Journal of Fluids Engineering, Vol. 120, 1998, pp. 463-470

High-Speed PIV Analysis Using Compressed Image Correlation

Douglas P. Hart

Massachusetts Institute of Technology
Department of Mechanical Engineering
Cambridge, MA 02139-4307

ABSTRACT

With the development of Holographic PIV (HPIV) and PIV Cinematography (PIVC), the need for a computationally efficient algorithm capable of processing images at video rates has emerged. This paper presents one such algorithm, sparse array image correlation. This algorithm is based on the sparse format of image data - a format well suited to the storage of highly segmented images. It utilizes an image compression scheme that retains pixel values in high intensity gradient areas eliminating low information background regions. The remaining pixels are stored in sparse format along with their relative locations encoded into 32 bit words. The result is a highly reduced image data set that retains the original correlation information of the image. Compression ratios of 30:1 using this method are typical. As a result, far fewer memory calls and data entry comparisons are required to accurately determine tracer particle movement. In addition, by utilizing an error correlation function, pixel comparisons are made through single integer calculations eliminating time consuming multiplication and floating point arithmetic. Thus, this algorithm typically results in much higher correlation speeds and lower memory requirements than spectral and image shifting correlation algorithms.

This paper describes the methodology of sparse array correlation as well as the speed, accuracy, and limitations of this unique algorithm. While the study presented here focuses on the process of correlating images stored in sparse format, the details of an image compression algorithm based on intensity gradient thresholding is presented and its effect on image correlation is discussed to elucidate the limitations and applicability of compression based PIV processing.

NOMENCLATURE

Φ	Correlation function
β	Characteristic image pixel size [*m*]
Δ	Correlation search length [*pixels*]
Δs	Imaged particle displacement [*m*]
Δt	Time between image exposures [*sec.]*
$\Delta i,\Delta j$	Difference in pixel image [*pixels*]
∇	Gradient operator
γ	Image compression ratio
$\vec{V}$	Flow velocity [*m/s*]
D	Particle image diameter [*m*]
G_∇	Relative flow divergence
I	Pixel intensity
i,j	Image coordinates [*pixels*]
m,n	Data array indices
l	Variable-length encoded data entry length [*pixels*]
$\mathcal{M}$	Image magnification
M, N	Interrogation image diameter [*pixels*]
u,v	Pixel displacement in x and y directions
x,y	Pixel image coordinates

1. INTRODUCTION

Until recently, Particle Image Velocimetry, PIV, has been limited to applications in which two-dimensional, instantaneous velocity measurements are of interest. Most flows, however, are unsteady and three-dimensional in nature and thus, there has been a growing effort to develop three-dimensional velocity measurement techniques and techniques to quantitatively resolve unsteady flows. This effort has resulted in the development of Holographic PIV (HPIV) and PIV Cinematography (PIVC). Both these techniques are highly computationally intensive often requiring the determination of millions even tens of millions of vectors. With present software processing speeds, a single experimental run using HPIV or PIVC can take several hours of computer time to obtain results. Because of this, dedicated coprocessors are often utilized in these applications. These costly coprocessors, although significantly faster than present PC software processing, are still slower than desired. Ideally, one would like to process HPIV and PIVC images at a rate faster than they can be acquired. This negates the need to store the images requiring only that the results be stored. It also allows an investigator to observe PIVC results in near real-time and potentially use the information for system feedback control in much the same way LDV systems are now being used in industry.

At present, electronic imaging systems operate with pixel transfer rates on the order of 10 million pixels per second. At 8 bits per pixel, this is roughly twice the speed at which most PC's can stream uncompressed data to a hard disk. Even compressed by a factor of ten, more than one megabyte of storage is needed for each second of video signal. A typical statistical correlation with 64*x*64 pixel windows and 50% overlap requires more than 75 million multiplications and 225 million memory calls per second to process data at this rate - far faster than the capabilities of present PC technology. Fourier correlation techniques require significantly fewer operations but due to multiple memory calls and floating-point calculations, their processing requirements are still well beyond present PC capabilities for real time PIVC or video rate processing of HPIV images. Thus, if video rate PIV processing is to be achieved without the need for a dedicated coprocessor, an algorithm must be developed that significantly reduces the number of memory calls and arithmetic operations. This paper introduces one such algorithm, sparse array image correlation.

2. METHODOLOGY

Sparse array image correlation is based on storing and correlating a compressed data set that retains the particle displacement information from the original PIV image. By reducing coding and interpixel redundancy, far fewer memory calls and calculations must be made to correlate the image. PIV images typically contain significant data redundancy. Compression ratios of 30:1 or greater are normal. Thus, since the time required to correlate an image is proportional to the square of the number of data entries, significant gains in processing speed are possible.

Background

The simplest form of data reduction that can be made to a PIV image is to eliminate the low intensity pixels from the image file. Since the low intensity pixels contribute little to no information about particle displacement, their elimination has very little effect on the accuracy of the image correlation. Several high-speed algorithms have been developed based on this type of data reduction. The most recent of these algorithms is the one by Hatem and Aroussi (1995) in which a probability histogram of possible particle displacements is used to determine the velocity vector. Unlike Hatem and Aroussi's algorithm, sparse array image correlation relies on a true correlation of the image - it is not a particle tracking type algorithm and it does not rely on the binary (0,1) representation of particles. The relative intensity difference between pixels is maintained despite the utilization of an image compression scheme. A more pertinent algorithm to the present algorithm is the one by Landreth and Adrian (1987) in which each section of an image is orthoganally compressed and the low intensity pixel combinations are eliminated from the data before it is correlated. Like the present algorithm, Landreth and Adrian's algorithm processes the data in a sparse format. This is the basic scheme by which the present algorithm correlates images. Unlike Landreth and Adrian's algorithm, however, both coding redundancy and interpixel redundancy are reduced during image preprocessing without decoupling the x and y correlations. The two-dimensional spatial relationship and the relative intensity variation between pixels are maintained. Significant speed is gained by encoding the remaining data specifically for 32-bit processing and utilizing an error correlation function to eliminate multiplication and division operations.

As with all correlation schemes that require preprocessing of images, a tradeoff is made between the time required to reduce the data set and the time required to correlate the reduced data set. The original intent of the sparse array image correlation algorithm was to process PIV images at video rates. Therefore, the algorithm presented here uses a relatively simple data compression scheme to facilitate the processing of a video signal as it is being downloaded from a CCD camera. This allows a data set from a previous frame to be analyzed at the same time the video data from a camera is being compressed. It is desired to perform both preprocessing and correlation of the images at roughly the same rate, 1/30 of a second. The result is an image compression algorithm that is not necessarily optimized for data reduction but allows pipelining of the original image data set to reduce image preprocessing time and data transfer latency.

