background” values adjacent to it. Again we are assuming that there is little change from one scan-line to the next.

3.4.1. Procedure

Since vertical edges would most clearly indicate the scan-line shifts, we needed to extract these edges. Unfortunately, every change in brightness value creates an edge. Most of these edges are noise for our purpose, so we wished to extract only the major edges by removing the smaller features of the image.

The smaller, or higher spatial frequency features of the image were removed using a two-dimensional, low pass filter. For our case, we chose an “averaging filter”, that is we replaced each pixel with the average value of the pixels in a region five pixels by five pixels surrounding it. That is, the new pixel brightness for the pixel at (i, j), I(i,j) is given by:

$$I(i, j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} I(m+i, n+j)$$

For cases where I(i,j) was near the edge of the image, we did not “wrap around”, but replaced it with the average of value of the nearest five by five region that still fit in the image.

After low pass filtering, the gradient image was extracted using the Sobel operators as described above. We wished to disregard “weak” edges, i.e., those edges indicating only a slight change in brightness, assuming these were mainly due to noise.

To remove the weaker edges, we applied a mean filter to the gradient image using the following procedure:

1) First, we determined an appropriate threshold value for each scan-line using the equation

$$T = \frac{\left(\sum_{i=1}^{n} I_i\right)}{(n \bullet C)}$$

where “T” is the threshold value, “n” is the length of the scan-line, and “C” is an arbitrary control value specified by user. The larger C, the more likely and edge is to be selected, i.e., the fewer edges will be filtered out.

2) All of the pixels of the gradient image were then compared to the threshold value, T. Those pixels having a brightness value less than T were discarded, that is, set to 0, i.e.:

$$I_{new}(i, j) = \begin{cases} I(i, j) & ,(I(i, j) \geq T) \\ 0 & otherwise \end{cases}$$

3) We then calculate the differences between the odd and even lines of the image, as shown in figure 3.2.

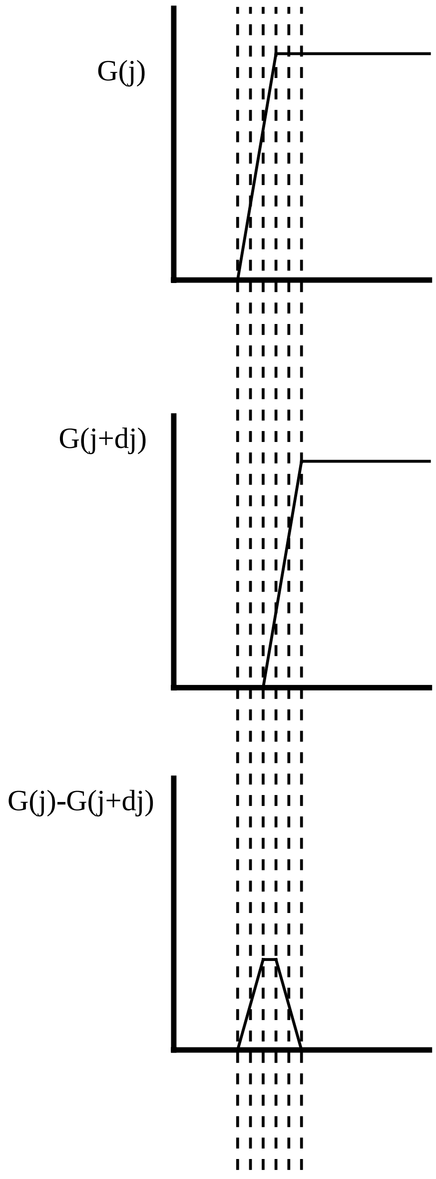
4) If the frame difference, FD, was positive then the odd line was shifted right by one pixel. If FD was negative, then the line was shifted left by one pixel.

5) Ideally, steps three and four should be repeated until a minimum frame difference is found. We found however, we found that it sufficed to set a low threshold value, and shift until the frame difference was below that value.

3.4.2 Problems with Pel Recursion

We determined that pel recursion does not work well for correcting these types of distorted images for several reasons.

