AIAA 2001-0695
A Comparative Study of Image Compression Techniques for Digital Particle Image Velocimetry

William M. Humphreys, Jr.
NASA Langley Research Center
Hampton, VA 23681

Ahmed M. Naguib
Michigan State University
East Lansing, MI 48824

39th Aerospace Sciences Meeting & Exhibit

8-11 January 2001 / Reno, NV

AIAA 2001-0695

A COMPARATIVE STUDY OF IMAGE COMPRESSION TECHNIQUES FOR DIGITAL PARTICLE IMAGE VELOCIMETRY

William M. Humphreys, Jr.*
NASA Langley Research Center
Hampton, Virginia 23681

Ahmed M. Naguib†
Michigan State University
East Lansing, Michigan 48824

ABSTRACT

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 a series of 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. Evaluation of the various methods was accomplished using quantitative perceptual and metrological performance measures. The results of the study indicate that several methods of compression are applicable to digital particle image velocimetry images. For selected applications where thresholding of the image gray levels can be performed, lossless algorithms yield excellent performance in terms of compression rates and introduction of negligible errors to the images. For applications where thresholding is not appropriate, JPEG and wavelet-based algorithms yield high compression rates with minor spatial errors introduced to the images. Fractal-based algorithms were shown to be capable of providing high compression rates, but the technique must be guided via the use of a suitably selected template image if spatial errors are to be minimized in the compressed images.

NOMENCLATURE

B_{tot}	Total bits in original image
$\tilde{B}_{tot}$	Total bits in compressed image
DCT	Discrete cosine transform, see eqn. (2)
$E_{u,RMS}$	Global spatial error estimate of u component vector, pixels, see eqn. (8)
$E_{v,RMS}$	Global spatial error estimate of v component vector, pixels, see eqn. (8)
E_u	Spatial error estimate of individual u component vector, percent, see eqn. (9)
E_v	Spatial error estimate of individual v component vector, percent, see eqn. (9)
FVR	False vector rate, see eqn. (10)

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

†Assistant Professor, Department of Mechanical Engineering, Member AIAA.

$f(i, j)$	Intensity of i^{th}, j^{th} pixel of original image
$f'(i, j)$	Intensity of i^{th}, j^{th} pixel of thresholded image
$\tilde{f}(i, j)$	Intensity of i^{th}, j^{th} pixel of compressed image
L_g	Maximum gray levels in image
m	Decomposition levels in discrete wavelet compression
M, N	Number of horizontal and vertical pixels, respectively, in image
P, Q	Number of horizontal and vertical displacement vectors, respectively, in vector map
$PSNR$	Peak signal-to-noise ratio, dB, see eqn. (7)
$u(i, j)$	u displacement component obtained from processed original image, pixels
$v(i, j)$	v displacement component obtained from processed original image, pixels
$\tilde{u}(i, j)$	u displacement component obtained from processed compressed image, pixels
$\tilde{v}(i, j)$	v displacement component obtained from processed compressed image, pixels
$\phi(x)$	Recursive wavelet basis function, see eqn. (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 which are capable of acquiring images at 10 frames / second and faster. Associated with the introduction of these new camera technologies, stereo DPIV systems have recently been described in the literature which incorporate as many as four separate cameras.[5] 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 which required the use of four separate 1300- by 1030-pixel cameras connected to 8-bit digitizing frame grabbers. During four days of testing, 32,800 separate images were acquired. These images required approximately 41 gigabytes 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 which will enable high speed DPIV acquisition at rates surpassing 1000 frames / 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 performance based on human visual perception – a perfectly acceptable criteria given that a majority of image processing algorithms are designed to provide outputs which are viewed by people. With a few exceptions (most notably astronomical imaging), less attention has been paid to the consequences of using compressed images in metrological applications like DPIV. In particular, the effect of image compression on post-processed 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. A few 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 de-noising 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. Nevertheless, 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 versus a lossy (i.e., loss of information) compression algorithm have not been thoroughly examined.

To complement and extend the previous work given in references 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

The authors chose one lossless, one modified lossless, and three lossy image compression techniques for this study. The lossless technique examined was the Lempel Ziv 77 (LZ77) dictionary-based compressor. 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 in reference 8. However, Cenedese created a binary image after thresholding, whereas the technique used for this study consists of a simple gray level truncation technique which maintains an 8-bit image. The lossy techniques which were examined are all based on various mathematical transformation and recursion equations and include the discrete cosine transform, the discrete wavelet transform, and a recursive fractal equation. Each of these algorithms is described in more detail subsequently. A summary of characteristics for the various candidate algorithms is presented in Table 1. Each algorithm was implemented for this study using readily available commercial off-the-shelf software packages.

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. The algorithm, shown in block diagram form in Figure 1, 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 to 16384 bytes. 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) – 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, non-textual data (e.g., binary image files) typically cannot be compressed more than approximately 30 percent 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 which couples the LZ77 compressor to a thresholding process. Before compression, image pixels are gray level thresholded using a simple truncation formula:

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

where *f(i,j)* represents the i^{th}, j^{th} original pixel value and *f'(i,j)* represents the corresponding thresholded value. The benefit of performing this operation prior to 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. While 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 non-compressed images.

JPEG Compression: The Joint Photographic Experts Group (JPEG) compression standard is the leading technique for use in numerous imaging applications[10], and is incorporated into a number of commercial DPIV processing systems. The algorithm is based on the two-dimensional discrete cosine transform (DCT) pair:

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

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

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

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 equation (2). The popularity of the JPEG technique is partly due to the speed of the algorithm – the DCT transform shown in equation (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.

A diagram illustrating how the DCT is applied for monochrome image compression is shown in Figure 2. An input image is broken up into a series of non-overlapping 8 x 8 pixel blocks. Each pixel block is independently transformed into the spatial frequency (wavenumber) 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 which 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 easily specify 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-wavenumber information in the 8 x 8 pixel block is removed. The loss of too much information results in a visible "checkerboard" pattern in the image, 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 x 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 image displacement vector derived from 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, shown in Figure 3, 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-wavenumber information in the image, while the detail images capture the high-wavenumber 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 non-zero coefficients *C* in the recursion. The most commonly used family of wavelets for image compression are the Daubechies wavelets. In particular, the Daubechies third-order wavelet is suited for DPIV image compression because this function produces a satisfactory representation of an idealized one-dimensional model of particles in a DPIV image. Other commonly used wavelet families for image compression include Coiflets and Baylkin wavelets. In reference 6, Li presents a comparative study of these various families used for DPIV compression.

