AIAA JOURNAL
Vol. 40, No. 6, June 2002

Comparative Study of Image Compression Techniques for Digital Particle Image Velocimetry

William M. Humphreys Jr.*
NASA Langley Research Center, Hampton, Virginia 23681
and
Ahmed M. Naguib†
Michigan State University, East Lansing, Michigan 48824

A comparative, quantitative study of image compression techniques for use with digital particle image velocimetry has been performed. Several candidate compression algorithms were selected for the study, including a lossless technique and two mathematical transform-based methods. Each of the compression algorithms was implemented using commercial off-the-shelf software packages. Three image sequences were selected to exercise the various compression methods. These sequences included a set of industry standard images and two sets of images obtained from experimental work conducted at NASA Langley Research Center. Evaluation of the various methods was accomplished using quantitative perceptual and metrological performance measures. The results of the study indicate that several of the tested methods of compression are suitable for digital particle image velocimetry. A lossless LZ77 technique, coupled with pixel thresholding of image gray levels before compression, yielded excellent performance in terms of compression level and negligible introduction of spatial errors to the images. A lossy JPEG algorithm was shown to provide acceptable performance; however, significant spatial errors and increased numbers of false vectors derived from processing of the compressed images were observed at high compression levels. Finally, a lossy wavelet algorithm was shown to provide excellent performance in terms of minimal introduction of spatial errors and a reduction in the false vector rate over a wide range of compression levels.

Nomenclature

B_{tot}	=	total bits in original image
$\tilde{B}_{\text{tot}}$	=	total bits in compressed image
E_u	=	spatial error estimate of individual u component vector, % [see Eq. (8)]
E_v	=	spatial error estimate of individual v component vector, % [see Eq. (8)]
$f(i, j)$	=	intensity of ith, jth pixel of original image
$f'(i, j)$	=	intensity of ith, jth pixel of thresholded image
$\tilde{f}(i, j)$	=	intensity of ith, jth pixel of compressed image
L_g	=	maximum gray levels in image
M, N	=	number of horizontal and vertical pixels, respectively, in image
m	=	decomposition levels in discrete wavelet compression
P, Q	=	number of horizontal and vertical displacement vectors, respectively
$u(i, j)$	=	u displacement component obtained from processed original image, pixels
$\tilde{u}(i, j)$	=	u displacement component obtained from processed compressed image, pixels
$v(i, j)$	=	v displacement component obtained from processed original image, pixels
$\tilde{v}(i, j)$	=	v displacement component obtained from processed compressed image, pixels
$\sigma_{E_u}, \sigma_{E_v}$	=	standard deviation of spatial error estimate histograms
$\phi(x)$	=	recursive wavelet basis function [see Eq. (3)]

Introduction

DIGITAL particle image velocimetry (DPIV) has become an accepted technique for the measurement of two- and three-component planar velocities in a wide variety of fluid flows, and several good tutorials and reviews outlining advances in the state of the art of the technique have appeared recently.[1-4] Over the past several years, significant improvements have been achieved in DPIV acquisition hardware with the introduction of high-resolution, large-format cameras that are capable of acquiring images at speeds in excess of 10 frames per second. Associated with the introduction of these new camera technologies, stereo DPIV systems that incorporate as many as four separate cameras[5] have recently been described in the literature. With the increase in the number of cameras employed in these systems, as well as the increase in the pixel count per camera, the issues of image storage and management become relevant. For instance, the authors recently completed a series of DPIV tests in the NASA Langley Subsonic Basic Research Tunnel that required the use of four separate 1300 × 1030 pixel cameras connected to 8-bit digitizing frame grabbers. During four days of testing, 32,800 separate images were acquired. These images consumed approximately 41 GB of hard drive storage space and were archived on 68 CD-ROM disks. For a typical turbulence study where acquisition of several thousand images is required for a single flow condition, the total storage space needed for these images can quickly become problematic. In addition, new technologies are forthcoming that will enable high-speed DPIV acquisition at rates surpassing 1000 frames per second. The cost of storage media has rapidly diminished over the past decade; nevertheless, there remains a need to achieve maximum efficiency in the archival and retrieval of vast numbers of acquired images.

One established method for improving the efficiency of handling large numbers of images involves compression of the image data during storage. A large number of books and papers have been written describing and contrasting various methods of image compression. Classically, studies characterizing the performance of various compression methods have concentrated on qualitative measures of

Presented as Paper 2001-0695 at the AIAA 39th Aerospace Sciences Meeting, Reno, NV, 8–11 January 2001; received 27 February 2001; revision received 4 October 2001; accepted for publication 19 October 2001.

*Research Engineer, Advanced Measurement and Diagnostics Branch. Senior Member AIAA.

†Assistant Professor, Mechanical Engineering Department. Member AIAA.

performance based on human visual perception, a perfectly acceptable criteria given that a majority of image processing algorithms are designed to provide outputs that are viewed by humans. With a few exceptions (most notably astronomical imaging), less attention has been paid to the consequences of using compressed images in metrological applications such as DPIV. In particular, the effect of image compression on postprocessed data accuracy has not been extensively addressed. Because DPIV technology is now being applied in numerous laboratory and industrial settings to collect large numbers of images, a few groups have begun to look at the implications of using compression techniques to reduce the storage requirements for DPIV data. Several papers describing techniques compatible for use in DPIV acquisition have been presented in the literature over the past two years. In particular, Li[6] presented a good examination of various wavelet methods in relation to DPIV image denoising and information retention. Freek et al.[7] examined the accuracy of using JPEG compression on sequences of synthetic DPIV images. Cenedese et al.[8] examined using two different compression techniques, namely, the JPEG algorithm and a lossless technique applied to binary versions of the original DPIV images. These studies represent advances in understanding the effects of applying image compression to DPIV. However, a comprehensive study using a number of quantifiable performance measures characterizing the measurement uncertainty in the displacement vector maps derived from analysis of compressed DPIV images has not been performed. Also, the implications for choosing a lossless vs a lossy, that is, loss of information, compression algorithm have not been thoroughly examined.

To complement and extend the previous work given in Refs. 6–8, a comparative yet quantitative study has been conducted of several popular image compression techniques, both lossless and lossy, with regard to their effect on the accuracy of DPIV-derived displacement vector maps. Several key performance measures based on compression rates, signal-to-noise levels, and spatial distortions were chosen to evaluate each candidate technique. Three different sequences of DPIV images, two experimentally obtained and one an industry standard sequence, were chosen to exercise each candidate. This paper presents brief descriptions of the candidate compression techniques, performance measures chosen to evaluate each candidate, and some representative results comparing the performance of the various techniques.

Candidate Compression Algorithms

This paper describes one lossless, one modified lossless, and two lossy image compression techniques chosen for this study. The lossless technique examined was the Lempel–Ziv 77 (LZ77) dictionary-based compressor (see Ref. 9). This algorithm was included in the study to provide baseline lossless compression rates, as well as to establish guidelines for determining when use of a purely lossless technique may be preferable. The modified lossless technique consisted of the LZ77 algorithm coupled with gray level thresholding of the images before compression. This technique is somewhat similar to that described by Cenedese et al. in Ref. 8. However, Cenedese et al. created a binary image after thresholding, whereas the technique used for this study consists of a simple gray-level truncation technique that maintains an 8-bit image. The lossy techniques that were examined are both based on mathematical transformation equations and include the discrete cosine transform and the discrete wavelet transform. Each of these algorithms, implemented for this study using commercial off-the-shelf software packages, is described subsequently in more detail.

LZ77 Compression

The LZ77 lossless compression algorithm was first described by Ziv and Lempel[9] in 1977 and is often referred to as a sliding window coding algorithm. Variations of the LZ77 algorithm are incorporated into popular programs such as PKZIP and LHarc.[10] The algorithm encodes incoming data by maintaining the last n bytes of the data as a dictionary buffer. The length of the buffer is user defined and is typically fixed in the range of 2048–16,384 B. When an incoming block of data matches part of the dictionary buffer, three values are sent to a compressed output file (which also contains the dictionary buffer), consisting of the matching position in the buffer, the matching length, and the byte or character following the match in the buffer. Compressed files are reconstructed via a table look-up procedure using the dictionary buffer and pattern matching data. The LZ77 algorithm provides for fast compression and decompression. However, nontextual data, for example, binary image files, typically cannot be compressed more than approximately 30% using LZ77 due to the relatively short matching sequences of bytes in the data.