In the images we examined, there was too little difference between the noise and the content of the image. Low pass filtering therefore removed not only noise, but useful information as well, “blurring” the image. Since pixel shifts may be small, only one or two pixels, sharp,



Note: G(j)-G(j+dj) = FD or Frame Difference

Figure 3.2: Frame Dif

discontinuous edges that may have revealed a discontinuity are made continuous by the lowpass filter.

Because we operated on the vertical edges, or gradient image, the oI term of equation (5) is the second derivative of the original image. Second derivatives are very sensitive to noise, so the displacement vectors,D,, calculated above could end up bringing two noisy spots on the adjacent scan-lines together, rather than correcting the image.

A third reason is the creation of "false" edges. Our assumption that pixels shifted pass the end of the frame would "wrap around" to the opposite side of the frame often creates situations that contradict our assumption that differences between scan-lines are small. For example, if an image has a very dark region on the right edge and a very light image on the left edge. If a scan-line in this region is shifted, "wrapping around" would result in dark pixels surrounded by light pixels or vice versa. The gradient image would then have an edge, which unfortunately, would be "false", in that there original image contained no edge in that area. However, since the false edge is between regions of very dark pixels and very light pixels, it would also be "strong", and thus would not be removed by either low pass or mean filtering.

An example of false edges can be seen in the appendix of images in image A.1.0 of "Lena" . On the left side of the upper part of the original image is a dark wooden frame, but on the opposite side of the image is a light region, depicting some light object in soft focus. When a scan-line containing part of the wooden frame is shifted to right, a "false edge" will be created. The gradient of the edge will be high, and therefore will not be filtered out.

3.5 Handling Sub-Pixel Shifts

All of the above methods calculate the amount of scan-line shift in units of "pixels", or whole numbers. However, there was no assurance that the USFS images were shifted by whole numbers of pixels. In fact, given the possible sources of error, it is unlikely that they were.

To handle cases sub-pixels shifts, using Nyquist's theorem, the "brightness curve" for each scan-line was recreated using to the formula:

$$I(x) = \sum_{i=0}^{n} I(i)sync(i-x)$$

where I(i) is the brightness at pixel i, and n is the number of pixels per scan-line.

I(x) was then resampled at a higher rate. For our experiments, we used 5n samples of I(x), i.e., we resampled at a rate five times that of the original image. We were then able to adjust for shifts in multiples of one fifth of a pixel.

3.6 Iteration

Minimizing the sum absolute or squared differences or cross-correlation both work well for detecting the shift of a scan-line relative to the following line. If one of the lines is unshifted, then shifting the other will correct the image, and thus these methods work well if only every other line has been shifted.

Unfortunately, almost all the scan-lines in the USFS images are shifted. We tried to adapt the methods that work when only every other scan-line is shifted by applying them iteratively. Because cross correlation was found to be one of the most effective methods for calculating line shifts, we chose to apply that in our iterative method.

For even iterations, we corrected the image as though the even lines were unshifted, and shifted odd lines, and vice versa for odd iterations. To keep successive iterations from undoing the work of the previous iterations, the lines were shifted by only half of the amounts calculated by the cross correlation method.

While there was a very small improvement after exactly two iterations, the SNRs of the corrected images began to fall for successive iterations, eventually alternating between two SNRs a few hundredths of a decibel between each other.

Despite the measure of shifting only half the amount indicated by the cross correlation method, successive iteration did appear to "undo" each other. For all test images, the improvement was only one or two tenths of a decibel, and undetectable to the naked eye.

4. Evaluation

After correction by the various methods described above, the differences in the corrected images and the original were calculated and compared to the differences between the original and distorted images. Two types of comparisons were made, as described below.

4.1 Average Absolute Difference Ratio

The average absolute difference ratio is computed using the formula:

$$Error = \frac{1}{(xy)} \sum_{i=1}^{x} \sum_{j=1}^{y} \left| I(i,j) - I'(i,j) \right|$$

where "x" and "y" are the horizontal and vertical dimensions of the image, I(i, j) are the pixel brightness values of the original image, and I'(i, j) are the pixel values of the image for which the error is being calculated.