After image decomposition using a suitable wavelet family defined by equation (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 ¼ 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 the detail coefficients are approximately zero (when *A* contains all the relevant information) an information compression of ¼ 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 ratio is given by

$$Max(CR) = \frac{1}{(1/4)^m} \tag{4}$$

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 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 like the JPEG algorithm, since the DWT operates over a range of spatial wavenumbers. 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.

Fractal Compression: A different class of image compression based on fractal self-similarity has appeared in recent years, and shows promise for achieving very high compression ratios while retaining reasonable details in the compressed images.[11] Only a brief overview of the technique is presented here – for more detail the reader is referred to an excellent description of fractal compression by Nelson and Gailly.[12] In general, the fractal image algorithm creates a compressed file containing a packed list of transformation coefficients which map the original image to itself, in essence creating a mathematical model for the image. Using this model, the original image can be reproduced at any resolution desired, even resolutions higher than the original uncompressed image. Because only the mathematical model need be saved, very high compression ratios are possible using this technique. However, since a self-similarity model is employed to compress an image, the algorithm is susceptible to the creation of unwanted artificial details or *artifacts* in the reconstructed image. These artifacts can corrupt the information content of the image. For example, artifacts in a fractal reconstructed DPIV image can appear as spatial distortions in individual particle images, greatly reducing the accuracy of any processed displacement vector data derived from these images. However, these artifacts can be reduced by use of a *fractal template* containing small representations of features in the image to be compressed. Impressive improvement in compressed image quality can be achieved using templates. Most commercial software packages which implement fractal compression support the use of fractal templates.

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 referred to as the compression rate, and is defined as the ratio of the number of bits in the original image to those in the compressed image:

$$Compression\ Rate = \frac{B_{tot}}{\tilde{B}_{tot}} \tag{5}$$

where B_{tot} and $\tilde{B}_{tot}$ 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, defined as:

$$Bit\,Rate = \frac{bits}{pixel} = \frac{\tilde{B}_{tot}}{MN} \tag{6}$$

where *M, 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, defined as:

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

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, the PSNR is not a sufficient predictor of metrological errors which 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. The authors chose to implement two different spatial error estimates based on examination of processed displacement vector data obtained from original and compressed DPIV images.

One measure implemented for this study consists of a root-mean-square (RMS) error estimate computed using the displacement vectors derived from processing both the original and compressed DPIV images. This measure provides a look at global spatial biases imparted to the images as part of the compression process. The RMS error estimates are defined as:

$$\begin{aligned} E_{u,RMS} &= \frac{1}{PQ}\sqrt{\sum_{i=1}^{P}\sum_{j=1}^{Q}\ [u(i,j) - \tilde{u}(i,j)]^2} \\ E_{v,RMS} &= \frac{1}{PQ}\sqrt{\sum_{i=1}^{P}\sum_{j=1}^{Q}\ [v(i,j) - \tilde{v}(i,j)]^2} \end{aligned} \tag{8}$$

where $u(i,j), v(i,j)$ are the ith and jth horizontal and vertical components of displacement in the processed data obtained from the original DPIV image, $\tilde{u}(i,j), \tilde{v}(i,j)$ are the corresponding components obtained from processing the compressed image, and *P,Q* are the number of horizontal and vertical displacement vectors in the processed data, respectively. Equation (8) is a modified version of an error function described by Huang, et.al.[13], for use in DPIV error investigations. In practice, the $E_{u,RMS}$ and $E_{v,RMS}$ functions should be approximately zero if the individual vector component errors are truly random in nature.

To accompany the error estimates given in equation (8), an additional performance measure which examines individual image displacements on a percentage basis can be defined by:

$$\begin{aligned} E_u &= \frac{\tilde{u}(i,j) - u(i,j)}{u(i,j)} * 100.0 \\ E_v &= \frac{\tilde{v}(i,j) - v(i,j)}{v(i,j)} * 100.0 \end{aligned} \tag{9}$$

Using equation (9), 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 versus percent deviation from the original displacement vector. For those compression algorithms which preserve spatial integrity of the image, a histogram formed using equation (9) should exhibit a narrow distribution centered around zero percent 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 non-symmetric about zero percent deviation.

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

$$FVR = \frac{\#\,False\ Vectors}{PQ} \qquad (10)$$

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 2 lists the relevant characteristics of each sequence and Figure 4 shows a representative image from each sequence.

The industry standard sequence was obtained from the Visualization Society of Japan (VSJ) through their PIV Standard Project.[14] 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 65536, 8-bit pixels. The sequence requires cross-correlation analysis for processing, and the images contain a “low” particle image density of a few thousand 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.[15] 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 auto-correlation analysis for processing. Each image in the sequence contains a “medium” particle image density of several thousand particles per image. Finally, a separated flow image sequence was recently acquired by the authors in the NASA Langley Subsonic Basic Research Tunnel. Each image in this sequence contains 1,339,000 8-bit pixels. The sequence requires cross-correlation analysis for processing, and each image contains a “high” particle image density of over 5000 particles per image.

The choice of the three sequences listed in Table 2 was based on several factors. First, it was desired to have three different particle image densities available since 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 Figure 4, 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 Figure 4 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 5 depicts a flowchart showing the compression algorithm evaluation procedure which 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 was performed to derive the baseline displacement vector map and false vector rate 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 reference 1. Table 3 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 – Urbana. 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 equations (5) and (6). The image was then decompressed and equation (7) was used to compute the PSNR. The decompressed image was then analyzed and validated using the identical processing parameters employed to analyze the original image. Note that identical processing *must* be done on both the original and decompressed images in order 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 equation (10), and equations (8) and (9) were used to compute the global RMS errors and E_u, E_v histograms. The removal of false vectors before computation of the RMS errors ensures that any spatial errors which are detected are generated by valid vectors only. This is a reasonable step because the presence of invalid vectors would result in inflated RMS errors given that erroneous vectors typically deviate substantially from valid ones.

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 equation (1) and compressed. The images were then decompressed and processed in accordance with the procedures outlined previously, and the results were averaged. Table 4 lists the average bit rates, false vector rates, PSNR levels, and $E_{u,RMS}$ and $E_{v,RMS}$ values for this portion of the study.

Figure 6 illustrates the bit rate and PSNR level versus applied threshold. It is immediately evident from examination of the results in Table 4 and the graphs in Figure 6 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, while 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 and thus reducing the compressed image size.