Image Compression

The first step in sparse array image correlation is to generate a data array that contains enough information to determine the displacement of particles in a PIV image or between two images in the case of cross-correlation. In order to facilitate processing, it is desired to retain the minimum amount of data to obtain a specified resolution in the final results. Unfortunately, it is difficult to determine *a priori* the exact information that is needed to achieve this. It can be shown, however, from the statistical correlation function,

$$\Phi_{\Delta i,\Delta j} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[I_{m+\Delta i,n+\Delta j}\cdot I_{m,n}\right]}{\sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N}I_{m,n}^{2}}\cdot\sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N}I_{m+\Delta i,n+\Delta j}^{2}}}$$

that pixels with high intensity contribute more to the overall value of the correlation coefficient than pixels of low intensity. This characteristic of the statistical correlation function adversely affects the ability to determine the subpixel displacement of tracer particles in a PIV image by unduly weighting the significance of high-intensity pixels. Much of the information contained in a PIV image that allows subpixel resolution of tracer particle movement resides in the intensity of pixels representing the edges of the particle images. It is not the level of pixel intensity in a PIV that allows the displacements to be determined through correlation. It is the relative *change* in intensity between the background and the tracer particle images that makes this possible. In much the same way two blank pieces of paper are aligned on a desk, image correlation relies on the change in intensity around the edges of the objects being aligned and not the featureless, low intensity gradient, regions. Thus, in principle, all pixels in low intensity gradient regions can be eliminated from a PIV image with only a slight loss in correlation information as long as the relative positions and intensities of the remaining pixels are maintained. Except for a small number of pixels representing tracer particles, PIV images are predominantly blank. Therefore, the data size necessary to determine tracer particle movement within PIV images can be significantly reduced with little or no loss in accuracy. This is the basis by which sparse array image correlation works. Eliminating pixels that have little effect on the determination of tracer particle movement reduces the data set representing a PIV image. The remaining pixel intensities are recorded in sparse format along with their relative positions. This sparse data set is then used to determine movements of the tracer particles in the fluid.

Segmentation

PIV images are strongly bimodal, composed of light particle images on a dark background, Figure 1. It is, therefore, relatively easy to eliminate low intensity, background, pixels from the data. The simplest technique to accomplish this is to set a threshold level and retain only those pixels with intensities above the threshold. A relatively robust and accurate technique for setting the appropriate threshold level is to perform a histogram concavity analysis [Rosenfield and De La Torre, 1982]. A simpler and somewhat faster technique is to generate an intensity distribution curve that indicates the number of pixels with intensities above a specified level. Since the curve is an accumulation of pixel numbers, it is piecewise smooth, at least to the resolution of the CCD camera and thus, it is a simple matter to select a threshold level that corresponds to a specific slope on the curve. This technique is not as robust or accurate as the histogram concavity analysis but, because the pixel intensities in PIV images are so strongly bimodal, the precise threshold level is often not critical.

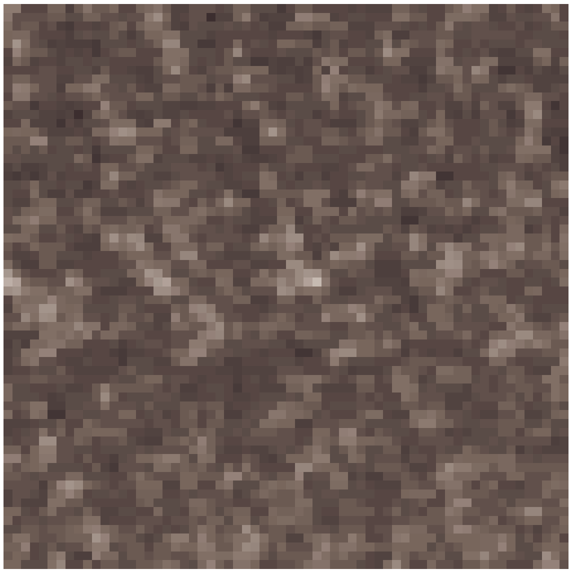

Figure 1: *Typical 64x64 pixel region of a PIV image.*

Several PIV and Particle Tracking Velocimetry (PTV) algorithms have been developed based on the intensity thresholding of images [Hart 1996, Hatem and Aroussi, 1995.]. While at first this appears to be a simple and robust way of reducing a PIV image, there are a number of difficulties with this method that makes it inappropriate for poor quality images. Consider a double exposed one-dimensional intensity plot of two tracer particles in a flow, Figure 2(*a*). The intensity profile of the particle images appear Gaussian with a spot diameter that depends on the, particle diameter, image magnification, imaged wave length, pixel size, focal length, and aperture of the camera recording the image. As illustrated by Δs_1 and Δs_2 in this figure, any gradient in the flow, $\nabla\vec{v}$, over the observed region results in unequal displacements between the first and second exposures of the tracer particles. If $(\Delta s_1 - \Delta s_2)$ is small relative to the particle image spot diameter, *D*, then the peak correlation of the sub-window is an average of the displacements represented by the double exposure of the two particles, $(\Delta s_1 - \Delta s_2)/2$, Figure 2(*b*). If, however $(\Delta s_1 - \Delta s_2)$ is large relative to *D*, then there exists no clear peak correlation, Figure 2(*c*). Although algorithms exist that are highly robust to large local velocity gradients in the flow such as the spring model algorithm by Okamoto, Hassan, and Schmidl (1995), in general, large velocity gradients result in an increase in spurious vectors. Thresholding an image has the effect of reducing the spot diameter of the particle images as illustrated by the dotted line in Figure 2(*a*). Thus, thresholding can result in a loss in the information necessary to obtain average particle displacement information. Furthermore, most PIV images suffer from an inconsistency in the relative intensity between particle images. This is particularly true of images that are under exposed. In these cases, the information lost by thresholding to obtain significant data reduction can result in the loss of particle displacement information even for relatively small flow divergence.