4.2 Signal to Noise Ratio

Another method of measuring error is the signal to noise ratio, or SNR, computed using the formula:

where all values are the same as the section above.

$$SNR = 10\log\left(\frac{\left(\sum_{i=1}^{x}\sum_{j=1}^{y} I(i,j)^2\right)}{\sum_{i=1}^{x}\sum_{j=1}^{y} (I(i,j) - I'(i,j))^2}\right)$$

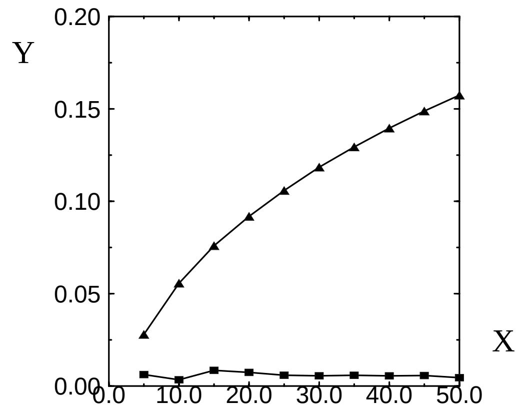
The SNR ratio has an advantage over the average absolute error ration in that it it closer to the human eye. That is, if person were asked to judge the quality of the corrected image, his evaluation would more closely correspond to the SNR than the average absolute difference ratio.

5. Results

This section contains graphs and charts showing the results of the various methods described in the sections above.

Table 5.1.1: Error Correction for "Lena" Using Minimized Absolute Difference Method (Average Absolute Difference Ratio)

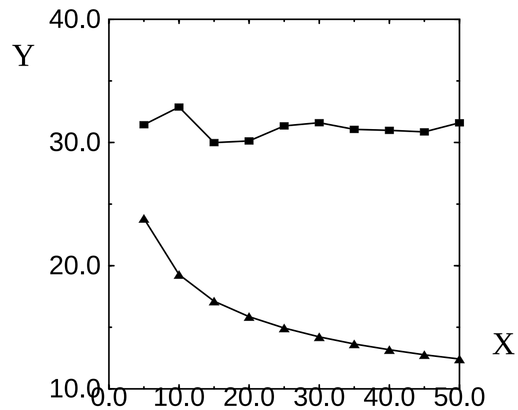
Number of shifts (pixels)	Shifted Image	Corrected Image
5	0.027932	0.006205
10	0.055624	0.003387
15	0.075922	0.008516
20	0.091828	0.007375
25	0.105804	0.005854
30	0.118471	0.005544
35	0.129409	0.005841
40	0.139591	0.005518
45	0.148832	0.005706
50	0.157424	0.004529



Y: Average Absolute Difference Ratio
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image
Figure 5.1.1: Graph for Table 5.1.1

Table 5.1.2: Error Correction for "Lena" Using Minimized Absolute Difference Method (Signal to Noise Ratio)

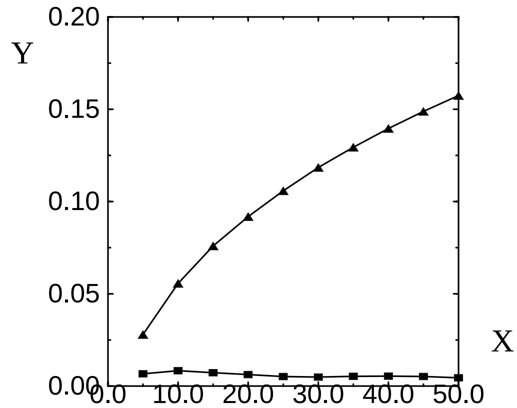
Number of shifts (pixels)	Shifted Image	Corrected Image
5	23.829769	31.441505
10	19.274696	32.875484
15	17.118313	29.988087
20	15.862347	30.125053
25	14.940739	31.344040
30	14.222882	31.607264
35	13.648379	31.064394
40	13.176785	30.985987
45	12.770381	30.857838
50	12.424780	31.598087



Y: Signal to Noise Ratio (dB)
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image
Figure 5.1.2: Graph for Table 5.1.2

Table 5.1.3: Error Correction for "Lena" Using Minimized Square Difference Method (Average Absolute Difference Ratio)