An interesting result is seen in Table 4 in the FVR, $E_{u,RMS}$ and $E_{v,RMS}$ values. While 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 there are no discernable bias errors in the vector maps. The reason for this insensitivity to threshold level stems from the construction of the DPIV auto- and 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 3, 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. Of course, if the compression threshold is increased beyond that set in the processing software, then increases in the FVR and RMS bias levels 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, then it is advisable to 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 5 lists the average bit rates, false vector rates, PSNR levels, and $E_{u,RMS}$ and $E_{v,RMS}$ values for this portion of the study.

Figure 7(a) 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. Figure 7(b) 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 one percent 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 2.5 to 3.5 percent for the lowest bit rates. The NIT sequence displays excellent stability in the FVR across the entire 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 auto-correlation analysis for the images. The auto-correlation analysis algorithm developed by the authors uses a restrictive search box for the correlation peak which 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 Figure 7(b) with the NIT results appearing noticeably lower.

Figure 8 illustrates average $E_{u,RMS}$ and $E_{v,RMS}$ values for the three image sequences. In general most of the results are tightly grouped between 0.010 and 0.020 pixels with the $E_{v,RMS}$ values for the VSJ sequence appearing slightly higher at between 0.015 and 0.025 pixels. These results would tend to indicate that there are only small global spatial biases being imparted to the images by the JPEG algorithm, and that these biases do not increase appreciably as the compression rate is increased. Note that while these results indicate that the JPEG algorithm performs well in terms of the overall ensemble of particle displacement vectors generated from processing the compressed images, it is not indicative of errors which may be introduced to *individual* displacement vectors. These individual spatial errors can best be seen through the formation of E_u, E_v histograms using equation (9). Figure 9 displays a series of representative E_u, E_v histograms for a JPEG compression level of 5. Several observations can be made from an examination of these histograms. 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, E_v functions given in equation (9) 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, 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. Using the local u and v components for normalization make 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 10 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 Figure 4. Similar changes are much harder to detect in the NIT image, while 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.

Figure 11 shows overlaid image displacement vector maps for JPEG compression levels of 1 and 5 for a representative image pair taken from the VSJ sequence. An examination of these maps reveals only minor differences in the validated vector fields as the compression level is increased from level 1 to level 5. In particular, the largest deviations are noted in the upper right quadrant of the vector map of Figure 11(b). The JPEG compression algorithm produced comparable vector maps for the other two image sequences examined in the study.

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 6 lists the average bit rates, false vector rates, PSNR levels, and $E_{u,RMS}$ and $E_{v,RMS}$ values for this portion of the study.

Figure 12 illustrates the change in PSNR level as the bit rate is reduced. Compared with the results shown in Figure 7(a) 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 wavenumber 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 Figure 13 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 de-noising effect which the DWT imparts to the images, an effect mentioned by Li in reference 6. The separated flow sequence would be most sensitive to any de-noising since it contains the highest levels of background and particle image noise among the three sequences, and indeed from examining Figure 13, this sequence shows the largest reduction in the FVR.

Figure 14 shows the average $E_{u,RMS}$ and $E_{v,RMS}$ values for the three image sequences using wavelet compression. As was observed for the JPEG evaluation, most of the results are tightly grouped between 0.010 and 0.020 pixels for bit rates above 0.5 bits/pixel. There is consistency in the RMS values across a wide range of bit rates. These results would tend to indicate that there are only small global spatial biases being imparted to the images by the wavelet algorithm, and that these biases do not change appreciably as the compression rate is increased.

Figure 15 displays a series of representative E_u, 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 percent deviation. The reason for the broadening of the distribution can be explained via an examination of the wavelet compressed images shown in Figure 16 and the overlaid image displacement vector maps shown in Figure 17. The VSJ image shown in Figure 16(a) 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 wavenumber 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. As a result, the vector map shown in Figure 17(b) shows more visible deviations between vectors derived from the original and compressed images. 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.

Fractal Results: Due to an image size limitation of 500,000 pixels in the commercial fractal compression program used for this study, only the VSJ industry standard sequence could be processed. Nevertheless, four different fractal compression levels were chosen, with level 1 representing the least compression and level 4 the highest. As with the previous evaluations, the 20 image pairs in the VSJ sequence were compressed both with and without the use of fractal templates and processed. Table 7 lists the average bit rates, false vector rates, PSNR levels, and $E_{u,RMS}$ and $E_{v,RMS}$ values for this portion of the study. As can be seen from the results presented in Table 7, the fractal algorithm performs extremely poorly when used without templates, with extreme degradation of the compressed images and 40 – 50 percent increases in the FVR observed. These results are so poor that it is essential that fractal image templates be used to guide the process of compressing DPIV images. For this reason, the results which follow are based on the use of templates during compression.

Figure 18 shows the PSNR levels and change in FVR rate as a function of bit rate for the fractal algorithm using templates. The PSNR levels are essentially constant across the entire range of bit rates employed for the study. This result is not surprising given the self-similarity nature of the fractal compression algorithm. The use of self-similarity equations implies that reconstructed fractal images have similar spatial wavenumber characteristics regardless of the amount of compression applied to the original image. DPIV images compressed using the fractal technique over a range of bit rates should have similar background and particle image features, and thus similar PSNR levels. In a like manner, the change in FVR between the baseline and compressed processed images is shown to be a consistent 3.5

percent for all bit rates, as can be seen in Figure 18(b). The $E_{u,RMS}$ and $E_{v,RMS}$ results shown in Figure 19 are also constant at 0.022 pixels for the u component and 0.033 pixels for the v component.

Figure 20 displays representative E_u, E_v histograms for the VSJ sequence for a fractal compression level of 4. As with the wavelet evaluation, the VSJ image sequence shows a much broader E_u distribution and a slightly larger E_v distribution over the JPEG results. The reason for this broadening can be seen in Figure 21. The VSJ image shown in this figure is perceptively more distorted than for the other candidate compression algorithms. The selection of a proper fractal template is critical towards controlling the quality of the compressed images, and another choice for the template may improve the VSJ image displayed in Figure 21. The optimization of fractal templates for DPIV compression is an area for future study.

Figure 22 depicts two overlaid image displacement vector maps derived from processing a representative original and fractal compressed image pair for compression levels of 1 and 4. There are similar slight vector deviations in both maps, again showing an insensitivity of introduced spatial errors to the level of compression applied. This is a remarkable result given the obviously degraded appearance of the image in Figure 21, but implies that while pixel amplitudes are being distorted with the fractal algorithm, only minimal spatial distortions are being introduced to the images. Thus, if the level of error is acceptable for a particular DPIV application and if an appropriate DPIV template image can be generated, extremely high compression levels may be achieved using fractal compression.