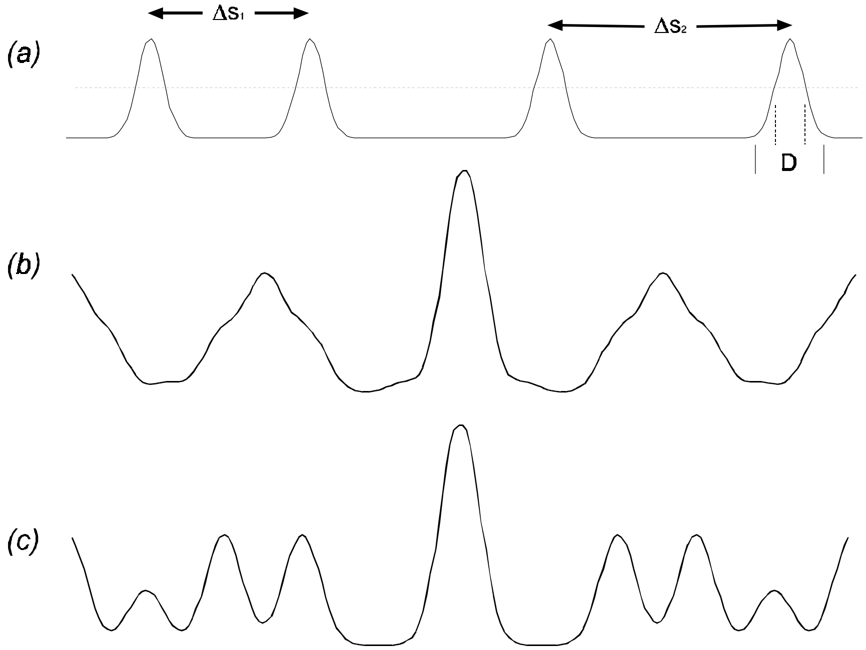

Figure 2: *Intensity plot of two tracer particle images, 2(a). The intensity profile of each of the tracer particles is Gaussian with a spot diameter that depends on the pixel dimensions, focal length, and aperture of the camera. A gradient in the flow results in unequal spacing between exposures, $\Delta s_1/\Delta s_2 \neq 1$. If the gradient is small, the correlation of the image yields the average of the particle displacements, 2(b). If the gradient is large, the average particle displacement can not be determined, 2(c). Threshold intensity compression has the effect of reducing the image spot diameter as shown by the dotted line in 2(a).*

A more robust, although slightly more computationally intensive, method of segmenting an image is to rely on the magnitude of the gradient in intensity of the pixels in the image. To reduce computational intensity, the magnitude of the intensity gradient is often approximated as the absolute value of the gradients in the *x* and *y* directions, $|\nabla I| \cong \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|$. To first order, this can be calculated as, $|\nabla I| \cong \left|I_{(i+1,j)} - I_{(i,j)}\right| + \left|I_{(i,j+1)} - I_{(i,j)}\right|$. An appropriate magnitude for the cutoff in the intensity gradient can then be selected in the same manner as it is done for intensity thresholding. Pixel intensities in regions where the gradient is sufficiently high are retained and the rest are discarded (assumed to have a value of zero). The result is the compression of an image where only the pixels around the edges of tracer particles are retained. The center of the particle images which have a low intensity gradient are discarded, Figure 3. Because of this, intensity gradient segmentation of PIV images usually results in a smaller data set then images segmented by intensity thresholding. The gradient method of segmentation is the method of choice for most bimodal images [Gonzalez, Woods, 1993]. It is, however, particularly well suited to the compression of images for correlation since it is the change in pixel intensities that allows subpixel particle displacements to be determined by correlation and not the average intensity of the particle images.

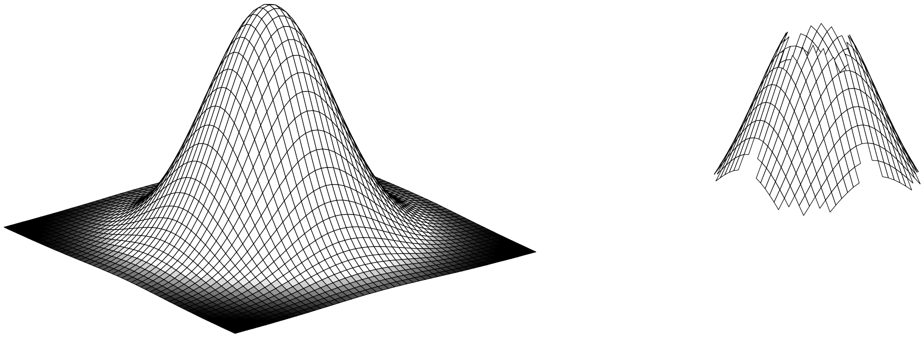

Figure 3: *Particle image intensity plot illustrating the effect of gradient pixel segmentation.*

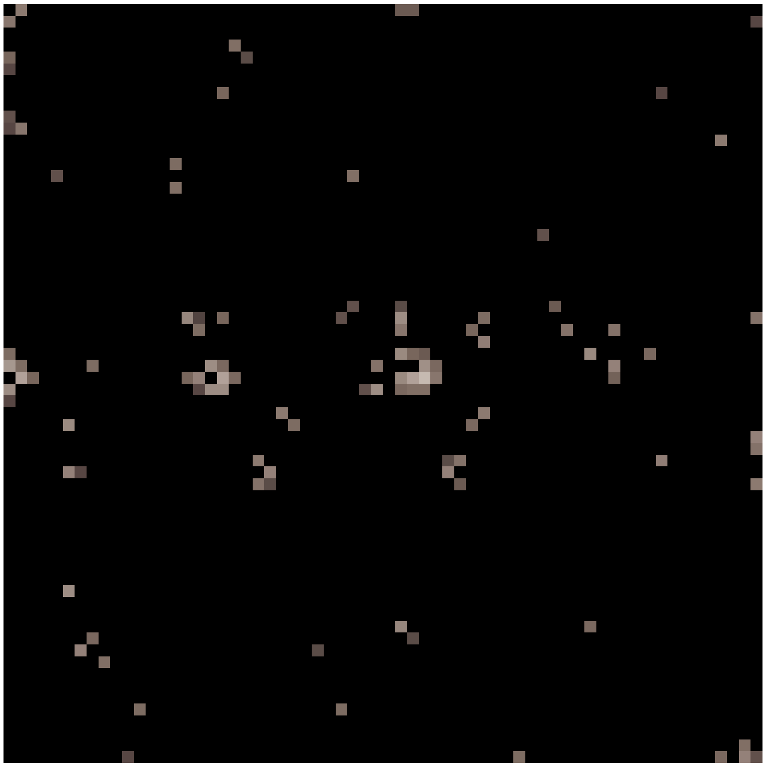

Figure 4: *Reconstructed image of Figure 1 after gradient level compression. This image has been compressed to 1/30 of its original size. Sufficient correlation information remains in this image to accurately determine particle movement even after gradient level compressing this image by a factor of 200:1. Most PIV images can be compressed 30:1 with very little loss in correlation information.*

Data Encryption

Once an image is compressed it is stored with each pixel indices and intensity combined into a single 32-bit word. This reduces the number of memory calls that must be made when correlating. For example, ***i***=2, ***j***=2, ***I***=254 is stored as 00000000001000000000001011111110 *binary* = 2,097,918. By masking the bits, the values of ***i***, ***j***, and ***I*** can be extracted from this single entry in a few clock cycles of most processors.

Along with the sparse image array, an indices table is generated which contains the location in the sparse image array of the first entry representing a pixel combination in the next line of a PIV image. This line index array is used to jump to the next value of ***j*** in the sparse image array when a specified pixel separation is exceeded in the *i*th direction. When correlating large images, this index array significantly speeds processing.

Window Sorting

The reduction in the number of data entries in the PIV image data set by the elimination of pixels in regions with a low intensity gradient and the encoding of the remaining data greatly improves the speed at which correlation windows can be sorted from the data set. In addition, the line index array reduces the number of multiple entries into the sparse image array that must be made to extract the pixels located in a given correlation subwindow. Despite this, window sorting is a slow memory intensive task that requires considerable processing time. The present algorithm requires almost as much time to sort the correlation subwindows from the image data as it does to correlate the subwindows once they have been sorted.