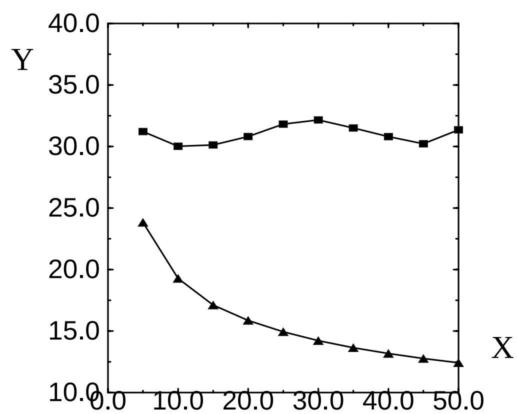
Number of shifts (pixels)	Shifted Image	Corrected Image
5	0.027932	0.006656
10	0.055624	0.008338
15	0.075922	0.007246
20	0.091828	0.006254
25	0.105804	0.005190
30	0.118471	0.004886
35	0.129409	0.005304
40	0.139591	0.005389
45	0.148832	0.005198
50	0.157424	0.004514



Y: Average Absolute Difference Ratio
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image
Figure 5.1.3: Graph for Table 5.1.3

Table 5.1.4: Error Correction for "Lena" using Minimized Square Difference Method (Signal to Noise Ratio)

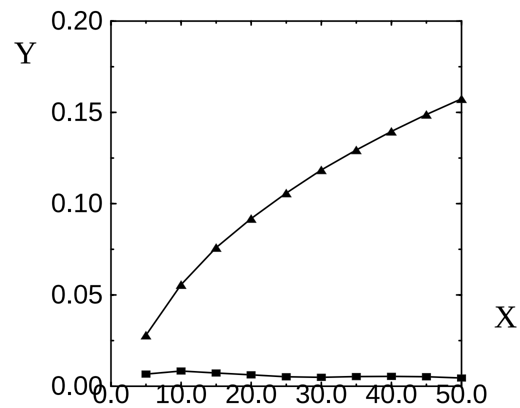
Number of shifts (pixels)	Shifted Image	Corrected Image
5	23.829769	31.213411
10	19.274696	30.016613
15	17.118313	30.124527
20	15.862347	30.810120
25	14.940739	31.816856
30	14.222882	32.157730
35	13.648379	31.505196
40	13.176785	30.801151
45	12.770381	30.221769
50	12.424780	31.353008



Y: Signal to Noise Ratio (dB)
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image
Figure 5.1.4: Graph for Table 5.1.4

Table 5.1.5: Error Correction for "Lena" Using Circular Cross Correlation (Average Absolute Difference Ratio)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	0.027932	0.006656
10	0.055624	0.008338
15	0.075922	0.007246
20	0.091828	0.006254
25	0.105804	0.005190
30	0.118471	0.004886
35	0.129409	0.005304
40	0.139591	0.005389
45	0.148832	0.005198
50	0.157424	0.004514

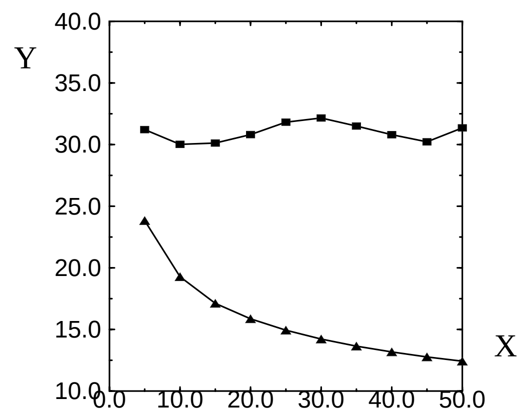


Y: Average Absolute Difference Ratio
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.1.5: Graph for Table 5.1.5

Table 5.1.6: Error Correction for "Lena" Using Circular Cross Correlation (Signal to Noise Ratio)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	23.829769	31.213411
10	19.274696	30.016613
15	17.118313	30.124527
20	15.862347	30.810120
25	14.940739	31.816856
30	14.222882	32.157730
35	13.648379	31.505196
40	13.176785	30.801151
45	12.770381	30.221769
50	12.424780	31.353008