SUMMARY

Based on the results of this study, there are several candidate algorithms which 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 false vector rate (in some cases actually *decreasing* the rate) and imparting negligible increases in the E_u, E_v histograms. However, depending on the spatial wavenumber 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 false vector rate can be tolerated, then compressed bit rates of 0.5 – 2 bits / pixel can be achieved using JPEG or fractal compression. However, when using fractal compression, a proper template image *must* be selected if the image quality is to be optimized. 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 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 Publishers, Boston, 2000.

3. Samimy, M., and Wernet, M.P., “Review of Planar Multiple-Component Velocimetry in High-Speed Flows”, *AIAA Journal*, Volume 38, Number 4, pp. 553-574, 2000.

4. Westerweel, J., “Fundamentals of Digital Particle Image Velocimetry”, *Measurement Science and Technology*, Volume 8, pp. 1379-1392, 1997.

5. Kahler, C.J., and Kompenhans, J., “Multiple Plane Stereo PIV – Technical Realization and Fluid-Mechanical Significance”, *Proceedings of 3rd International Workshop on PIV*, Santa Barbara, CA, 1999.

6. Li, H., “Particle Image Velocimetry Based on Wavelet Image Compression Technique”, *Proceedings of 3rd International Workshop on PIV*, Santa Barbara, CA, 1999.

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*, Volume 27, pp. 310-320, 1999.

8. Cenedese, A., De Gregorio, F., Pococco, A., and Querzoli, G., “Effects of Image Compression on PIV and PTV Analysis”, *in Particle Image Velocimetry: Progress Towards Industrial Application*, Kluwer Academic Publishers, Boston, Massachusetts, pp. 429-438, 2000.

9. Ziv, J., and Lempel A., “A Universal Algorithm for Sequential Data Compression”, *IEEE Transactions on Information Theory*, Volume 23, Number 3, pp. 337-343, 1977.

10. Wallace, G.K., “The JPEG Still Picture Compression Standard”, *Communications of the ACM*, Volume 34, Number 4, pp. 31-44, 1991.

11. Fisher, Y., et.al., *Fractal Image Compression: Theory and Application*, Springer Verlag, New York, 1995.

12. Nelson, M, and Gailly, J., *The Data Compression Book*, 2nd Edition, M&T Books, New York, 1996.

13. Huang, H., Dabiri, D., and Gharib, M., “On Errors of Digital Particle Image Velocimetry”, *Measurement Science and Technology*, Volume 8, pp. 1427-1440, 1997.

14. Okamoto, K., Nishio, S., Saga, T. and Kobayashi, T., "Standard Images for Particle Imaging Velocimetry", *Proceeding of the International Workshop on PIV*, Fukui, Japan, 1997.

15. 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, 20th AIAA Advanced Measurement and Ground Testing Technology Conference*, Alburquerque, NM, 1998.

16. Soloff, S.M., and Meinhart, C.D., *CleanVec: PIV Vector Validation Software*, Version 1.13 Build 41, Released 1999.

Table 1 – Candidate Compression Algorithms Suitable for DPIV Use

Algorithm	Information Retention	Basis of Technique
LZ77	Lossless	Dictionary Encoding
LZ77 with Thresholding	“Pseudo” Lossless	Dictionary Encoding
JPEG	Lossy	Discrete Cosine Transform
Wavelet	Lossy	Discrete Wavelet Transform
Fractal	Lossy	Self-Similarity Equation

Table 2 – Image Sequences Evaluated

Sequence	Horizontal Pixels	Vertical Pixels	Image Type	Particle Image Density	Flow
VSJ 3D Standard Sequence 301	256	256	Single-Exposure, Multiple-Frame	Low	Wall Impinging Jet
Normal Incidence Impedance Tube	1320	1035	Double-Exposure, Single-Frame	Medium	Acoustically Driven, Zero-Mean Flow
Separated Flow	1300	1030	Single-Exposure, Multiple-Frame	High	Low Reynolds Number Separated Flow

Table 3 – Auto / Cross Correlation Processing Parameters

Sequence	Interrogation Size, pixels	Interrogation Size Overlap, %	Peak Detection	Image Threshold
VSJ 3D Standard Sequence 301	32 x 32	50	3-point Parabolic Fit	150
Normal Incidence Impedance Tube	128 x 128	50	3-point Parabolic Fit	150
Separated Flow	64 x 64	50	3-point Parabolic Fit	150

Table 4 – Results of LZ77 Compression Tests

Sequence	Threshold	Bit Rate, bits/pixel	Baseline FVR, %	FVR, %	PSNR, dB	$E_{u,RMS}$, pixels	$E_{v,RMS}$, pixels
VSJ 301	0	5.48	0.98	0.98	Infinity	0.0000	0.0000
	50	2.27	0.98	0.98	57.28	0.0000	0.0000
	75	1.63	0.98	0.98	49.46	0.0000	0.0000
	100	1.17	0.98	0.98	44.16	0.0000	0.0000
	125	0.82	0.98	0.98	40.31	0.0000	0.0000
	150	0.54	0.98	0.98	38.10	0.0000	0.0000
NIT	0	5.61	0.47	0.47	Infinity	0.0000	0.0000
	50	5.11	0.47	0.47	64.62	0.0000	0.0000
	75	2.69	0.47	0.47	34.02	0.0000	0.0000
	100	1.21	0.47	0.47	28.32	0.0000	0.0000
	125	0.78	0.47	0.47	26.93	0.0000	0.0000
	150	0.58	0.47	0.47	26.42	0.0000	0.0000
Sep. Flow	0	5.64	8.88	8.88	Infinity	0.0000	0.0000
	50	4.69	8.88	8.88	52.88	0.0000	0.0000
	75	3.65	8.88	8.88	40.12	0.0000	0.0000
	100	2.83	8.88	8.88	34.04	0.0000	0.0000
	125	2.13	8.88	8.88	29.03	0.0000	0.0000
	150	1.51	8.88	8.88	26.20	0.0000	0.0000

Table 5 – Results of JPEG Compression Tests

Sequence	Compress Level	Bit Rate, bits/pixel	Baseline FVR, %	FVR, %	PSNR, dB	$E_{u,RMS}$, pixels	$E_{v,RMS}$, pixels
VSJ 301	1	2.06	0.98	1.21	49.19	0.0073	0.0121
	2	1.32	0.98	1.60	44.43	0.0105	0.0174
	3	0.92	0.98	2.21	41.51	0.0131	0.0210
	4	0.79	0.98	3.42	40.51	0.0150	0.0244
	5	0.72	0.98	3.22	39.72	0.0151	0.0239
NIT	1	1.43	0.47	0.48	44.59	0.0102	0.0108
	2	0.83	0.47	0.48	44.59	0.0102	0.0108
	3	0.54	0.47	0.38	42.04	0.0120	0.0160
	4	0.46	0.47	0.58	41.09	0.0124	0.0167
	5	0.42	0.47	0.53	40.38	0.0132	0.0183
Sep. Flow	1	1.42	8.88	9.47	37.83	0.0113	0.0079
	2	0.86	8.88	10.47	34.56	0.0133	0.0084
	3	0.58	8.88	11.37	32.34	0.0184	0.0110
	4	0.49	8.88	11.70	31.50	0.0220	0.0123
	5	0.46	8.88	12.33	30.86	0.0186	0.0116