Correlation window sorting in sparse array format is considerably more difficult than it is in an uncompressed format since the spacing of the data entries is image dependent. A simple block transfer as is commonly done in an uncompressed format cannot be done in the sparse array format. A solution to this is to generate the sparse array at the same time that the correlation windows are being extracted from the image. This technique works well, as long as there is no significant overlap of the correlation windows. If there is significant overlap, the number of redundant memory calls greatly slows processing. The most computationally efficient technique is to presort all of the correlation windows as the sparse array is generated. This technique requires a significant increase in memory storage depending on the overlap in the correlation windows. A 50% overlap results in a four times increase in memory storage. The 32-bit sparse array data encryption scheme, itself, requires four times the number of bits per pixel. Therefore, there is an increase in memory storage requirement by a factor of sixteen. Image compression, however, sufficiently reduces the number of data entries such that there is a net reduction in data storage by roughly a factor of four compared with storing the entire image in memory at one time. In addition, presorting the windows in this manner moves the processing time for window sorting from the basic correlation algorithm into the image-preprocessing algorithm. This allows more time for image correlation within the 1/30 of a second video framing speed. Presorting the correlation subwindows at the same time the image is compressed is, therefore, the optimum solution in the majority of applications.

Search Length Selection

Processing speed can be further increased while, at the same time, reducing the odds of obtaining spurious correlation values by limiting the search for a maximum correlation. This is done by allowing the user to specify a maximum change in $\Delta\boldsymbol{i}$ and $\Delta\boldsymbol{j}$ based on knowledge of the image being correlated. An adaptive scheme can be used to narrow the correlation search - a scheme that predicts the range of correlation values to calculate based on previous calculations from subwindows of the same image. This procedure, however, is not particularly robust and can result in spurious errors in obtaining the maximum correlation. Because the sparse array correlation process is inherently very fast, adaptive schemes generally do not gain enough processing speed to warrant their use. It is sufficient to set a single value for the correlation range for an entire image.

Subwindow Correlation

By using the error correlation function rather than a statistical correlation function, image correlation can be carried out using integer addition and subtraction only. These are very fast operations for most microprocessors requiring only a few clock cycles. It is far faster to perform these calculations than to use a 'look-up table' scheme to avoid 8-bit or 4-bit pixel multiplication. The use of the error correlation function, therefore, significantly improves processing speed over the more commonly used statistical correlation function. A detailed analysis of the error correlation function in comparison to the statistical correlation function is presented in a paper by Roth, Hart, and Katz (1995). It was shown that the error correlation function produces essentially the same results as the more computationally intensive statistical correlation function.

The error correlation function can be expressed as,

$$\Phi_{\Delta i,\Delta j} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[I_{m,n} + I_{m+\Delta i,n+\Delta j} - \left|I_{m,n} - I_{m+\Delta i,n+\Delta j}\right|\right]}{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[I_{m,n} + I_{m+\Delta i,n+\Delta j}\right]}$$

such that,

$$\Phi_{\Delta i,\Delta j} = 1 - \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[\left|I_{m,n} - I_{m+\Delta i,n+\Delta j}\right|\right]}{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[I_{m,n} + I_{m+\Delta i,n+\Delta j}\right]}$$

The value of this correlation function ranges from 1 when the images are perfectly correlated to 0 when there is no correlation between the images. Because it relies on the difference in pixel intensities, it does not unduly weight the significance of high-intensity pixels as does the statistical correlation function. Aside from being faster to calculate than the statistical correlation function, it has the added benefit of being easier to implement in hardware without the need for a microprocessor. The error correlation function, therefore, has potential for use in hardware based PIV systems.

Unlike the more common statistical correlation function, the error correlation function used in sparse array image correlation is not computed one entry at a time. The entire correlation table is constructed by summing entries as they are found while iterating through the sparse image array. When auto-correlating subwindows, each entry in the sparse image array is compared with the entries below it and a correlation approximation between the entries is added into the correct location in the correlation table based on the difference in ***i*** and ***j*** between the array entries. If the location is out of range of the specified search length in the ***i***th direction, the entry is ignored and processing continues with the next entry specified in the line index array. If the location is out of range in the ***j***th direction, the entry is ignored and a new series of iterations are made starting with the next sparse image array entry. Because the sparse array is correlated from the top down, only the half of the correlation table representing the positive ***j*** direction is calculated. The auto-correlation of an image is symmetrical and thus, calculation of both halves of the correlation table is unnecessary.

Cross-correlation is accomplished by generating two sparse image arrays representing the two images being correlated. The entries of one array are then compared to *all* of the entries of the other array that are within the search length. Because the difference in array indices can be both positive and negative in the ***i*** and ***j*** directions, the entire non-symmetrical correlation table is calculated.

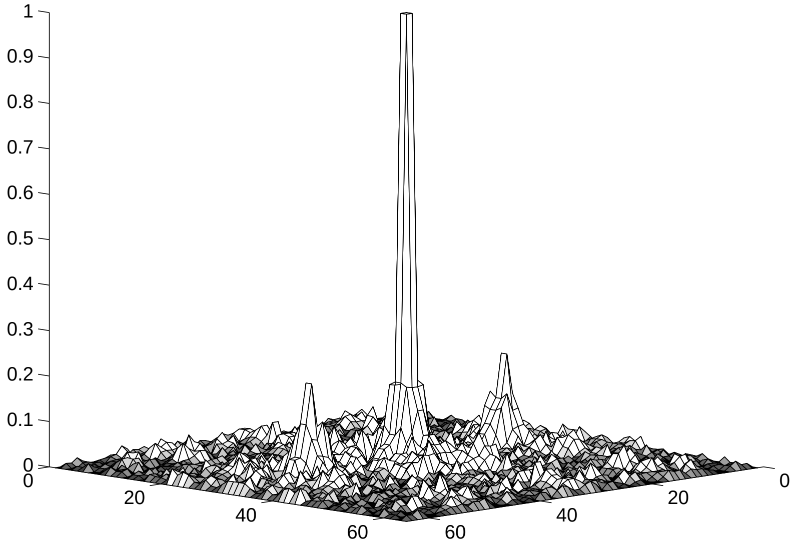

Figure 5: *Correlation table resulting from sparse array auto-correlation of the image in Figure 4. In practice, the sparse array image data set is correlated from the top down so that only half of the correlation table is calculated. Both halves of the symmetrical correlation table are shown here for clarity. Note the steep correlation peaks. These sharp peaks aid subpixel interpolation.*

Once the correlation table is complete, the table is searched for the maximum correlation value. A simple bilinear interpolation scheme is then used to determine the correlation maximum within subpixel resolution. Bilinear interpolation is ideal in this application since reducing the data set by image preprocessing and using the error correlation function results in a very steep, nearly linear, correlation peak.