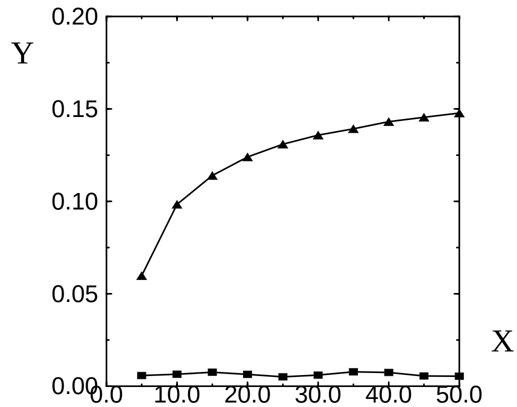


Y: Signal to Noise Ratio (dB)
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.1.6: Graph for Table 5.1.6

Table 5.2.1:Error Correction for "Landscape Photo1" Using Minimized Absolute Difference Method (Average Absolute Difference)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	0.059807	0.005788
10	0.098371	0.006480
15	0.113979	0.007577
20	0.124022	0.006409
25	0.130912	0.005083
30	0.135824	0.006029
35	0.139263	0.007802
40	0.143110	0.007437
45	0.145484	0.005550
50	0.147767	0.005439

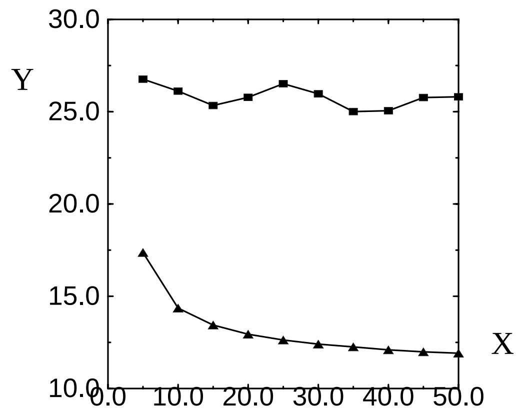


Y: Average Absolute Difference Ratio
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.2.1: Graph for Table 5.2.1

Table 5.2.2: Error Correction for "Landscape Photo1" Using Minimized Absolute Difference Method, (Signal to Noise Ratio)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	17.368773	26.763247
10	14.357676	26.115129
15	13.441083	25.333845
20	12.942650	25.778025
25	12.629339	26.516363
30	12.407932	25.969589
35	12.256508	25.005035
40	12.097805	25.051352
45	11.986806	25.767166
50	11.913532	25.805981

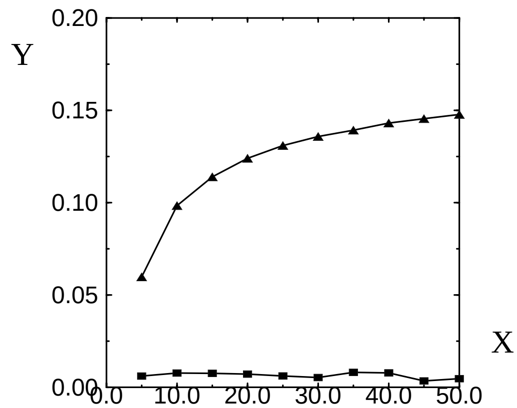


Y: Signal to Noise Ratio (dB)
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.2.2: Graph for Table 5.2.2

Table 5.2.3: Error Correction for "Landscape Photo1" Using Minimized Square Difference Method (in Average Absolute Difference)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	0.059807	0.006084
10	0.098371	0.007707
15	0.113979	0.007543
20	0.124022	0.007131
25	0.130912	0.006138
30	0.135824	0.005285
35	0.139263	0.008105
40	0.143110	0.007795
45	0.145484	0.003437
50	0.147767	0.004655

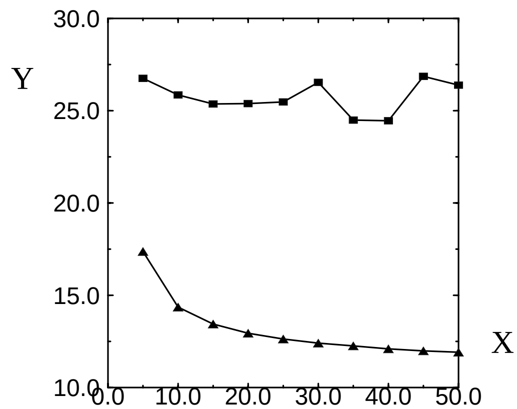


Y: Average Absolute Difference Ratio
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.2.3: Graph for Table 5.2.3