To overcome the limited image compression capability of the LZ77 technique, a modified algorithm is proposed that couples the LZ77 compressor to a thresholding process. Before compression, image pixels are gray level thresholded using a simple truncation formula:

$$f'(i,j) = \begin{cases} f(i,j) & \text{if} \qquad f(i,j) \geq \text{ threshold} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $f(i,j)$ represents the ith, jth original pixel value and $f'(i,j)$ represents the corresponding thresholded value. The benefit of performing this operation before compression derives from the structure of a typical DPIV image. Such images are composed of small groups of illuminated pixels superimposed on essentially a black background dominated by noise. By careful choice of the threshold level, the background can be forced to zero, thereby increasing the matching length between sequences of pixels and the LZ77 dictionary buffer. This can dramatically increase the compression rate over using LZ77 alone. Although not a lossless technique in the strict sense, the method can still be thought of as a pseudo lossless technique. As will be seen, if the threshold level is carefully chosen, increased compression can be achieved with no change in the accuracy of the DPIV processed vector displacement data as compared with use of noncompressed images.

JPEG Compression

The Joint Photographic Experts Group (JPEG) compression standard is the leading technique for use in numerous imaging applications[11] and is incorporated into a number of commercial DPIV processing systems. The algorithm is based on the two-dimensional discrete cosine transform (DCT) pair:

$$\text{DCT}(x,y) = \frac{1}{\sqrt{2N}} C(x)C(y) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \times \cos\left[\frac{(2i+1)x\pi}{2N}\right] \cos\left[\frac{(2j+1)y\pi}{2N}\right]$$

$$f(i,j) = \frac{1}{\sqrt{2N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(x)C(y)\,\text{DCT}(x,y) \times \cos\left[\frac{(2i+1)x\pi}{2N}\right] \cos\left[\frac{(2j+1)y\pi}{2N}\right]$$

$$C(k) = \begin{cases} \dfrac{1}{\sqrt{2}} & \text{if} \quad k = 0 \\ 1 & \text{if} \quad k > 0 \end{cases} \tag{2}$$

where $f(i,j)$ represents the value of the ith and jth pixel in the original image and DCT(x,y) represents the corresponding image transform coefficient at a spatial frequency identified by coordinates x and y. The image f is assumed to be square in Eq. (2). The popularity of the JPEG technique is partly due to the speed of the algorithm; the DCT transform shown in Eq. (2) is separable, allowing it to be efficiently implemented as a series of one-dimensional fast Fourier transforms. This is accomplished by first performing a one-dimensional transform of each pixel row in the image, followed by a transform of each pixel column.

The DCT is applied to compress a monochrome image by first breaking up the image into a series of nonoverlapping 8 × 8 pixel blocks. Each pixel block is independently transformed into the spatial frequency (wave number) domain using the DCT. Because of the small size of the pixel blocks, a very efficient table look-up implementation of the DCT can be formed that dramatically improves performance of the algorithm. After converting each block to the spatial frequency domain, the DCT coefficients are represented in integer form, and a coefficient quantizer (implemented as a series of multiplicative quantization matrices) is used to reduce the number of bits required to store each coefficient. The coefficients are then encoded using lossless run-length and entropy schemes. The use of variable quantization matrices allows the user to specify easily the amount of compression to perform, thereby providing a measure of control over the resultant image quality. However, an overly aggressive quantization matrix may cause severe degradation in compressed image quality. As the number of bits required to store the coefficients is decreased, high wave number information in the 8 × 8 pixel block is removed. The loss of too much information results in a visible checkerboard pattern in the image, commonly referred to as Gibb's phenomenon. Gibb's phenomenon is a major limiting factor in application of high JPEG compression rates. In particular, for cases where a DPIV particle image crosses an 8 × 8 pixel block boundary (which is quite likely), Gibb's phenomenon manifests itself as a change in the spatial distribution of the particle image during compression. This spatial change imparts an error to the displacement vector derived from analysis of the particle image.

Wavelet Transform Compression

Wavelet-based image compression relies on performing a mathematical transformation of the original image, followed by a reduction in the storage size of the transform coefficients. In a typical discrete wavelet transform (DWT) analysis of an image, a set of four complementary filters is used to decompose the image into four different components: approximation (A), horizontal detail (HD), vertical detail (VD), and diagonal detail (DD). The approximation image captures the low wave number information in the image, whereas the detail images capture the high-wave-number information along the horizontal, vertical, and diagonal directions. The specific filter coefficients used in the decomposition are related to the wavelet family used in the analysis. Wavelets are orthogonal functions, and functionally the DWT is very similar to the discrete Fourier transform (DFT) and DCT, which also incorporate orthogonal transforming functions. The main difference between the DWT and the DFT/DCT is in the characteristics of the basis functions used. Whereas the DFT and DCT use sinusoids as basis functions, the DWT uses a set of basis functions that are defined by the recursive difference equation

$$\phi(x) = \sum_{k=0}^{J-1} C_k \phi(2x - k) \tag{3}$$

where J represents the number of nonzero coefficients C in the recursion. The most commonly used family of wavelets for image compression are the Daubechies wavelets (see Ref. 12). In particular, the Daubechies third-order wavelet is suited for DPIV compression because this function produces a satisfactory representation of an idealized one-dimensional model of the gray level distribution making up a particle image. Other commonly used wavelet families for image compression include Coiflets and Baylkin wavelets. In Ref. 6, Li presents a comparative study of these various families used for DPIV compression.

After image decomposition by the use of a suitable wavelet family defined by Eq. (3), each of the component images (A, HD, VD, and DD) can be fully described using a number of wavelet coefficients that is equal to one-quarter the size of the original image. That is, after a one-step DWT analysis, the number of pieces of information required to represent the original image fully is exactly the same as before. However, if the information in the image is for the most part localized in space, the majority of the detail wavelet coefficients should be negligible. Hence, most of these coefficients can be discarded without significant loss of image fidelity, resulting in a reduction in the amount of information needed to represent the original image and, thus, achieving image compression. If all of the detail coefficients are approximately zero (when A contains all of the relevant information) an information compression of one-quarter is achieved. This represents the best attainable compression using a one-step DWT approach. However, if one is to apply the DWT analysis recursively on A up to an m-step DWT decomposition, it is easy to see that the best attainable compression rate (CR) is given by

$$\max(\text{CR}) = 1\Big/\left(\tfrac{1}{4}\right)^m \tag{4}$$

Thus, in wavelet-based image compression, one may increase the amount of compression by increasing the decomposition level m and/or discarding more detail wavelet coefficients.

In practice, it is not possible to discard all of the detail coefficients to achieve maximum compression while at the same time maintaining an acceptable image quality. Thus, the insignificant detail wavelet coefficients are determined and discarded using a user-selectable threshold level. The easiest approach for setting such a threshold is known as the global approach, where a single threshold value is used for all detail images regardless of their DWT coefficient level. In such a case, the threshold value is varied systematically until one achieves an acceptable balance between the fraction of coefficients discarded and the retained image energy. However, this method is subjective. A somewhat better approach is one where the selected threshold value depends on the level of decomposition. In this method, the threshold is typically selected as a fraction of the largest detail wavelet coefficient at each level, the median of the wavelet coefficients at the different levels, etc. Again, this method is subjective and incapable of transparently adapting to different types of images. Much of the research into improving the quality of wavelet-based compressed images has concentrated on developing new methods of choosing the best threshold for removal of detail coefficients.

The ability to threshold wavelet coefficients over the complete set of detail images provides an advantage over DFT- and DCT-based techniques such as the JPEG algorithm because the DWT operates over a range of spatial wave numbers. In other words, wavelet image compression removes information from the image across multiple spatial scales and, thus, should provide superior image quality at higher compression rates relative to other techniques.