3. PROCESSING SPEED

Computational Intensity

The computational intensity of sparse array image correlation is comparable to the better known statistical correlation technique except that the image data set is compressed in preprocessing. If the data set is reduced to a fraction, γ, of the original image data set, than the number of data comparisons that must be made is, $\frac{1}{2}\gamma\Delta^2(\gamma N^2 - 1) + \gamma N^2$ for sparse array auto-correlation and $\gamma^2\Delta^2 N^2$

for cross-correlation. For PIV images where the particle seeding densities are high such that, $\gamma N^2 >> 1$ and $\gamma\Delta^2 >> 1$ then $\frac{1}{2}\gamma\Delta^2(\gamma N^2 - 1) + \gamma N^2$ is approximately equal to, $\frac{1}{2}\gamma^2\Delta^2 N^2$. A typical PIV data set can be reduced by a factor of 30 such that γ=0.3. Thus, a typical 64*x*64-pixel correlation subwindow requires a little less than one thousand data comparisons to complete an auto-correlation with a search window of 20*x*20 pixels. During each comparison, three memory calls are made, one to retrieve a data entry to be compared with the data entry already in the processors register, one to retrieve the value of the correlation table entry, and one to place the comparison result in memory. Memory calls require a great deal more processing time than integer addition and subtraction so that the time for each data entry comparison is essentially the time it takes to make these memory calls [Hennessy & Patterson 1990]. PCI based systems can transfer over 60*Mbytes* of data per second or about two million 32-bit data entries per second over the bus. By ordering data entries sequentially when extracting the correlation subwindows from the image data set, bus transfer rates of this speed can be achieved by block memory transfers. Thus, correlation speeds of 2,000 *vec./sec.* are theoretically possible for typical PIV images under these conditions.

Speed Relative to Spectral FFT Correlation

FFT spectral correlation is known to be a computationally efficient method of PIV processing. It is accomplished by taking the two-dimensional Fourier transform of an image and multiplying it by the complex conjugate of the Fourier transform of another image (or the same image in the case of auto-correlation) before taking the inverse transform. The computational intensity of this method of image correlation is $3N^2\log(N) + N^2$. It is thus, correctly referred to as an $N^2\log(N)$ correlation algorithm. In comparison, sparse array image correlation is an N^2 algorithm for a fixed correlation search length, Δ. It is, therefore, faster than FFT spectral correlation as long as $(\Delta\gamma)^2$ is smaller than $3\log(N)+1$, Figure 6. At low compression ratios, $\gamma \cong 1$, FFT spectral correlation is far faster than sparse array correlation for any reasonable correlation search length. At compression ratios greater than $\Delta/2$, however, sparse array image correlation results in significant computational savings, Figure 7.

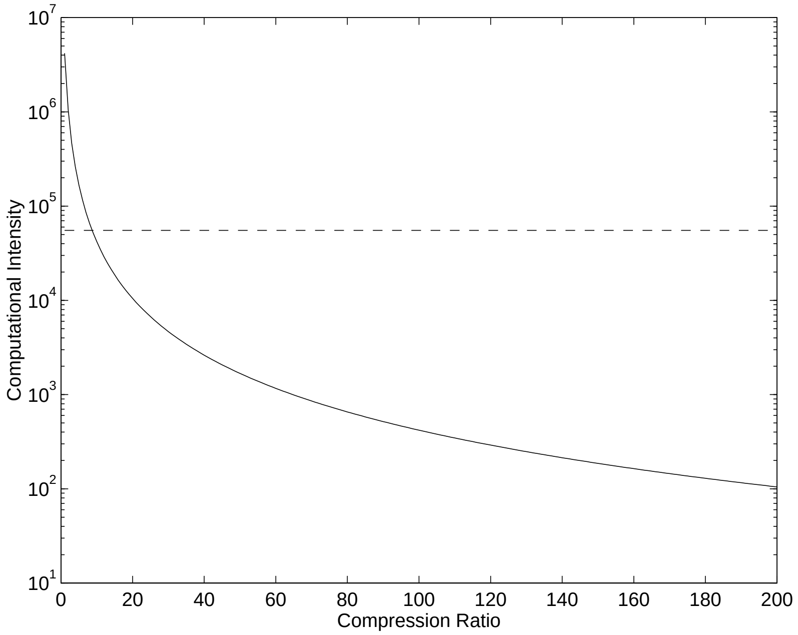

Figure 6: *Sparse array cross-correlation processing computational intensity (solid line) relative to spectral correlation (dotted line) as a function of image compression ratio for a 64x64 pixel correlation subwindow. Note that this is a semi-log plot. The computational intensity of sparse array cross-correlation for densely seeded images is* $\gamma^2\Delta^2\mathrm{N}^2$ *where FFT spectral correlation is* $3\mathrm{N}^2\log\mathrm{N} + \mathrm{N}^2$.

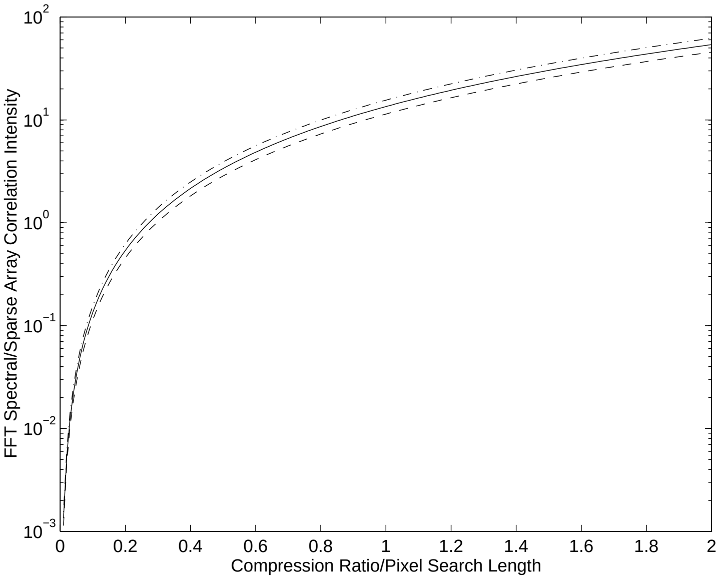

Figure 7: *Semi-log plot of the computational processing intensity of FFT spectral correlation relative to sparse array correlation as a function of the compression ratio divided by the search length. Plots of N=32, 64, and 128 are shown. Above* γ/Δ *of about 0.25, spectral correlation is far more computationally intensive than sparse array correlation.*

4. ACCURACY AND ROBUSTNESS

The process of correlating images in sparse format using the algorithm presented here is independent of the method by which the image data set is generated. With no compression of the image, sparse array image correlation is *identical* to the more common statistical image shifting correlation method except for the use of the error correlation function. A comprehensive study of the error correlation function by Roth *et. al.* showed that there exists no significant variation between the results from the error correlation function and that of the statistical correlation function [Roth, Hart, Katz, 1995.]. There is, thus, no reason to believe that the error correlation function is any more or less accurate than statistical or spectral correlation techniques. Any inaccuracy or lack of robustness in the present algorithm can be attributed entirely to the loss of data from image compression. The speed of sparse array image correlation, however, is strongly dependent on the reduction in the image data set through compression. Little is gained by using this algorithm if the data set remains unchanged, Figure 6. It is therefore necessary to address the problems associated with image compression to assess the limitations of the sparse array image correlation algorithm.