Table 6 – Results of Wavelet Compression Tests

Sequence	Quality Level	Bit Rate, bits/pixel	Baseline FVR, %	FVR, %	PSNR, dB	$E_{u,RMS}$, pixels	$E_{v,RMS}$, pixels
VSJ 301	1	3.59	0.98	0.53	39.99	0.0098	0.0173
	2	1.89	0.98	0.63	40.37	0.0097	0.0170
	3	0.80	0.98	0.86	39.48	0.0133	0.0209
	4	0.40	0.98	3.54	36.81	0.0240	0.0329
NIT	1	4.04	0.47	0.38	50.67	0.0078	0.0114
	2	1.96	0.47	0.38	50.12	0.0081	0.0117
	3	0.78	0.47	0.45	46.19	0.0091	0.0123
	4	0.40	0.47	0.48	41.50	0.0124	0.0152
Sep. Flow	1	4.00	8.88	6.76	38.60	0.0132	0.0082
	2	1.93	8.88	6.75	37.45	0.0140	0.0084
	3	0.81	8.88	7.40	33.69	0.0141	0.0088
	4	0.40	8.88	8.57	30.80	0.0177	0.0112

Table 7 – Results of Fractal Compression Tests

Sequence	Quality Level	Bit Rate, bits/pixel	Baseline FVR, %	FVR, %	PSNR, dB	$E_{u,RMS}$, pixels	$E_{v,RMS}$, pixels
VSJ 301	1	2.49	0.98	38.85	34.49	0.0925	0.1296
No Template	2	1.85	0.98	36.97	34.51	0.1680	0.2244
	3	1.49	0.98	37.42	34.38	0.1385	0.1773
	4	0.38	0.98	51.60	34.20	0.5840	0.6911
VSJ 301	1	2.50	0.98	4.38	33.18	0.0219	0.0338
Template	2	1.87	0.98	4.30	33.20	0.0221	0.0335
	3	1.47	0.98	4.63	33.21	0.0220	0.0334
	4	1.46	0.98	4.49	33.26	0.0223	0.0326

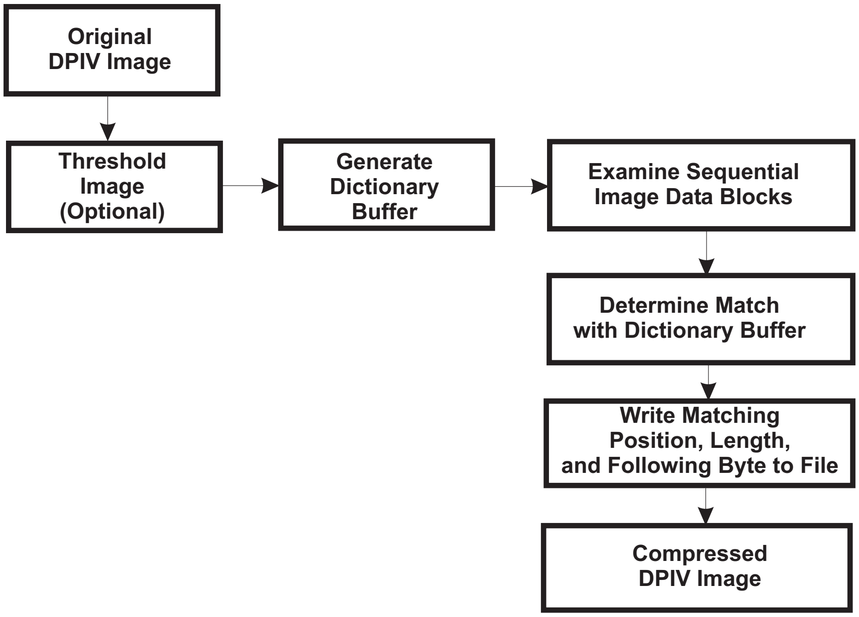
Original DPIV Image
Threshold Image (Optional)
Generate Dictionary Buffer
Examine Sequential Image Data Blocks
Determine Match with Dictionary Buffer
Write Matching Position, Length, and Following Byte to File
Compressed DPIV Image

Figure 1. LZ77 Compression.

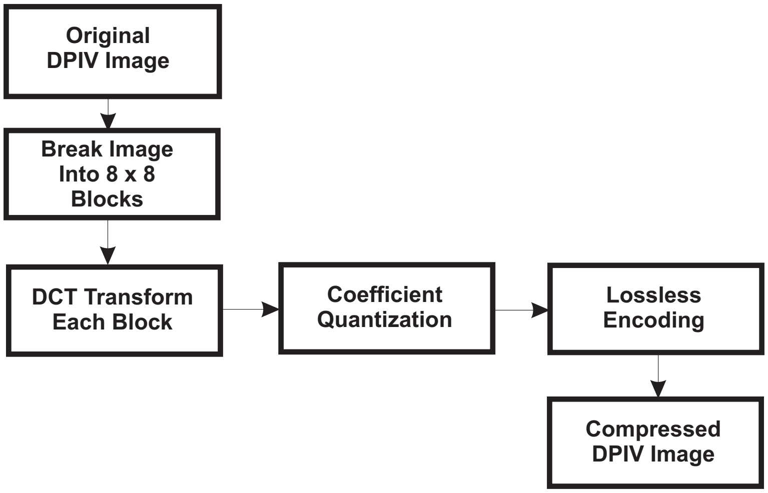

Figure 2. JPEG Compression.