Compression Performance Measures

One of the challenges in conducting a comparative study of algorithms for image compression involves the selection of performance measures used to evaluate the results of the study. The image processing community has traditionally used three primary measures of performance for evaluating compression algorithms. The first measure is the CR and is defined as the ratio of the number of bits in the original image to those in the compressed image:

$$\text{CR} = B_{\text{tot}}/\tilde{B}_{\text{tot}} \tag{5}$$

where B_{tot} and $\tilde{B}_{\text{tot}}$ are the total bits contained in the original and compressed images, respectively. Obviously, as a higher compression rate is achieved, a smaller and, thus, more efficient compressed image file, is generated. The second measure, which is similar to the compression rate is the bit rate, is defined as

$$\text{bit rate} = \text{bits/pixel} = \tilde{B}_{\text{tot}}/MN \tag{6}$$

where M and N are the image pixel height and width, respectively. The bit rate can also be used to determine the efficiency of an algorithm, with a lower bit rate representing a more compact compressed image file. The third commonly used performance measure is the peak signal to noise ratio (PSNR), defined as

$$\text{PSNR(dB)} = 10\log_{10}\left\{L_g^2 \Bigg/ \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[f(i,j) - \tilde{f}(i,j)]^2\right\} \tag{7}$$

where L_g represents the maximum number of gray levels in the image (256 for an 8-bit monochrome image), $f(i, j)$ represents the ith, jth original image pixel value, and $\tilde{f}(i, j)$ represents the corresponding compressed image pixel value. The PSNR is commonly used to provide an objective level of performance of the compression algorithm in terms of the fidelity of the information retained in the compressed image. It is commonly accepted that images with PSNR levels above 32 dB are perceptually lossless. Nevertheless,

Table 1 Image sequences evaluated

Sequence	Horizontal pixels	Vertical pixels	Image type	Image density	Flow
VSJ 3D standard 301	256	256	Single exposure, multiple frame	500–1000	Wall impinging jet
Impedance tube	1320	1035	Double exposure, single frame	1000–2000	Acoustic, zero-mean flow
Separated flow	1300	1030	Single exposure, multiple frame	>5000	Separated flow

the PSNR is not a sufficient predictor of metrological errors that may be introduced to the compressed images. DPIV is a time-of-flight measurement technique where retention of pixel spatial information is equally important to retention of pixel amplitudes. Therefore, additional performance measures need to be defined.

For this study, the authors chose to implement a spatial error estimate based on examination of processed displacement vector data obtained from original and compressed DPIV images. This performance measure, which examines individual image displacements on a percentage basis, can be defined by

$$E_u = \frac{\tilde{u}(i, j) - u(i, j)}{u(i, j)} \times 100.0$$

$$E_v = \frac{\tilde{v}(i, j) - v(i, j)}{v(i, j)} \times 100.0 \qquad (8)$$

By the use of Eq. (8), E_u and E_v are computed with respect to the local u and v vector magnitudes over the entire ensemble of processed displacement vectors. The results are presented in the form of a histogram representing a vector percentage count vs percent deviation from the original displacement vector. For those compression algorithms that preserve spatial integrity of the image, a histogram formed using Eq. (8) should exhibit a narrow distribution centered around 0% deviation. As spatial errors in the individual u and v vector components increase (as would be expected if the compression rate were increased), the histogram will broaden and/or become nonsymmetric about 0% deviation. A quantitative measure of the spread of the E_u and E_v histograms can be obtained via computation of the standard deviation of the distributions, σ_{E_u} and σ_{E_v}. These values were computed and tabulated for all E_u and E_v histograms obtained for this study.

The final performance measure used for this study is the DPIV false vector rate (FVR), defined as the ratio of the number of false vectors detected during validation of the vector field to the total number of vectors processed:

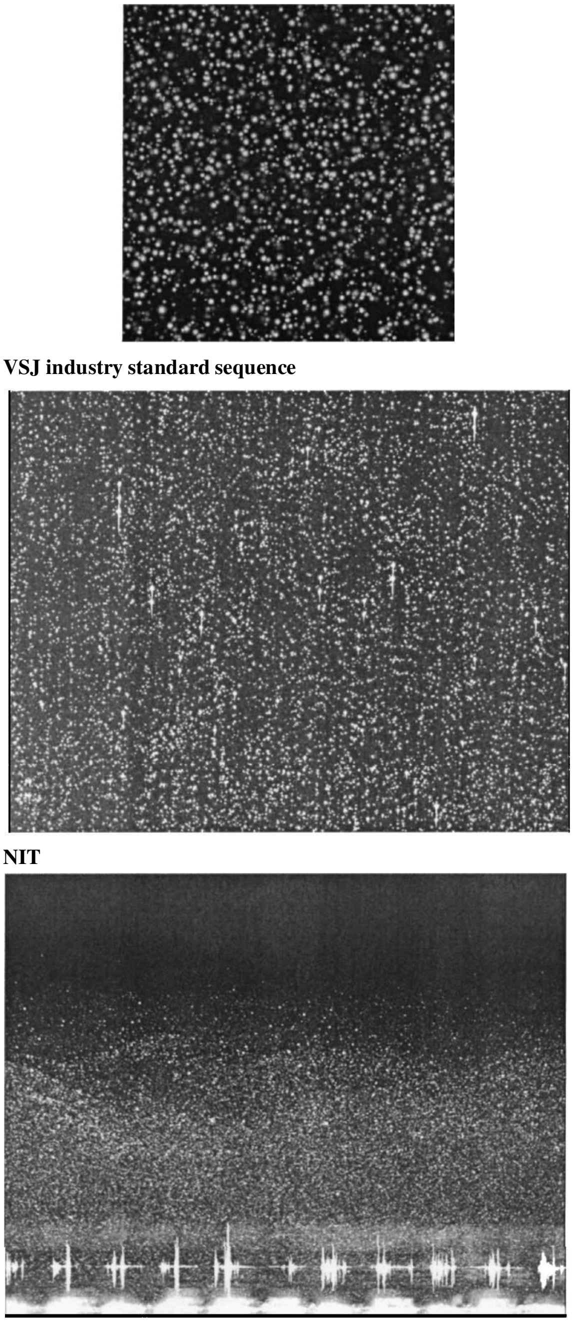
$$\mathrm{FVR} = \frac{\#\text{false vectors}}{PQ} \qquad (9)$$

The FVR is represented as a percentage and is computed for displacement vector fields obtained from both the original image before compression (to obtain a baseline) as well as the image after compression. Increases in the FVR due to compression effects are indicative of introduced spatial errors severe enough to cause the DPIV processing algorithms to select an incorrect vector magnitude and/or direction for one or more interrogation regions in the compressed images. Thus, a significant change in the FVR is a critical indicator of large-scale errors introduced to the DPIV displacement vector maps as a result of the compression process.

Image Sequences Chosen for Study

One industry standard image sequence and two sequences obtained from DPIV facility applications at NASA Langley Research Center were chosen to test the candidate compression algorithms. Each sequence contains 20 single images or image pairs depending on whether single or double exposed frames were acquired. Table 1 lists the relevant characteristics of each sequence and Fig. 1 shows a representative image from each sequence.

The industry standard sequence was obtained from the Visualization Society of Japan (VSJ) through their particle image velocimetry (PIV) standard project.[13] The VSJ standard images are fully documented and publicly available, providing a way for researchers in the field to test and compare various DPIV processing algorithms. Each image in the standard sequence contains 65,536, 8-bit pixels. The sequence requires cross-correlation analysis for processing, and the images contain approximately 500–1000 particles per image. The normal incidence impedance tube (NIT) sequence was acquired in an acoustically driven, zero-mean flow experiment conducted at NASA Langley Research Center in 1998 (Ref. 14). Each image in this sequence contains 1,366,200 8-bit pixels. The sequence was acquired using single-frame, double-exposure imaging and, thus, requires the use of autocorrelation analysis for processing. Each image in the sequence contains approximately 1000–2000 particles per image. Finally, a separated flow image sequence was recently acquired by Humphreys and Bartram in the NASA



VSJ industry standard sequence

NIT

Separated flow

Fig. 1 Test sequence sample images.

Langley Subsonic Basic Research Tunnel.[15] Each image in this sequence contains 1,339,000, 8-bit pixels. The sequence requires cross-correlation analysis for processing, and each image contains greater than 5000 particles per image.

The choice of the three sequences listed in Table 1 was based on several factors. First, it was desired to have three different particle image densities available because this tests the ability of the image compression algorithms to handle various spatial frequency ranges. In general, the greater the particle image density, the higher the frequency content of the resultant images. It was also desirable to have various background noise levels available in the sequences. By examining the representative images shown in Fig. 1, it can be seen that the VSJ sequence images are the cleanest in the sense that there is very little background noise present as compared with the other two sequences. Finally, it was desirable to have a sequence contaminated with unwanted flare light as shown in the separated flow image in Fig. 1 where unwanted reflections of laser light were observed at the bottom of the image. This flare light can be thought of as a low-frequency contamination superimposed on the relatively higher-frequency content represented by the particle images. This ensemble of image types provides the capability for testing the candidate compression algorithms under a number of realistic conditions.