Velocity Gradient Affects

A method of determining the probability that a particular correlation is valid is to perform a non-parametric correlation and observe the peak correlation value relative to the mean. This can be accomplished by ranking the pixels in an image before correlation. Pixels with the same intensity are assigned an average of the rank they would receive if they had different values. In this manner, non-parametric correlation provides a means by which the effects of image compression can be assessed. Images with a poor rank correlation value relative to the mean are more likely to produce spurious vectors and to lack information needed to obtain accurate subpixel resolution. Thus, non-parametric correlation provides, in essence, a measure of the correlation signal to noise ratio.

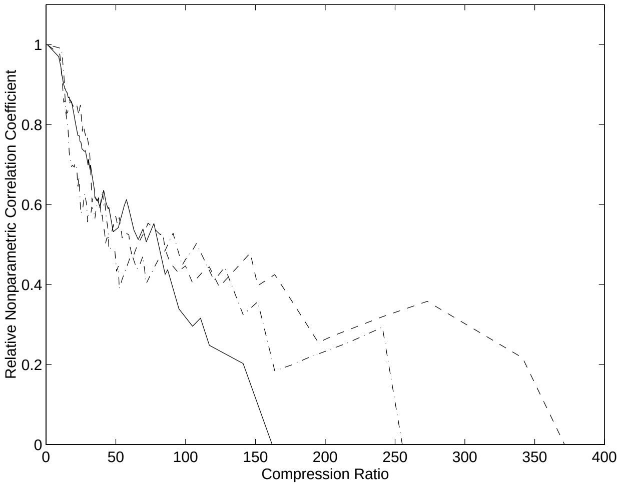

Figure 8: *Relative non-parametric correlation coefficient plotted as a function of intensity threshold image compression for G_{∇} =0, 1, and 2. The correlation coefficient is based on non-parametric rank correlation and is normalized relative to the uncompressed image. As the imaged flow divergence increases, the compression ratio at which the correlation is lost, indicated by a zero relative non-parametric correlation coefficient, decreases.*

As discussed in Section 2., the particle image diameters are important for resolving the average particle displacement in flows where there exist large gradients in the flow velocity. This is the factor which image compression affects the most. Other parameters such as seeding density, average flow velocity and relative correlation window size that influence all PIV correlation processing are unaffected. Using rank correlation, the effects of intensity threshold compression are plotted in Figure 7 for several values of the ratio of the flow divergence to the particle image diameter based on the correlation window diameter, *N*, $G_\nabla = \frac{M|\nabla \cdot \bar{v}|N\Delta t}{D}$. For *ideal* PIV images where there exists no divergence in the flow, $G_\nabla = 0$, and no variation in peak image intensity, the compression ratio has little effect on the relative correlation of the image until a significant portion of the data of the tracer particle images are eliminated. These images can be compressed to a small fraction of their original size and retain enough correlation information to determine the subpixel displacement of the tracer particles. As shown in Figure 8, images where G_∇ does not equal zero are affected to an increasingly greater extent by compression as G_∇ increases.

Intensity Variation Effects

Variations in intensity resulting from poor illumination and/or variations in particle characteristics severely affect the ability to extract particle displacement information from a PIV image using correlation. This is particularly true of auto-correlation processing, as there often exists a systematic intensity variation between the first and second exposures of the particle images. This type of intensity variation adversely affects both spectral correlation and compressed image correlation processing. It, however, limits the level to which an image can be compressed and thus has a much more pronounced effect on the speed and accuracy of compressed image correlation. Non-systematic intensity variations, which result from differences in tracer particle characteristics and non-uniform illumination, affect both spectral correlation and compressed image correlation to roughly the same degree.

Consider the Gaussian intensity profile of a tracer particle image. This profile can be approximated by $I_r \cong I_o e^{-(4.3r/D)^2}$ where I_o is the peak intensity of a particle centered at (x_o,y_o) and *r* is the distance from the center. The magnitude of the gradient in intensity is then equal to $|\nabla I_r| \cong \frac{37rI_o\beta}{D^2}e^{-(4.3r/D)^2}$ where β is the characteristic size of a single pixel in the image. The maximum magnitude of the intensity gradient of a particle image occurs at a distance *r*=*D*/4.3 and has a value $\left|\nabla I_{r=D/4.3}\right| \cong 8.6\frac{I_o\beta}{D}$. If an intensity gradient threshold level is set above this value, all correlation information for this particle will be lost. Note, however, that the minimum particle diameter is roughly *D*/2 as long as the intensity gradient threshold level is set below $8.6\frac{I_o\beta}{D}$. This is not true of image segmentation based on intensity level thresholding where the particle image diameter approaches zero as the threshold level is increased. If there are significant variations in particle peak intensities within an image, however, then both methods of image segmentation adversely affect correlation although intensity thresholding to a somewhat less extent for the same compression ratio. This is illustrated in Figure 9 by plotting the non-parametric correlation peak value obtained from Figure 1 as a function of compression ratio for both gradient level compression and threshold compression. As illustrated in this figure, threshold intensity compression results in less information loss at low compression ratios. At higher compression ratios, however, gradient intensity compression results in less information loss.

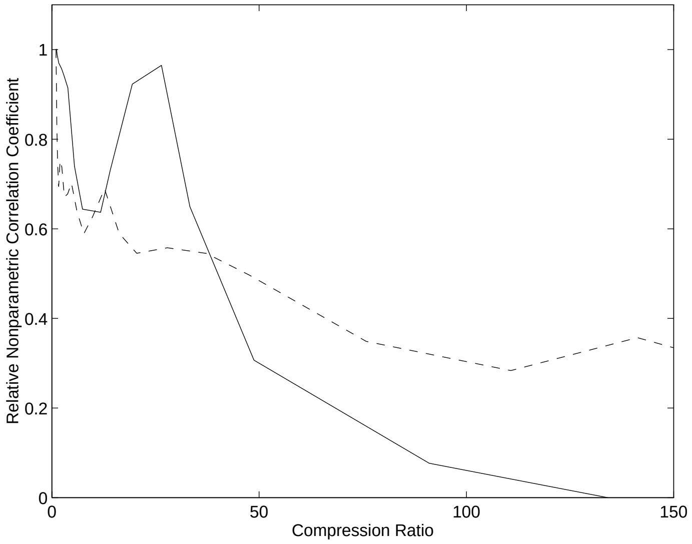

Figure 9: *Relative non-parametric correlation coefficient plotted as a function of intensity gradient image compression (dotted line) and intensity threshold compression (solid line) for the image in Figure 1. Intensity threshold image compression results in less information at compression ratios below 40:1. At higher compression ratios, gradient image compression results in less information loss allowing the image to be compressed to less than 1/200 of its original size before particle displacement information is lost. Threshold compression, however, losses the particle displacement information at a compression ratio of less than 150:1.*