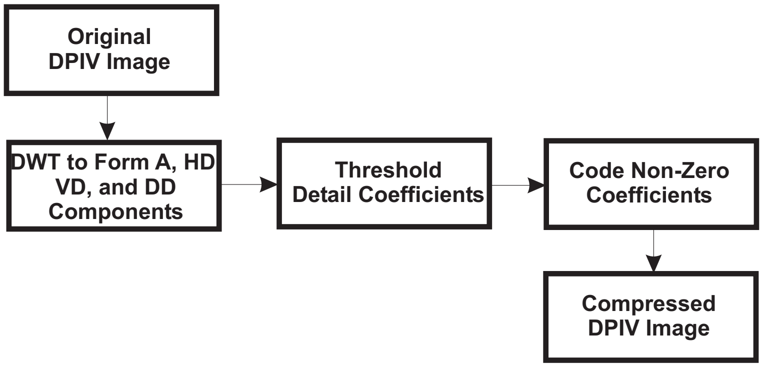
Original DPIV Image
DWT to Form A, HD VD, and DD Components
Threshold Detail Coefficients
Code Non-Zero Coefficients
Compressed DPIV Image

Figure 3. Wavelet Compression.

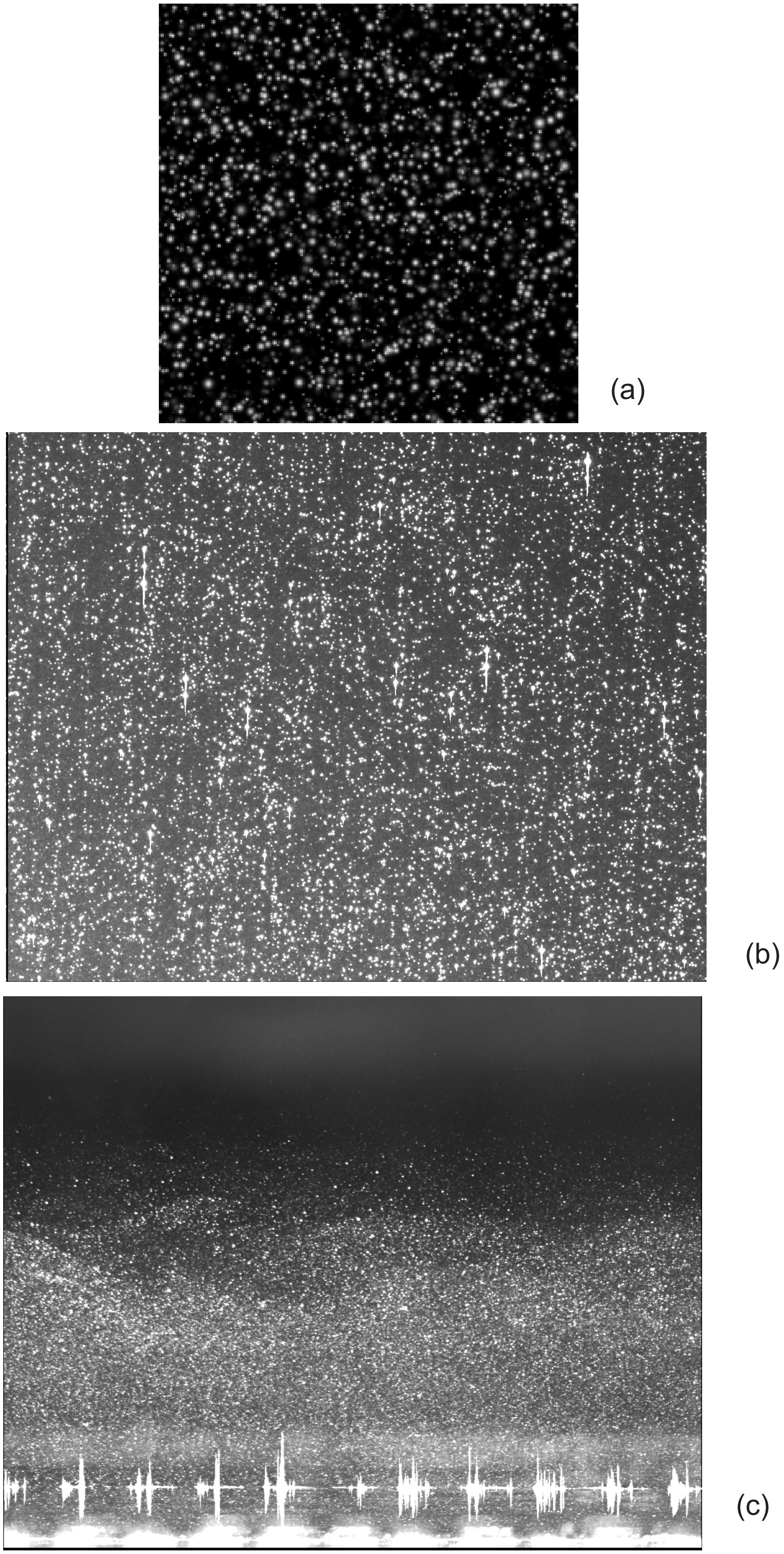

Figure 4. Test Sequence Sample Images.
(a) VSJ Industry Standard Sequence
(b) LaRC Normal Incidence Impedance Tube
(c) LaRC Separated Flow