Evaluation Procedure

Figure 2 is a flowchart showing the compression algorithm evaluation procedure that was adopted for this study. For each image or image pair in the three test sequences, an initial auto- or cross-correlation analysis and validation were performed to derive the baseline displacement vector map and FVR information needed for subsequent processing. The auto- and cross-correlation processing routines were written by the authors and are based on classical DPIV spatial analysis techniques as described by Raffel et al. in Ref. 1. Table 2 lists parameters used to process each of the sequences. Identification and tabulation of false vectors present in the displacement vector maps were performed using magnitude difference algorithms contained in the CleanVec validation system developed by Soloff and Meinhart[16] at the Laboratory for Turbulence and Complex Flow at the University of Illinois at Urbana–Champaign. Once the baseline analysis for an image was completed, the image was compressed using a candidate algorithm and the compression and bit rates computed using Eqs. (5) and (6). The image was then decompressed, and Eq. (7) was used to compute the PSNR. Decompressed images were analyzed and validated using identical processing parameters employed to analyze the original images. Note that identical processing must be done on both the original and decompressed images to remove any influences on the displacement vectors other than those introduced by the compression algorithm. After processing, false vectors identified during the validation step were removed from the original and decompressed vector maps, the FVR was computed using Eq. (9), and Eqs. (8) were used to compute the E_u and E_v histograms. The removal of false vectors before computation of the histograms ensures that any spatial errors that are detected are generated by valid vectors only. This is a reasonable step because the presence of invalid vectors would result in distorted histogram functions given that erroneous vectors typically deviate substantially from valid ones.

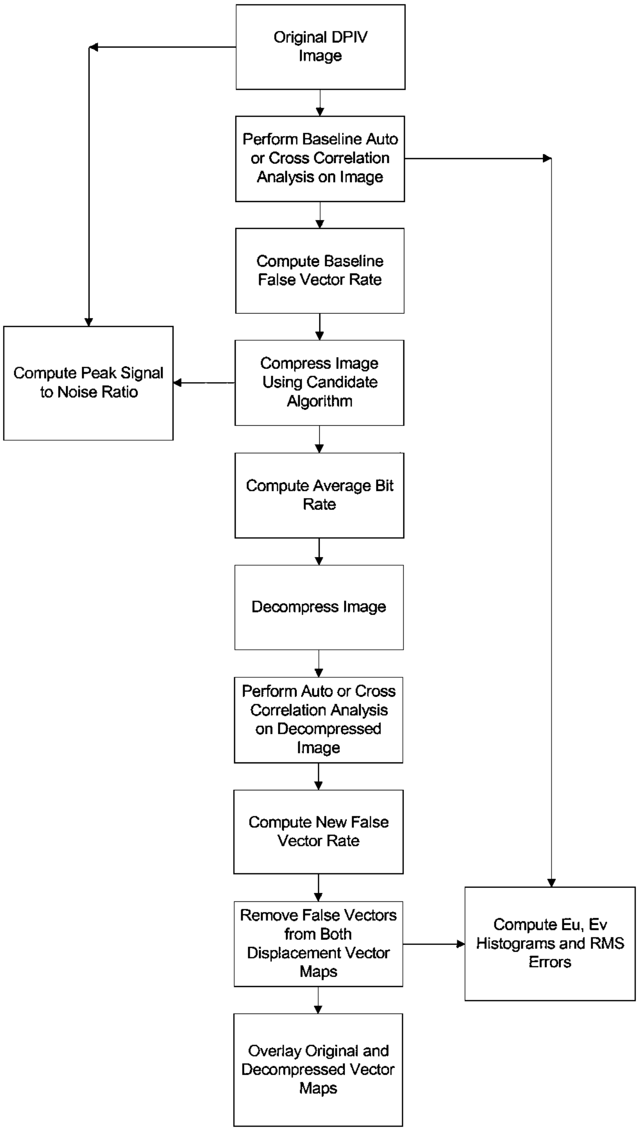

Original DPIV Image
Perform Baseline Auto or Cross Correlation Analysis on Image
Compute Baseline False Vector Rate
Compute Peak Signal to Noise Ratio
Compress Image Using Candidate Algorithm
Compute Average Bit Rate
Decompress Image
Perform Auto or Cross Correlation Analysis on Decompressed Image
Compute New False Vector Rate
Remove False Vectors from Both Displacement Vector Maps
Compute Eu, Ev Histograms and RMS Errors
Overlay Original and Decompressed Vector Maps

Fig. 2 Compression algorithm evaluation procedure.

Table 2 Auto-/cross-correlation processing parameters

Sequence	Interrogation size, pixels	Interrogation size overlap, %	Peak detection	Image threshold
VSJ 3D standard 301	32 × 32	50	Parabolic fit	150
NIT	128 × 128	50	Parabolic fit	150
Separated flow	64 × 64	50	Parabolic fit	150

Sample Results and Discussion

LZ77 Results

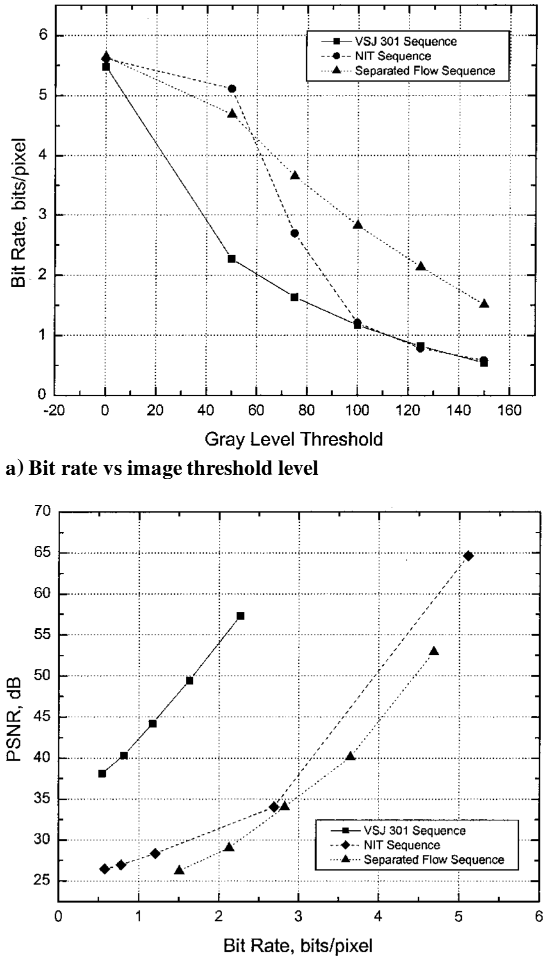
Six different gray-level thresholds spanning a range from 0 to 150 out of 256 gray levels were chosen for the LZ77 evaluation. A threshold of zero results in implementation of the standard LZ77 algorithm. For each test sequence, 20 images or image pairs were thresholded using Eq. (1) and compressed. The images were then decompressed and processed in accordance with the procedures outlined earlier, and the results were averaged. Table 3 lists the average bit rates, FVR, the PSNR levels, and the σ_{E_u} and σ_{E_v} values for this portion of the study.

Figure 3 illustrates the bit rate and PSNR level vs applied threshold. It is immediately evident from examination of the results in Table 3 and the graphs in Fig. 3 that thresholding an image before compression dramatically reduces the bit rate due to removal of background noise. The VSJ and NIT sequences experienced the lowest bit rates as the threshold was increased, whereas the separated flow sequence showed bit rates approximately three times as high. This is not surprising given that the VSJ and NIT sequences have lower particle image densities and, thus, more open background areas. A lower density allows more of these background areas to be zeroed out during thresholding, increasing the matching lengths between the LZ77 dictionary buffer and sequences of bytes in the image, thus reducing the compressed image size.

An interesting result is seen in Table 3 in the FVR values. Although there is a dramatic decrease in the PSNR level for each sequence as the bit rate is reduced, the effect on the accuracy of the processed displacement vector maps is negligible, even at the highest compression levels. There is no change in the FVR between the baseline and compressed results, and no discernable difference is observed in the vector maps. The reason for this insensitivity to threshold level stems from the construction of the DPIV auto- and