In practice, exposure levels are difficult to control and image-recording devices have limited intensity resolution. PIV images that have been over exposed have *tophat* particle image intensity profiles, Figure 10. While some information is lost because of this, these images often correlate accurately for the same reason images compressed using intensity gradient thresholding correlate accurately – it is the change in intensity at the edges of the particle images that hold the information necessary for accurate correlation and not the low gradient regions near the center of the particle images. PIV images that are slightly over exposed are often better suited to intensity gradient compression rather than intensity threshold compression. This is because the saturated intensity regions of the particle images that contain little correlation information and have a low intensity gradient are eliminated from the data. Because intensity gradient image compression is more robust to variations in image exposure, it is generally a better choice for PIV image compression even though it results in slightly more information loss at moderate compression ratios.

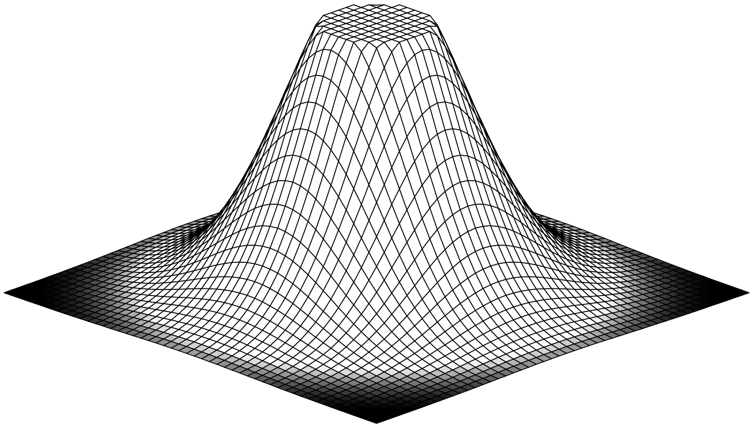

Figure 10: *Tophat profile of a particle image which has reached the saturation level of the image recording media. PIV images that contain particle images of this type can be significantly compressed without losing correlation.*

5. EXPERIMENTAL DEMONSTRATION

Variations in particle image intensity and size, correlated background noise, poor contrast, insufficient illumination and optical aberrations are only a few of the many factors which effect the quality of experimental images. For this reason, a comparison was made between spectral correlation and sparse array correlation based on the processing of experimentally obtain PIVC images taken of a highly unsteady vortical flow.

Images were used from the experimental measurement made of the flow inside a Cardio Assist Device, CAD. This device is used to aid the flow of blood in patients with weakened hearts. The flow in this device is highly unsteady and exhibits strong vortical flow formations [Huang, Hart, Kamm 1997]. A frequency doubled Nd:YAG laser was used to illuminate a 10*cm x* 10*cm* area. The flow was seeded with 50μm florescent particles. A Pulnex, TM-9701 512x486 pixel CCD camera, recorded PIVC images at 30*Hz*. Typical images are shown in Figure 11. Because of the curvature of the wall, all of the images exhibit significant variations in light intensity. These images were specifically chosen for comparison with the spectral correlation method because they exhibit features that are poorly suited to processing in sparse format. These features include significant local variation in illumination, large gradients in the flow velocity, heavy seeding densities, and very small tracer particle movement between images (less than 1 pixel on average) requiring accurate subpixel interpolation to resolve flow structures. Because of these features, the test images provide a means of illustrating the limitations of sparse array image correlation.

The experimental images were processed by cross-correlation using 64*x*64 pixel subwindows that overlapped by 50% in both the *x* and *y* directions. When compressed 30:1 with a maximum correlation search length of 32 pixels, the sparse array algorithm processed these images at roughly 300 *vec./sec.* on a Pentium 166*MHz* computer with 16*Mbyte*s of memory. This was about sixty times faster than spectral correlation, which generated 5 *vec./sec.* on the same machine. An example of the output of both sparse array correlation and of spectral correlation of the images in Figure 10 are shown in Figure 12. Sparse array correlation yielded results that were typically within 0.05 pixels of spectral correlation. No significant variation between the two correlation algorithms was observed with image compression ratios below 50:1. At higher compression ratios, 100:1, differences in the velocity profile in low velocity areas were observed, Figure 13. These variations, on the order of 0.05 pixels, are the result of information loss due to compression. At much higher compression ratios, 200:1 and higher, sparse array image correlation generated significant spurious vectors near the wall of the test section and in other regions where high velocity gradients exist, Figure 14. This behavior is consistent with the analysis discussed in Section 4. At these extremely high compression ratios, each vector represents the correlation of less than thirty pixels.

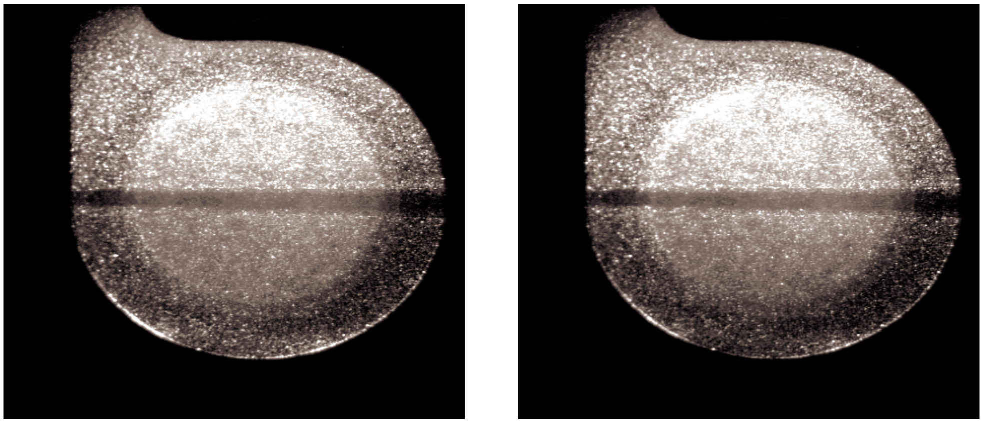

Figure 11: *Typical pair of PIV images recorded from a Cardio Assist Device, CAD. The light intensity in these images varies significantly due to the distortion caused by the three-dimensional shape of the device. Such images that exhibit significant changes in velocity and have high seeding densities are difficult to correlate in compressed format. Because of this, these images were selected to illustrate the limitations of sparse array image correlation processing.*