Table 5.2.4: Error Correction for "Landscape Photo1" Using Minimized Square Difference Method (Signal to Noise Ratio)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	17.368773	26.752371
10	14.357676	25.855167
15	13.441083	25.365400
20	12.942650	25.386990
25	12.629339	25.472643
30	12.407932	26.535307
35	12.256508	24.491171
40	12.097805	24.454563
45	11.986806	26.860371
50	11.913532	26.384432

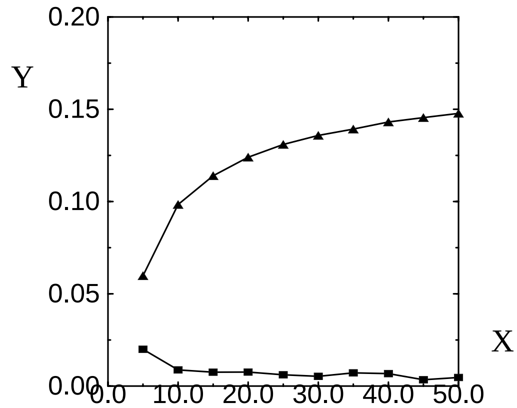


Y: Signal to Noise Ratio (dB)
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.2.4: Graph for Table 5.2.4

Table 5.2.5: Error Correction for "Landscape Photo1" for Circular Cross Correlation (Average Absolute Difference)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	0.059807	0.019955
10	0.098371	0.008786
15	0.113979	0.007548
20	0.124022	0.007605
25	0.130912	0.006138
30	0.135824	0.005285
35	0.139263	0.007171
40	0.143110	0.006757
45	0.145484	0.003437
50	0.147767	0.004655

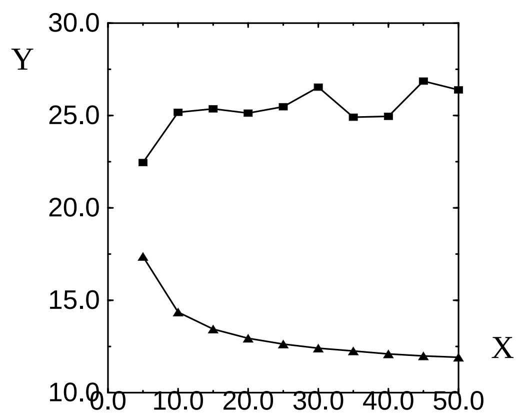


Y: Average Absolute Difference Ratio
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.2.5: Graph for Table 5.2.5

Table 5.2.6: Error Correction for "Landscape Photo1" for Circular Cross Correlation (Signal to Noise Ratio)

Number of shifts (pixels)	Shifted Image	Corrected Image
5	17.368773	22.453844
10	14.357676	25.172564
15	13.441083	25.359108
20	12.942650	25.127674
25	12.629339	25.472643
30	12.407932	26.535307
35	12.256508	24.907373
40	12.097805	24.953941
45	11.986806	26.860371
50	11.913532	26.384432



Y: Signal to Noise Ratio (dB)
X: Max Shifts
▲ : Shifted Image
■ : Corrected Image

Figure 5.2.6: Graph for Table 5.2.6

5.1 Results of four different images tested

Table 5.3.1: "Lena" with 20 Pixel Maximum Shift

Correction method	Shifted Image	Corrected Image
Before Correction	0.091828	15.86
Minimized Absolute Difference Method	0.007375	30.13
Minimized Square Difference Method	0.006254	30.81
Circular Cross Correlation	0.006254	30.81

Table 5.3.2: "Landscape Photo1" with 20 Pixel Maximum Shift

Correction method	Shifted Image	Corrected Image
Before Correction	0.124022	12.94
Minimized Absolute Difference Method	0.006883	25.50
Minimized Square Difference Method	0.007605	25.13
Circular Cross Correlation	0.007605	25.13

Table 5.3.3: "Landscape Photo2" with 20 Maximum Shift (pixel)

Correction method	Shifted Image	Corrected Image
Before Correction	0.100883	13.04
Minimized Absolute Difference Method	0.002321	32.50
Minimized Square Difference Method	0.002717	31.77
Circular Cross Correlation	0.002717	31.77