Table 3 Results of LZ77 compression tests

Threshold	Bit rate, bits/pixel	Baseline FVR, %	FVR, %	PSNR, dB	σ_{E_u}, %	σ_{E_v}, %
			VSJ 301			
0	5.48	0.98	0.98	Infinity	0.00	0.00
50	2.27	0.98	0.98	57.28	0.00	0.00
75	1.63	0.98	0.98	49.46	0.00	0.00
100	1.17	0.98	0.98	44.16	0.00	0.00
125	0.82	0.98	0.98	40.31	0.00	0.00
150	0.54	0.98	0.98	38.10	0.00	0.00
			NIT			
0	5.61	0.47	0.47	Infinity	0.00	0.00
50	5.11	0.47	0.47	64.62	0.00	0.00
75	2.69	0.47	0.47	34.02	0.00	0.00
100	1.21	0.47	0.47	28.32	0.00	0.00
125	0.78	0.47	0.47	26.93	0.00	0.00
150	0.58	0.47	0.47	26.42	0.00	0.00
			Separated flow			
0	5.64	8.88	8.88	Infinity	0.00	0.00
50	4.69	8.88	8.88	52.88	0.00	0.00
75	3.65	8.88	8.88	40.12	0.00	0.00
100	2.83	8.88	8.88	34.04	0.00	0.00
125	2.13	8.88	8.88	29.03	0.00	0.00
150	1.51	8.88	8.88	26.20	0.00	0.00



a) Bit rate vs image threshold level

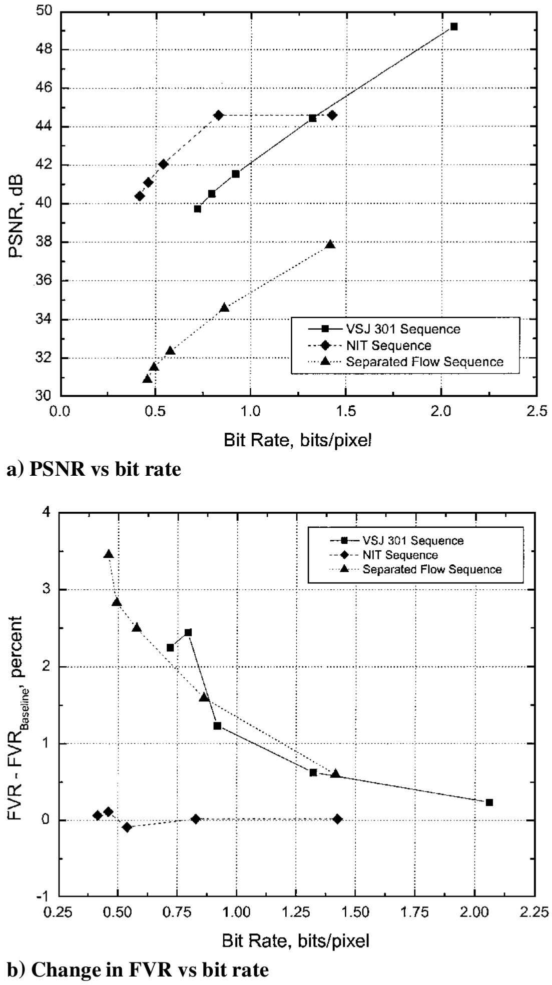
b) PSNR vs bit rate

Fig. 3 LZ77 compression results.

cross-correlation processing algorithms where a thresholding of individual interrogation regions in the image is performed before computing the correlation functions. As can be seen from the parameters shown in Table 2, the threshold level in the processing software was maintained at a value of 150. As long as the threshold level in the LZ77 algorithm was kept below the threshold level in the processing software, no degradation in the resultant displacement vector maps appeared. In fact, if the image and processing threshold levels are the same, the velocity errors (E_u and E_v) caused by compressing the images are zero. This can be seen in Table 3, where the σ_{E_u} and σ_{E_v} values obtained from the velocity errors are zero. On the other hand, if the compression threshold is increased beyond that set in the processing software, then increases in the FVR are expected to appear. Thus, when applying LZ77 compression with thresholding to DPIV images, a parametric study should be conducted to determine the optimal balance between compression threshold level and rate to maintain negligible errors in the processed displacement vector maps. For those applications where thresholding of interrogation regions cannot be performed during processing, it is advisable to

Table 4 Results of JPEG compression tests

Compression level	Bit rate, bits/pixel	Baseline FVR, %	FVR, %	PSNR, dB	σ_{E_u}, %	σ_{E_v}, %
			VSJ 301			
1	2.06	0.98	1.21	49.19	12.34	3.97
2	1.32	0.98	1.60	44.43	14.71	5.64
3	0.92	0.98	2.21	41.51	16.21	6.82
4	0.79	0.98	3.42	40.51	17.00	7.82
5	0.72	0.98	3.22	39.72	16.56	8.05
			NIT			
1	1.43	0.47	0.48	44.59	7.42	0.66
2	0.83	0.47	0.48	44.59	7.42	0.66
3	0.54	0.47	0.38	42.04	8.70	0.89
4	0.46	0.47	0.58	41.09	9.24	0.87
5	0.42	0.47	0.53	40.38	9.65	1.26
			Separated flow			
1	1.42	8.88	9.47	37.83	7.45	9.06
2	0.86	8.88	10.47	34.56	8.24	10.24
3	0.58	8.88	11.37	32.34	9.36	11.39
4	0.49	8.88	11.70	31.50	9.80	12.23
5	0.46	8.88	12.33	30.86	10.23	12.34



a) PSNR vs bit rate

b) Change in FVR vs bit rate

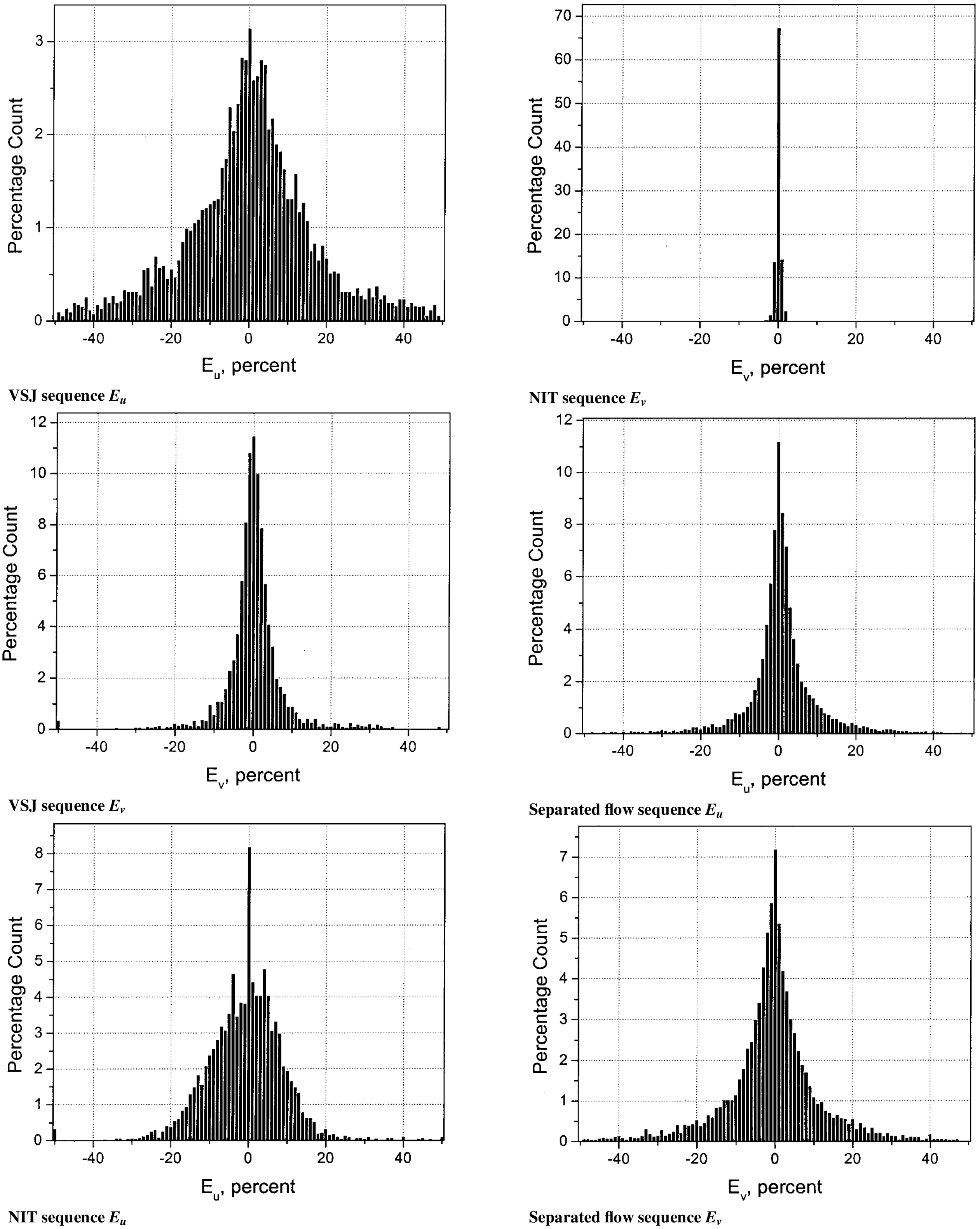
Fig. 4 JPEG compression results.

choose one of the lossy compression techniques examined in this study. Alternately, the LZ77 algorithm can be applied alone, at the expense of higher bit rates in the compressed images.