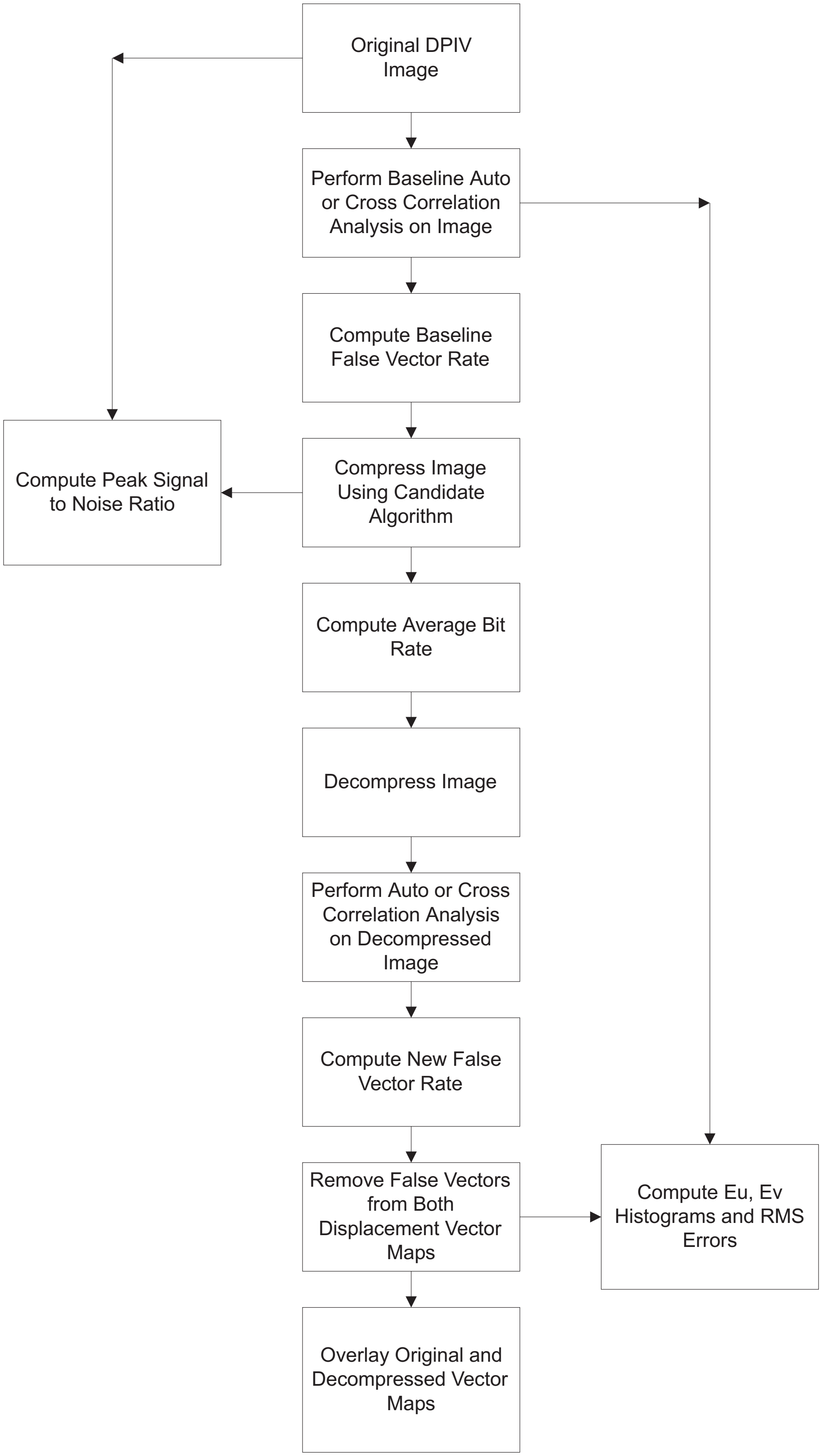
Original DPIV Image
Perform Baseline Auto or Cross Correlation Analysis on Image
Compute Baseline False Vector Rate
Compute Peak Signal to Noise Ratio
Compress Image Using Candidate Algorithm
Compute Average Bit Rate
Decompress Image
Perform Auto or Cross Correlation Analysis on Decompressed Image
Compute 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

Figure 5. Compression Algorithm Evaluation Procedure.

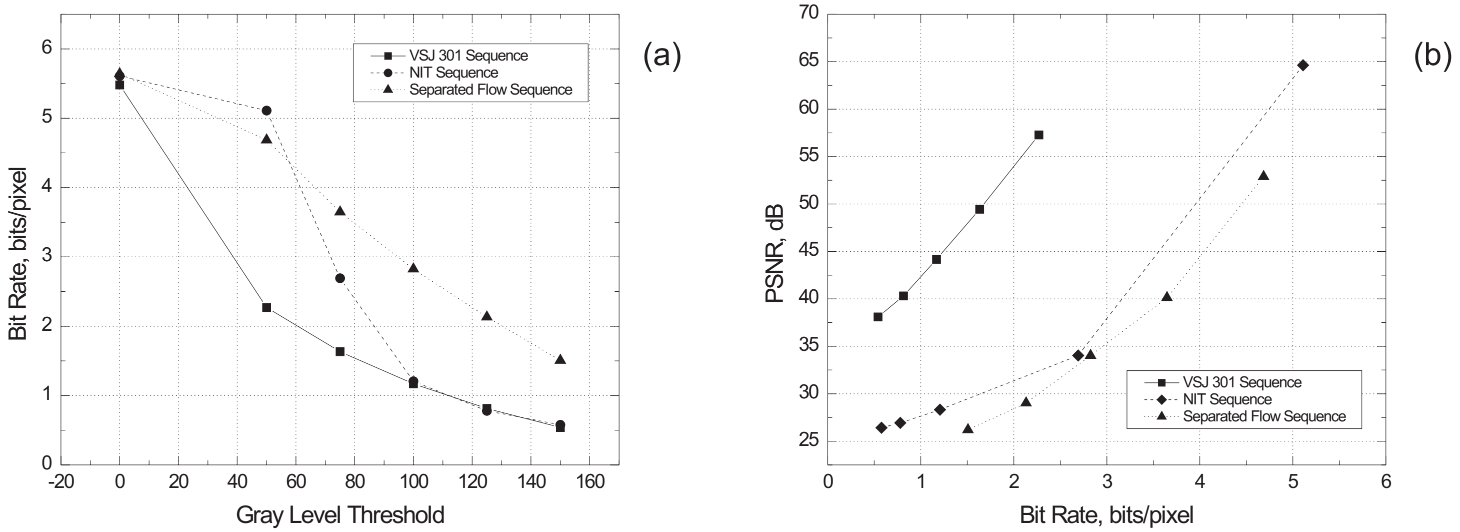

Figure 6. LZ77 Compression Results.
(a) Bit Rate vs Image Threshold Level
(b) LZ77 PSNR vs Bit Rate