Table 5.3.4: "Landscape Photo3" with 20 Pixel Maximum Shift

Correction method	Shifted Image	Corrected Image
Before Correction	0.171154	10.54
Minimized Absolute Difference Method	0.009793	24.60
Minimized Square Difference Method	0.011545	24.06
Circular Cross Correlation	0.011545	24.06

6. Future Work

6.1 Adjacent Line Correction

One simple method of correcting images in which every line is shifted was not pursued. The method is to assume the first line is correct, calculate the shift for the second line, and shift the second line so that it is correct, relative to the first. The second line would then be assumed to be correct, and the third would be shifted, and so on. Eventually, all the lines would have the same pixel shift as the first. Although the image would still not be correct, each scan-line would be shifted by the same amount, so it would then be a simple matter to shift every scan line until the image appears correct to the user.

6.2 Faster Cross Correlation

Another area to pursue in future research is a better way to compute the cross correlation function.

First, the maximum gradient of the even line would be determined. The cross correlation would then be computed from this point using a sliding window of a size set by the user. The values of the M/2 pixels to the left and right of the maximum gradient value of the even line would be multiplied by the values in a sliding window of M pixels on the odd line, as shown in figure 8.

This method would be faster, and more robust to noise.

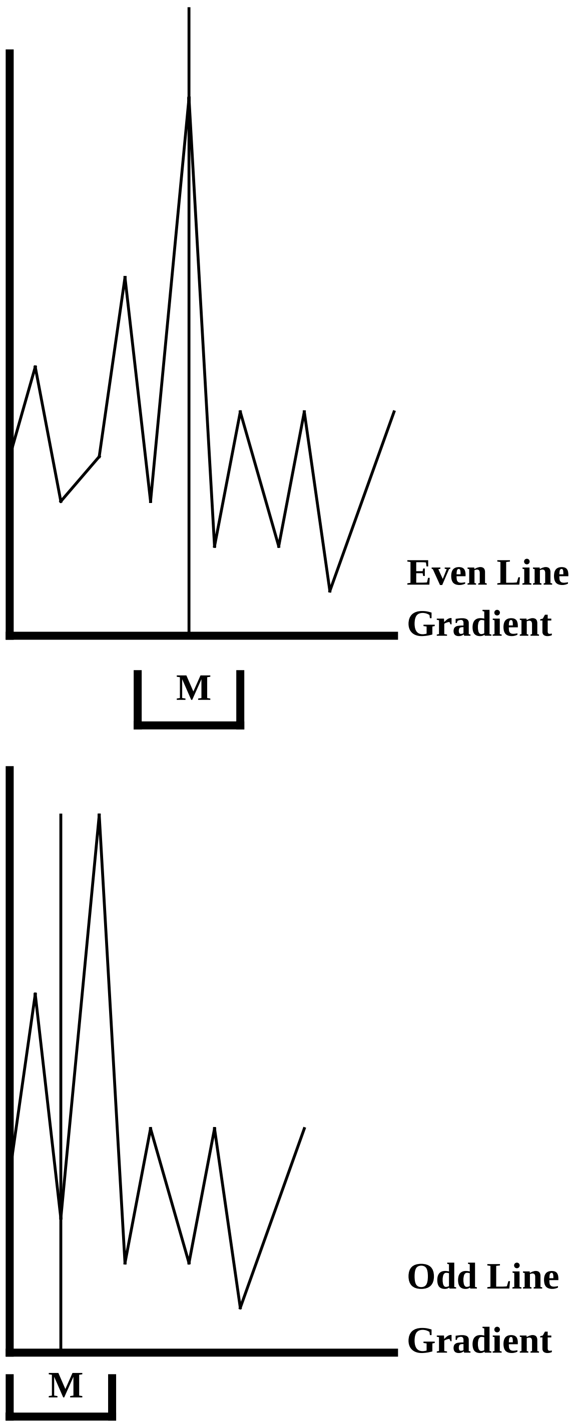


Figure 6.1: Sliding Window Cross-Correlation

7. References

1. J. G. Proakis, D. G. Manolakis,"*Digital Signal Processing; Principles, Algorithms and Applications, Second Edition*",Macmillan Publishing Company, New York, USA, 1992.