JPEG Results

Five different compression levels ranging from 1 to 5 were chosen for the JPEG evaluation, with level 1 representing the least compression and level 5 representing the highest. In a manner similar to that employed for the LZ77 evaluation, 20 images or image pairs in each sequence were compressed and processed. Table 4 lists the average bit rates, FVRs, PSNR levels, and the σ_{E_u} and σ_{E_v} values for this portion of the study.

Figure 4a shows the PSNR level as a function of the bit rate using JPEG compression. With the exception of the NIT sequence (which exhibits a flattening of the PSNR above 0.8 bits/pixel), the PSNR levels decrease uniformly as the bit rate is reduced. The majority of the PSNR levels are greater than 32 dB, indicative of little perceptual degradation in the images. This is unlike the PSNR results presented



VSJ sequence E_u

NIT sequence E_v

VSJ sequence E_v

Separated flow sequence E_u

NIT sequence E_u

Separated flow sequence E_v

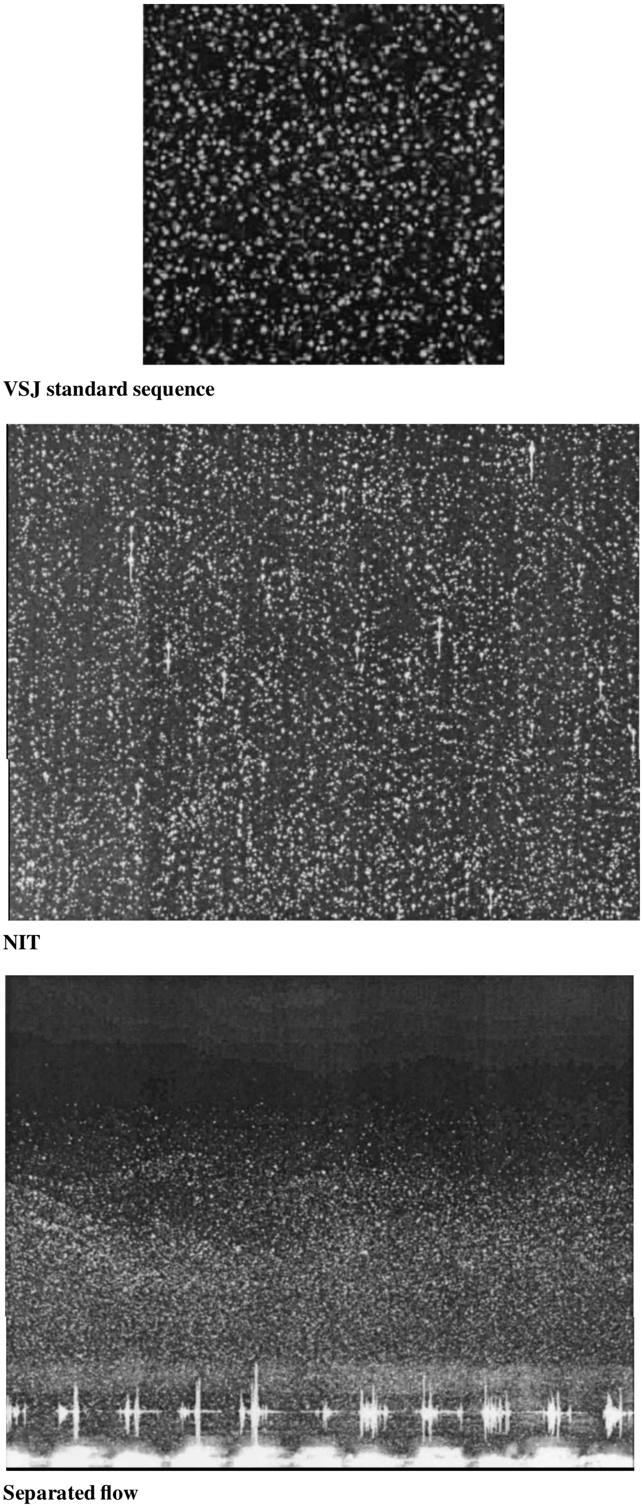
Fig. 5 Representative JPEG compression error histograms, compression level 5.

earlier in Fig. 3b for LZ77 compression. In particular, when using LZ77, both the NIT and separated flow sequences exhibit PSNR levels that are below 30 dB for the same bit rate range as that covered in Fig. 4a. This suggests that, for these two sequences, more pixel values are changed by the LZ77 algorithm than by JPEG compression at small bit rates. It is possible that this is caused by the use of a threshold in the former technique, which resets all pixel values in the background to zero regardless of their wave number content. As discussed earlier, both the NIT and separated flow sequences have a more noisy background containing larger pixel values than the VSJ sequence (which has comparable PSNR values in Figs. 3b and 4a). Therefore, the pixel values in the background of the NIT and separated-flow images will be affected substantially by the use of a threshold. This could cause a reduction in PSNR without adversely affecting the interrogated velocity vectors.

Figure 4b depicts the change in FVR as a function of bit rate. In general, for bit rates of 1 bit/pixel and above, less than a 1% change is noted in the FVR. Below 1 bit/pixel, the FVR starts to increase in the VSJ and separated flow sequences, with the change in FVR reaching from 2.5 to 3.5% for the lowest bit rates. The NIT sequence displays excellent stability in the FVR across the entire

Table 5 Results of wavelet compression tests

Compression level	Bit rate, bits/pixel	Baseline FVR, %	FVR, %	PSNR, dB	σ_{E_u}, %	σ_{E_v}, %
		VSJ 301				
1	3.59	0.98	0.53	39.99	14.47	5.53
2	1.89	0.98	0.63	40.37	14.63	5.49
3	0.80	0.98	0.86	39.48	15.32	7.87
4	0.40	0.98	3.54	36.81	19.20	13.76
		NIT				
1	4.04	0.47	0.38	50.67	5.55	0.69
2	1.96	0.47	0.38	50.12	5.71	0.71
3	0.78	0.47	0.45	46.19	6.50	0.75
4	0.40	0.47	0.48	41.50	9.07	1.04
		Separated flow				
1	4.00	8.88	6.76	38.60	7.78	9.42
2	1.93	8.88	6.75	37.45	7.95	9.46
3	0.81	8.88	7.40	33.69	8.58	10.34
4	0.40	8.88	8.57	30.80	9.81	12.44

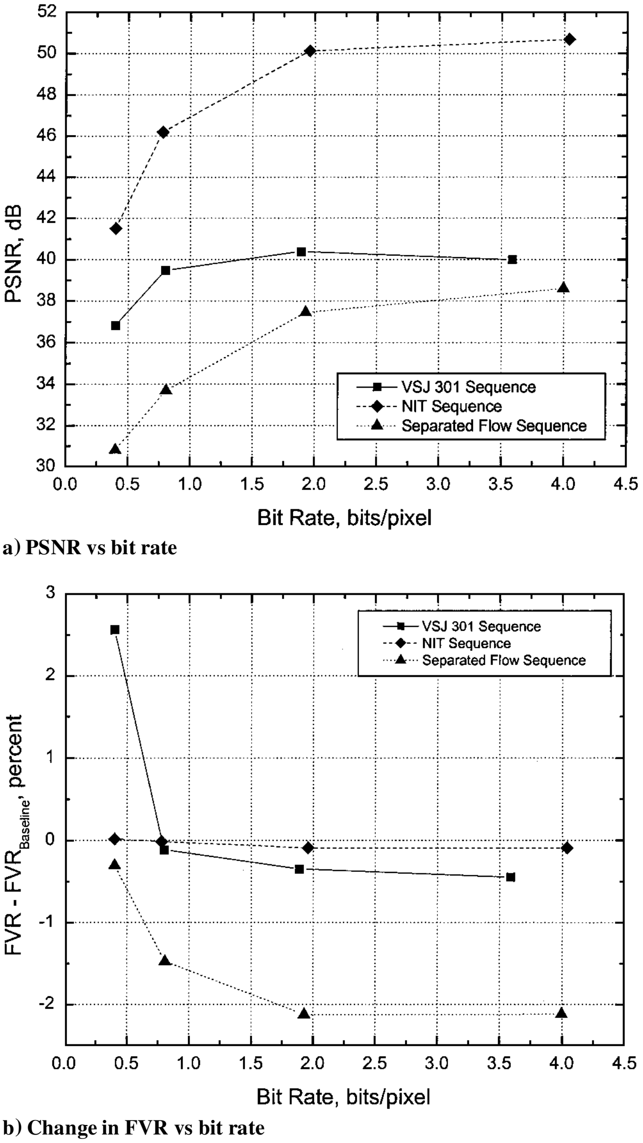


VSJ standard sequence

NIT

Separated flow

Fig. 6 Representative JPEG compressed images, compression level 5.