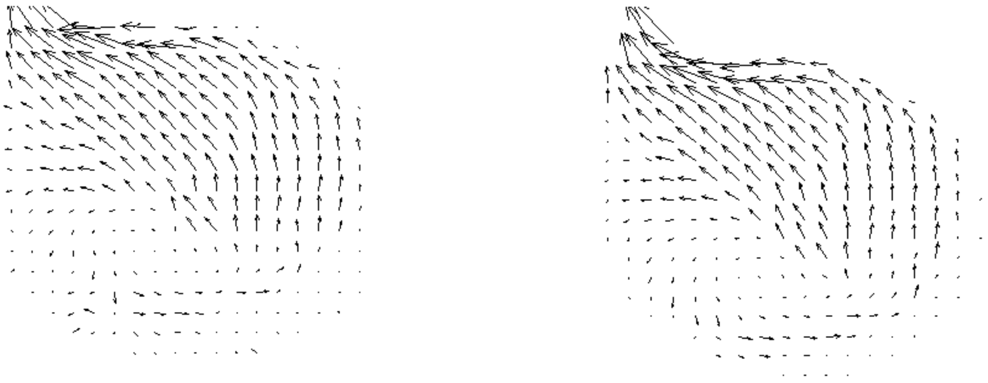

Figure 12: *Vector map obtained from sparse array cross-correlation of the images in Figure 10 with a 30:1 compression ratio (left) and by spectral correlation (right). The longest vector in these plots represent a displacement of less than 2 pixels. The majority of vectors shown represent displacements of tracer particle images of less than 1 pixel. Thus, subpixel interpolation is critical to visually resolving flow structures from these plots. Variations of less than 0.05 pixels are easily observed.*

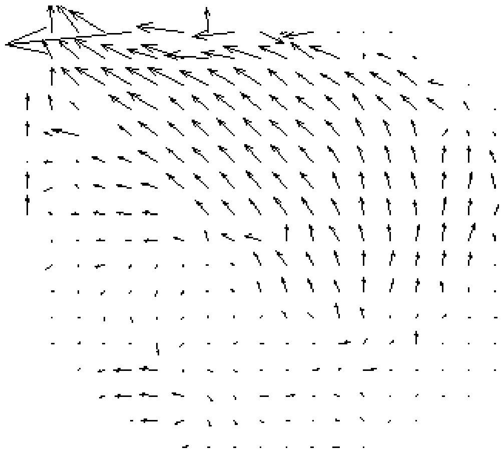

Figure 13: *Vector map obtained from sparse array cross-correlation of the images in Figure 10 with a 100:1 compression ratio. Compare this figure to the plots in Figure 11. Note the difference in areas where there exist low x and/or y velocities. This is a result of information loss from image compression limiting subpixel resolution.*

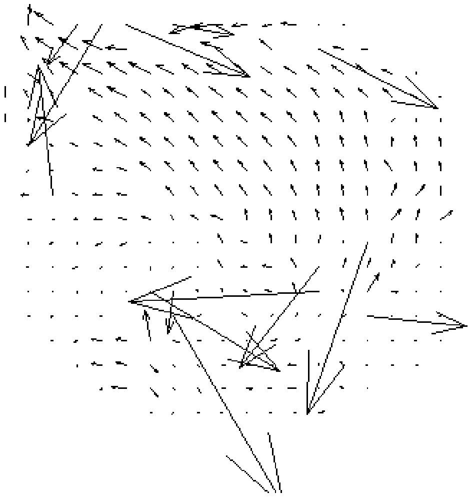

Figure 14: *Vector map obtained from sparse array cross-correlation of the images in Figure 10 using a 200:1 compression ratio. Note the spurious vectors near the wall and in high velocity gradient regions. This is consistent with the processing limitations discussed in Section 4. At this high compression ratio, each vector represents a correlation of less than 30 pixels.*

6. SUMMARY AND CONCLUSIONS

Sparse array image correlation is a technique by which PIV images can be accurately processed at high-speeds. It is based on the compression of images in which the number of data set entries containing tracer particle displacement information is reduced. Very high correlation speeds are obtained by encrypting the reduced data set into 32-bit integers and correlating the data entries using an error correlation function to eliminate multiplication, division and floating point arithmetic.

The maximum correlation value associated with sparse array image correlation is characterized by a steep peak that improves subpixel interpolation. The performance of this method of image correlation, however, is largely dependent on the level to which an image can be compressed without losing significant correlation information. Thus, its performance relative to the better known spectral correlation method is image dependent.

Through an analysis of the affects of flow divergence, tracer particle image diameter, and intensity variations, it was shown that typical PIV images can be highly compressed with no significant lose in correlation information. Characteristic limitations of sparse array image correlation were illustrated by comparing results from the spectral correlation of experimental images with the results from the sparse array correlation of the same images at varying levels of compression from 30:1 to 200:1. For applications requiring extremely high correlation speeds such as holographic particle image velocimetry (HPIV) and video rate particle image velocimetry cinematography (PIVC), sparse array image correlation appears to be a viable processing technique.

REFERENCES

Adrian, R. J. (1986) “Multi-point Optical Measurement of Simultaneous Vectors in Unsteady Flow—a Review.” *Int. J. Heat and Fluid Flow,* pp. 127-145.

Adrian, R. J. (1991) “Particle Imaging Techniques For Experimental Fluid Mechanics.” *Annual Review of Fluid Mechanics",* Vol. 23, pp. 261-304.

Gonzalez, R. C., Woods, R. E. (1993), Digital Image Processing, Addison-Wesley Pub. Co., Reading, MA., pp. 414-456.

Hatem, A.B., Aroussi, A., (1995), "Processing of PIV Images." SAME/JSME and Laser Anemometry Conference and Exhibition, August 13-18, Hilton Head, South Carolina, FED-Vol. 229, pp.101-108.

Hennessy, J.L., Patterson, D.A., (1990), “Computer Architecture - A Quantitative Approach.” Morgan Kaufmann Pub., San Mateo, CA.

Huang, H., Hart, D. P., Kamm, R. (1997), “Quantified Flow Characteristics in a Model Cardiac Assist Device“, ASME Summer Annual Meeting, Vancouver, B.C., Experimental Fluids Forum, pp. .

Landreth, CC. and Adrian, R.J. (1987) "Image Compression Technique for evaluating Pulsed Laser Velocimetry Photgraphs having High particle Image Densities", FED-Vol. 49.

Okamoto, K., Hassan, Y. A., Schmidl, W. D. (1995), “New Tracking Algorithm for Particle Image Velocimetry.” *Experiments in Fluids,* pp. 342-347.

Rosenfeld, A., De La Torre, P., (1983), “Histogram concavity analysis as an aid in threshold selection,” *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 13(3), pp. 231-235.

Roth, Hart, and Katz, (1995) ”Feasibility of Using the L64720 Video Motion Estimation Processor (MEP) to increase Efficiency of Velocity Map Generation for Particle Image Velocimetry (PIV).” ASME/JSME Fluids Engineering and Laser Anemometry Conference, Hilton Head, South Carolina, pp. 387-393.