2. A. V. Oppenheim, R. W. Schafer, *"Digital Signal Processing"*, Prentice-hall, Inc., Englewood Cliffs, New Jersey, USA, 1975.

3. S. C. Saxena, *"Correcting Interfield Shifts in Interlaced Imagery,"* Unpublished report submitted to Dr. Robert Moorhead, Fall, 94.

4. J. D. Foley, A. v. Dam, S. K. Feiner, J. F. Hughes, *"Computer Graphics: Principles and Practice, Second Edition in C"*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, USA, 1996.

5. R. J. Clarke, *"Digital Compression of Still images and Video"*, Harcourt Brace & Company, London, UK, 1995.

6. R. J. Schalkoff, *"Digital Image Processing and Computer Vision"*, John Willey & Sons, Inc., New York, USA, 1989.

7. J. P. Muller, *"Digital Image Processing In Remote Sensing"*, Taylor & Francis, London, UK, 1988.

8. J.B. Campbell, *"Introduction to Remote Sensing"*, The Guiford Press, New York, USA, 1987.

9. T.S. Huang, *"Picture Processing and Digital Filtering"*, Springer-Verlag, New York, USA, 1975.

10. R. C. Gonzalez, P. Wintz, *"Digital Image Processing, Second Edition"*, Addison-Wesley Publishing Company, Massachusetts, USA, 1987.

11. R.Bernstein, *"Digital Image Processing for Remote Sensing"*, IEEE Press Inc., New York, USA, 1978.

12. R. Chellappa, A. A. Sawchuk, *"Digital Image Processing and Analysis, Vol. 2: Digital Image Analysis"*, IEEE Computer Society Press, New York, USA, 1985.

13. P. Roman, *"Some Modern Mathematics for Physicists and other outsiders"*, Pergamon Press Inc., New York, USA, 1975.

14. A. N. Netravali, J. D. Robbins, *"Motion-Compensated Television Coding: Part 1"*, American Telephone and Telegraph Company The Bell System Technical Journal, Vol. 58, No. 3, (March 1979).

15. A.N. Netravali, J. A. Stuller, *"Motion-Compensated Transform Coding"*, American Telephone and Telegraph Company, The Bell System Technical Journal, Vol. 58, No. 7, (September 1979).

16. F. W. Mounts, *"A Video Encoding System Using Conditional Picture-Element Replenishment, "* B.S.T.J.,48,No.7(September 1969).

17. C. Cafforio, F. Rocca, *"Methods for Measuring Small Displacements of Television Images"*, IEEE Trans. Inform. Theory, IT-22, No. 5(September 1976).

18. P.A. Wintz, *"Transform Picture Coding,"* Proc. IEEE, 60, No.7 (July 1972).

19. L. Ljung, *"Analysis of Recursive Stochastic Algorithms"*, IEEE Transactions on Automatic Control, AC-22, No.4 (August 1977).

20. J. L. Potter, *"Scene Segmentation Using Motion Information"*, Computer Graphics and Image Processing, 6(1977).

Appendix A: Images Tested:

Appendix A: Images Tested:

A.1.0: Sample distorted image, "Lena", with 20 pixel shift.

A.1.1: Corrected image using Minimized Absolute Difference

A.1.2: Corrected image using Minimized Square Difference

A.1.3: Corrected image using Cross Correlation

A.2.0: Sample distorted image, "Landscape Photo1", with 20 maximum pixel shift

A.2.1: Corrected image using Minimized Absolute Difference

A.2.2: Corrected image using Minimized Square Difference

A.2.3: Corrected image using Cross Correlation

A.3.0: Sample distorted image, "Landscape Photo2", with 20 maximum pixel shift

A.3.1: Corrected image using Minimized Absolute Difference

A.3.2: Corrected image using Minimized Square Difference

A.3.3: Corrected image using Cross Correlation

A.4.0: Sample distorted image, "Landscape Photo2", with 20 maximum pixel shift

A.4.1: Corrected image using Minimized Absolute Difference

A.4.2: Corrected image using Minimized Square Difference

A.4.3: Corrected image using Cross Correlation