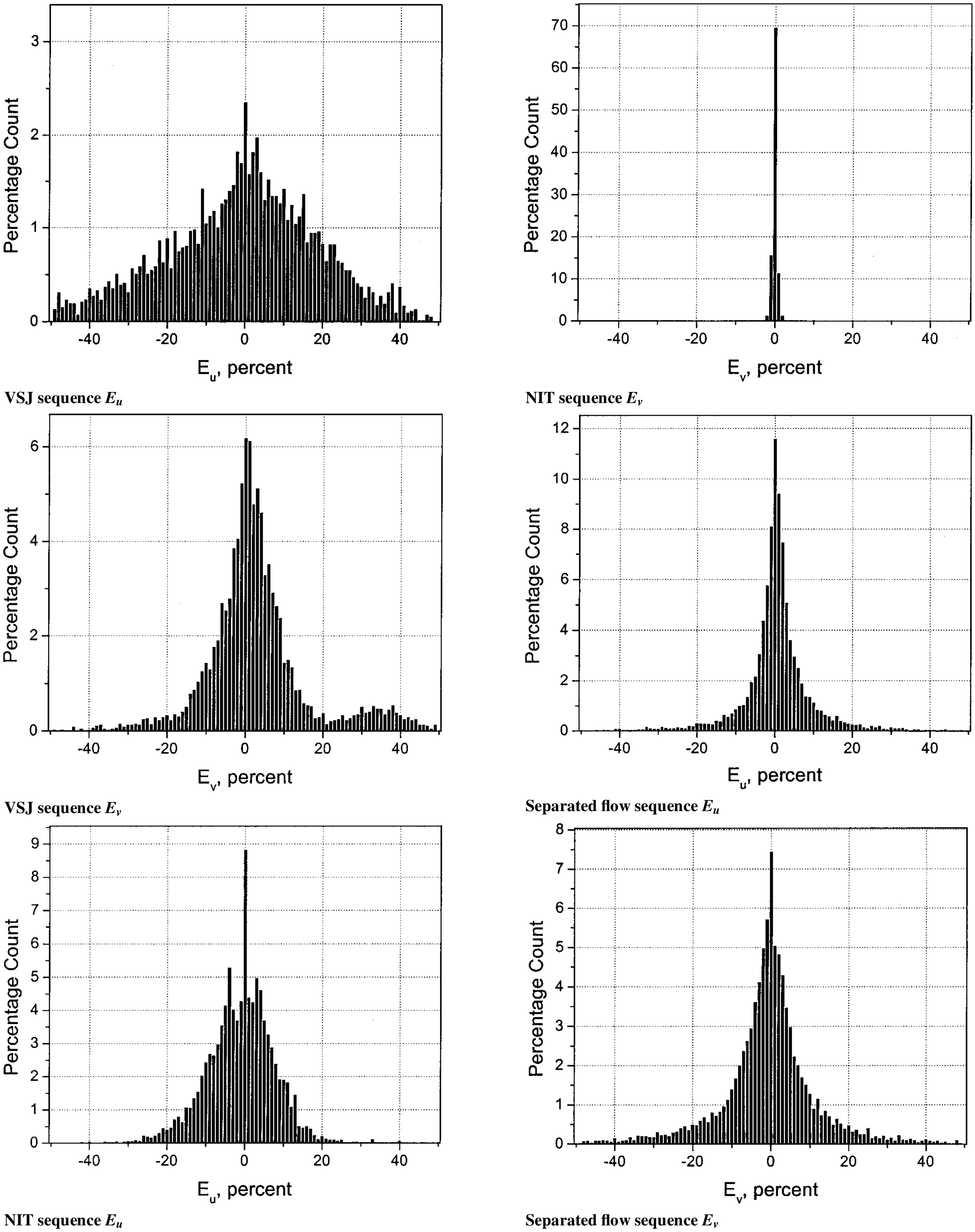
a) PSNR vs bit rate

b) Change in FVR vs bit rate

Fig. 7 Wavelet compression results.

range of bit rates employed in the evaluation. It is suspected that the excellent FVR trend shown for the NIT sequence is related to the use of autocorrelation analysis for the images. The autocorrelation analysis algorithm developed by the authors uses a restrictive search box for the correlation peak that may mitigate the generation of false vectors, even at low compression bit rates. The other two sequences examined for this study utilized cross-correlation analysis, which does not employ a search box. The FVR trends for the VSJ and separated flow sequences are superimposed on one another in Fig. 4b with the NIT results appearing noticeably lower.

Spatial errors introduced by the JPEG compression algorithm can be examined through the formation of E_u and E_v histograms using Eq. (8). Figure 5 displays a series of representative E_u and E_v histograms for a compression level of 5. Several observations can be made from an examination of these histograms and the resultant standard deviations listed in Table 4. First, the shape of the histogram distribution depends greatly on the type of DPIV image being compressed. For the VSJ and NIT sequences, the E_v histograms display a much more narrow distribution than the E_u histograms. This is to be expected, because the E_u and E_v functions given in Eq. (8)



VSJ sequence E_u

NIT sequence E_v

VSJ sequence E_v

Separated flow sequence E_u

NIT sequence E_u

Separated flow sequence E_v

Fig. 8 Representative wavelet compression error histograms, compression level 4.

are normalized by the local u and v vector displacement components. Thus, the functions are sensitive to the predominant flow direction. For the VSJ and NIT sequences, the predominant flow direction is aligned along the v direction, making local v components much larger than corresponding u components. On the other hand, the separated flow sequence contains highly variable local u and v components and, thus, displays more balanced E_u and E_v histogram distributions. The E_u and E_v functions could be normalized by the displacement vector magnitude; however, this would result in the generation of much more narrow distributions, which would make it harder to detect subtle changes in distribution shape. The use of local u and v components for normalization makes these histograms more useful in the present study in that they allow a better examination of changes as various compression levels or algorithms are employed.

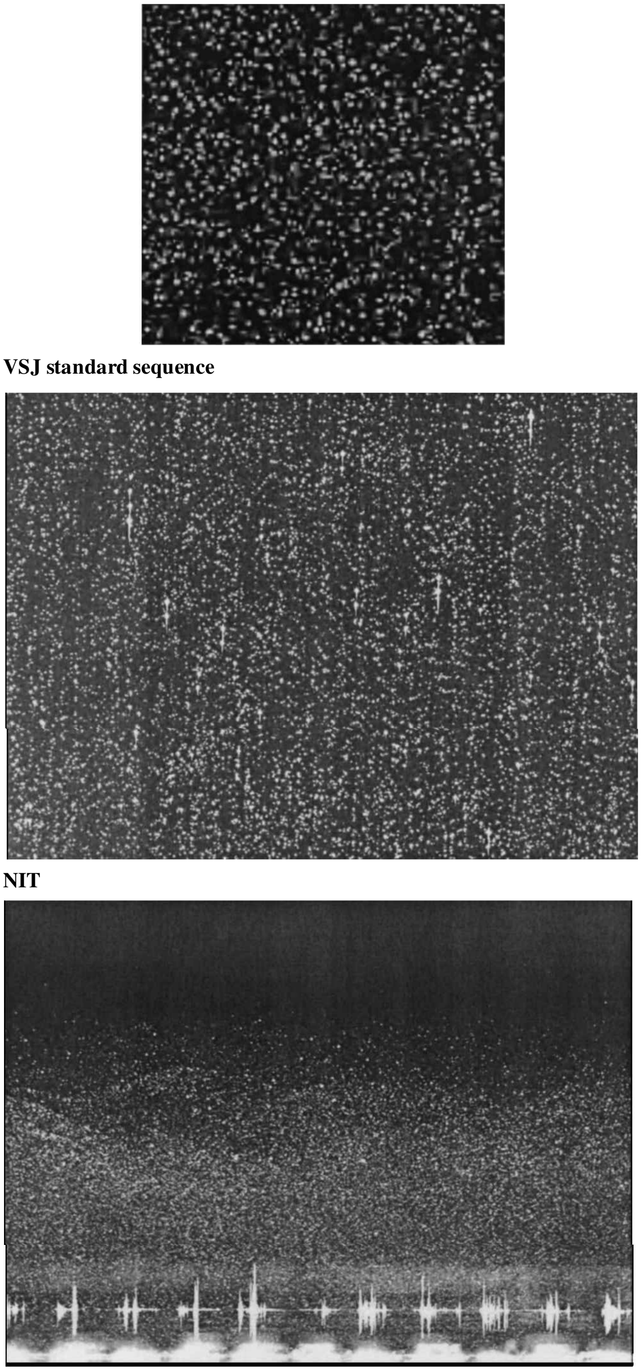
Figure 6 shows representative compressed images from the three sequences for a JPEG compression level of 5. Perceptual changes can be detected in the VSJ image as compared with the corresponding original image shown in Fig. 1. Similar changes are much harder to detect in the NIT image, whereas in the separated flow image, degradation of the image is most easily seen in the background area located at the top of the image. Visual differences in the images were reduced dramatically as the compression level was decreased from level 5 to level 1.

Wavelet Results

Four different compression levels were chosen for the wavelet evaluation, with level 1 representing the least compression and level 4 the highest. In a similar manner to the LZ77 and JPEG evaluations, 20 images or image pairs in each sequence were compressed and processed. Table 5 lists the average bit rates, FVRs, PSNR levels, and the σ_{E_u} and σ_{E_v} values for this portion of the study.

Figure 7a illustrates the change in PSNR level as the bit rate is reduced. Compared with the results shown in Fig. 4a for the JPEG algorithm, the wavelet compression produces remarkably consistent PSNR levels, with degradation of the PSNR only occurring for bit rates below 1 bit/pixel. This trend is consistent with the nature of the wavelet algorithm; the DWT removes information from the image across a range of wave number scales and, thus, does a better job of preserving image features such as edges, dots, etc., at higher compression rates. A similarly consistent trend can be seen in Fig. 7b, which depicts the change in FVR as a function of bit rate. The FVR is very consistent above bit rates of 1.5 bit/pixel and actually indicates a reduction in the FVR, in comparison to the uncompressed image results, for bit rates above 0.5 bit/pixel. The reason for this decrease can probably be attributed to the denoising effect that the DWT imparts to the images, an effect mentioned by Li in Ref. 6. The separated flow sequence would be most sensitive to any denoising because it contains the highest levels of background and particle image noise among the three sequences and, indeed, from examining Fig. 7b, this sequence shows the largest reduction in the FVR. Also note that the Fig. 7b results, when compared to Fig. 4b, demonstrate that the use of wavelet compression provides less false vectors than JPEG. This is particularly true for the VSJ and separated flow sequences. On the other hand, the FVR for the NIT sequence seems to be practically unaffected regardless of which of the two compression techniques is used. This is believed to be caused by the fairly restrictive peak search box used with the autocorrelation analysis of the NIT sequence images.

Figure 8 displays a series of representative E_u and E_v histogram functions for each of the three image sequences for a wavelet compression level of 4. In general, the histogram distributions for the NIT and separated flow image sequences are very similar to those shown for the JPEG evaluation. However, the VSJ image sequence shows a much broader E_u distribution and a slightly larger E_v distribution over the JPEG results. For reasons unknown, the E_v distribution for this sequence also appears to be slightly bimodal in structure, with a small peak occurring at between 20 and 50% deviation. The reason for the broadening of the distribution can be explained via an examination of the wavelet compressed images shown in Fig. 9. The VSJ image shown in Fig. 9 is perceptively more distorted than the other two images for similar compression bit rates. The reason for the increased distortion in the VSJ image is unclear but may be related to the wave number scales appearing in the image; the wavelet compression algorithm appears to have performed a poor job of thresholding the detail coefficients for this particular sequence. Based on these results, it appears the wavelet compression technique, in general, outperforms the JPEG algorithm; however, it is advisable to conduct test runs using various wavelet compression rates to determine the optimal bit rate to use for a particular class of image.


VSJ standard sequence

NIT

Separated flow

Fig. 9 Representative wavelet compressed images, compression level 4.

Summary

Based on the results of this study, there are several candidate algorithms that may be successfully applied to compress DPIV images with minimal error. For those applications where thresholding of individual DPIV interrogation regions can be performed before correlation functions are formed, it is possible to achieve compressed bit rates of less than 1 bit/pixel with no degradation of the processed vector maps by using the LZ77 algorithm. For those applications where thresholding cannot be performed, the wavelet compression algorithm, in general, yields the best performance in terms of imparting negligible increases in the FVR (in some cases actually decreasing the rate) and imparting negligible increases in the E_u and E_v histograms. However, depending on the spatial wave number content of individual images, test cases need to be examined when using wavelet compression to ensure that the proper detail coefficient thresholds are selected, thus minimizing spatial errors. If a modest increase in the FVR can be tolerated, then compressed bit rates of 0.5–2 bits/pixel can be achieved using JPEG compression. Regardless of the type of compression algorithm contemplated for use with DPIV, it is advised that test images be compressed and processed and that any errors due to the algorithm be identified before replacing the original images with compressed ones during data archiving.

Acknowledgments

The authors wish to thank Scott M. Bartram of the NASA Langley Advanced Measurement and Diagnostics Branch for his essential and long-lasting contribution to the acquisition of the NASA Langley Research Center DPIV data presented in this paper. The authors also wish to thank James F. Meyers of the Advanced Measurement and Diagnostics Branch for his helpful suggestions regarding the LZ77 algorithm evaluation and Laura M. Hudy of Michigan State University for her assistance in the collection of DPIV data for the separated flow sequence presented in this paper.

References

[1] Raffel, M., Willert, C., and Kompenhans, J., *Particle Image Velocimetry: A Practical Guide*, Springer-Verlag, New York, 1998.

[2] Stanislas, M., Kompenhans, J., and Westerweel, J. (eds.), *Particle Image Velocimetry: Progress Towards Industrial Application*, Kluwer Academic, Boston, 2000, pp. 203–523.

[3] Samimy, M., and Wernet, M. P., "Review of Planar Multiple-Component Velocimetry in High-Speed Flows," *AIAA Journal*, Vol. 38, No. 4, 2000, pp. 553–574.

[4] Westerweel, J., "Fundamentals of Digital Particle Image Velocimetry," *Measurement Science and Technology*, Vol. 8, No. 12, 1997, pp. 1379–1392.

[5] Kahler, C. J., and Kompenhans, J., "Multiple Plane Stereo PIV—Technical Realization and Fluid-Mechanical Significance," *Proceedings of 3rd International Workshop on PIV*, 1999, pp. 281–286.

[6] Li, H., "Particle Image Velocimetry Based on Wavelet Image Compression Technique," *Proceedings of 3rd International Workshop on PIV*, 1999, pp. 477–482.

[7] Freek, C., Sousa, J. M. M., Hentschel, W., and Merzkirch, W., "On the Accuracy of a MJPEG-Based Digital Image Compression PIV System," *Experiments in Fluids*, Vol. 27, No. 4, 1999, pp. 310–320.

[8] Cenedese, A., De Gregorio, F., Pocecco, A., and Querzoli, G., "Effects of Image Compression on PIV and PTV Analysis," *Particle Image Velocimetry: Progress Towards Industrial Application*, Kluwer Academic, Boston, 2000, pp. 429–438.

[9] Ziv, J., and Lempel, A., "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, Vol. 23, No. 3, 1977, pp. 337–343.

[10] Nelson, M., and Gailly, J., *The Data Compression Book*, 2nd ed., M and T Books, New York, 1996, pp. 201–214.

[11] Wallace, G. K., "The JPEG Still Picture Compression Standard," *Communications of the ACM*, Vol. 34, No. 4, 1991, pp. 31–44.

[12] Kaiser, G., *A Friendly Guide to Wavelets*, Birkhauser, Boston, 1994, pp. 176–199.

[13] Okamoto, K., Nishio, S., Saga, T., and Kobayashi, T., "Standard Images for Particle Imaging Velocimetry," *Measurement Science and Technology*, Vol. 11, No. 6, 2000, pp. 685–691.

[14] Humphreys, W. M., Jr., Bartram, S. M., Parrott, T. L., and Jones, M. G., "Digital PIV Measurements of Acoustic Particle Displacements in a Normal Incidence Impedance Tube," AIAA Paper 98-2611, 1998.

[15] Humphreys, W. M., Jr., and Bartram, S. M., "Measurement of Separated Flow Structures Using a Multiple-Camera DPIV System," *Proceedings of the 19th International Congress on Instrumentation in Aerospace Simulation Facilities (ICIASF)*, IEEE Press, Piscataway, NJ, 2001, pp. 82–93.

[16] Soloff, S. M., and Meinhart, C. D., *CleanVec: PIV Vector Validation Software*, Ver. 1.13, Lab. for Turbulence and Complex Flow, Univ. of Illinois, Urbana, IL, 1999.

W. R. Lempert
Guest Associate Editor