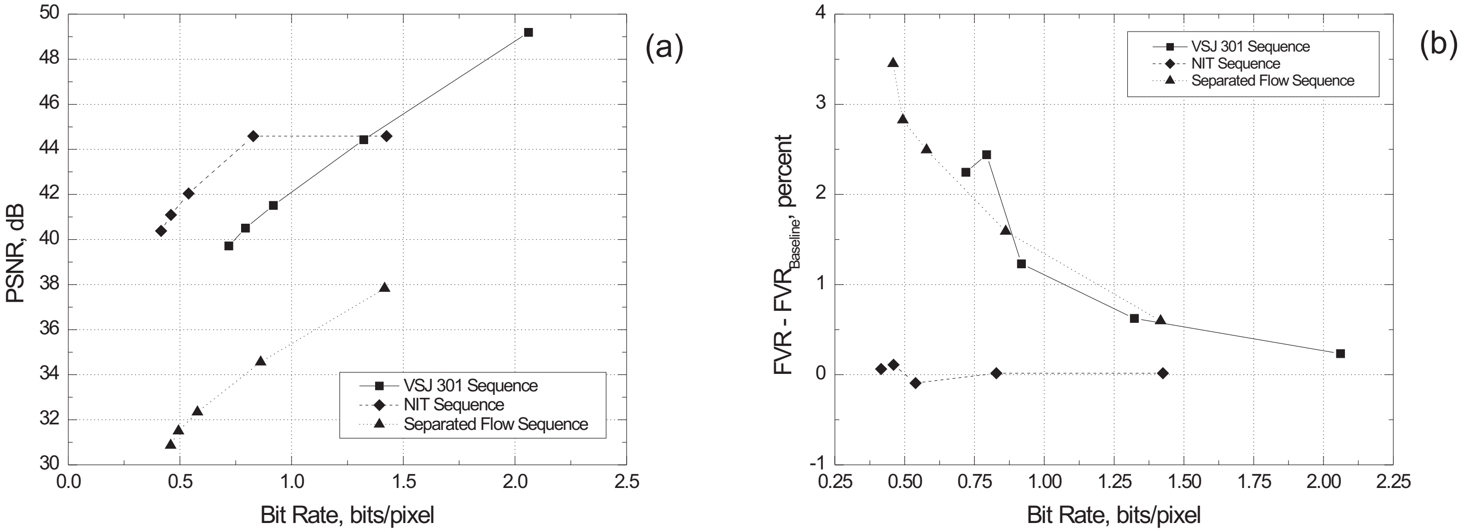

Figure 7. JPEG Compression Results.
(a) PSNR vs Bit Rate
(b) JPEG Change in FVR vs Bit Rate

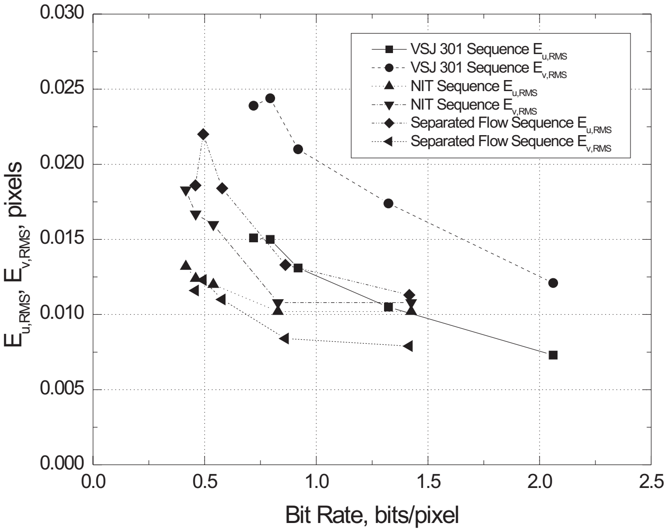

Figure 8. JPEG RMS Spatial Errors.

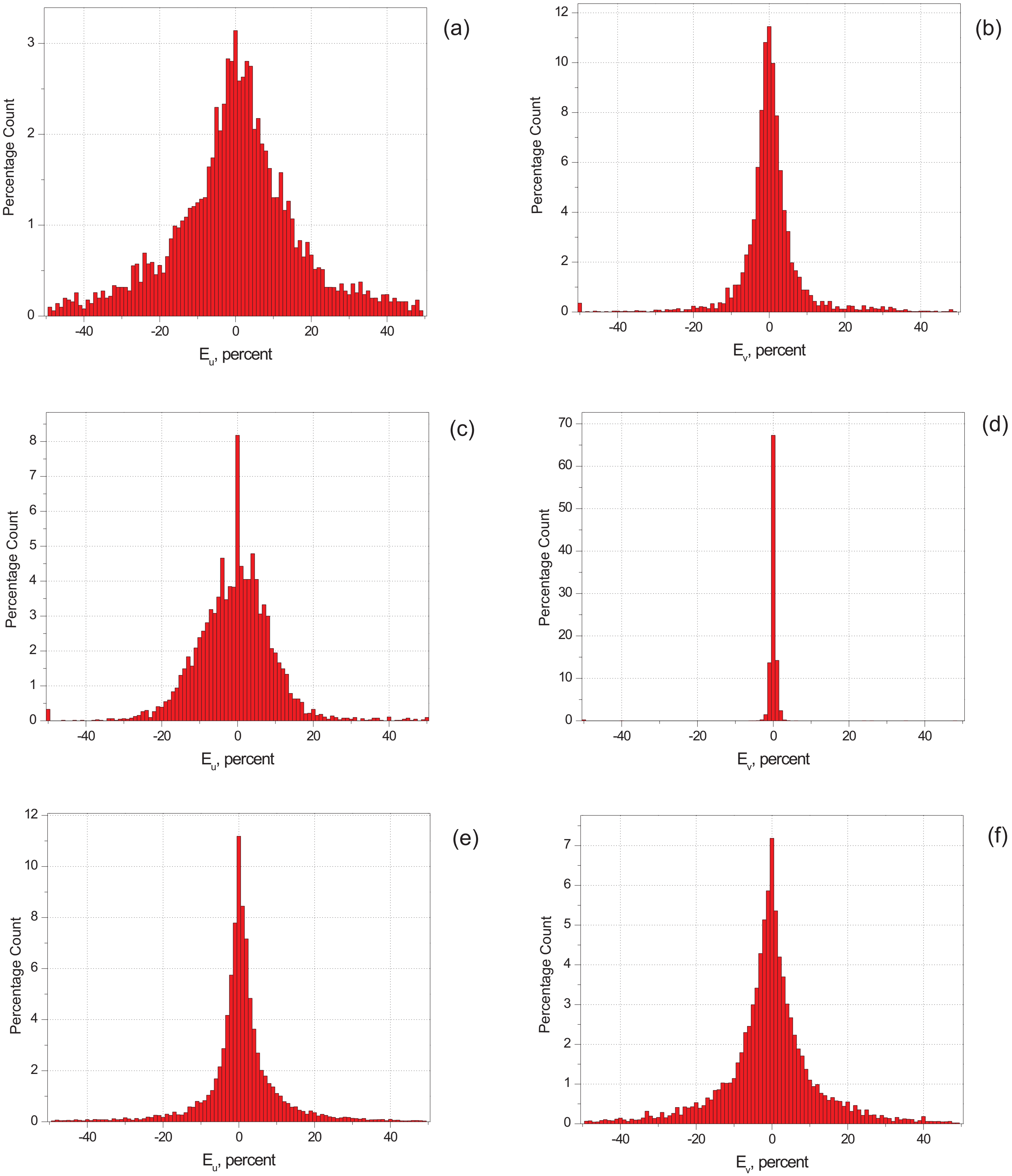

Figure 9. Representative JPEG Error Histograms - Compression Level 5.
(a) VSJ Sequence, E_u (b) VSJ Sequence, E_v
(c) NIT Sequence, E_u (d) NIT Sequence, E_v
(e) Separated Flow Sequence, E_u (f) Separated Flow Sequence, E_v

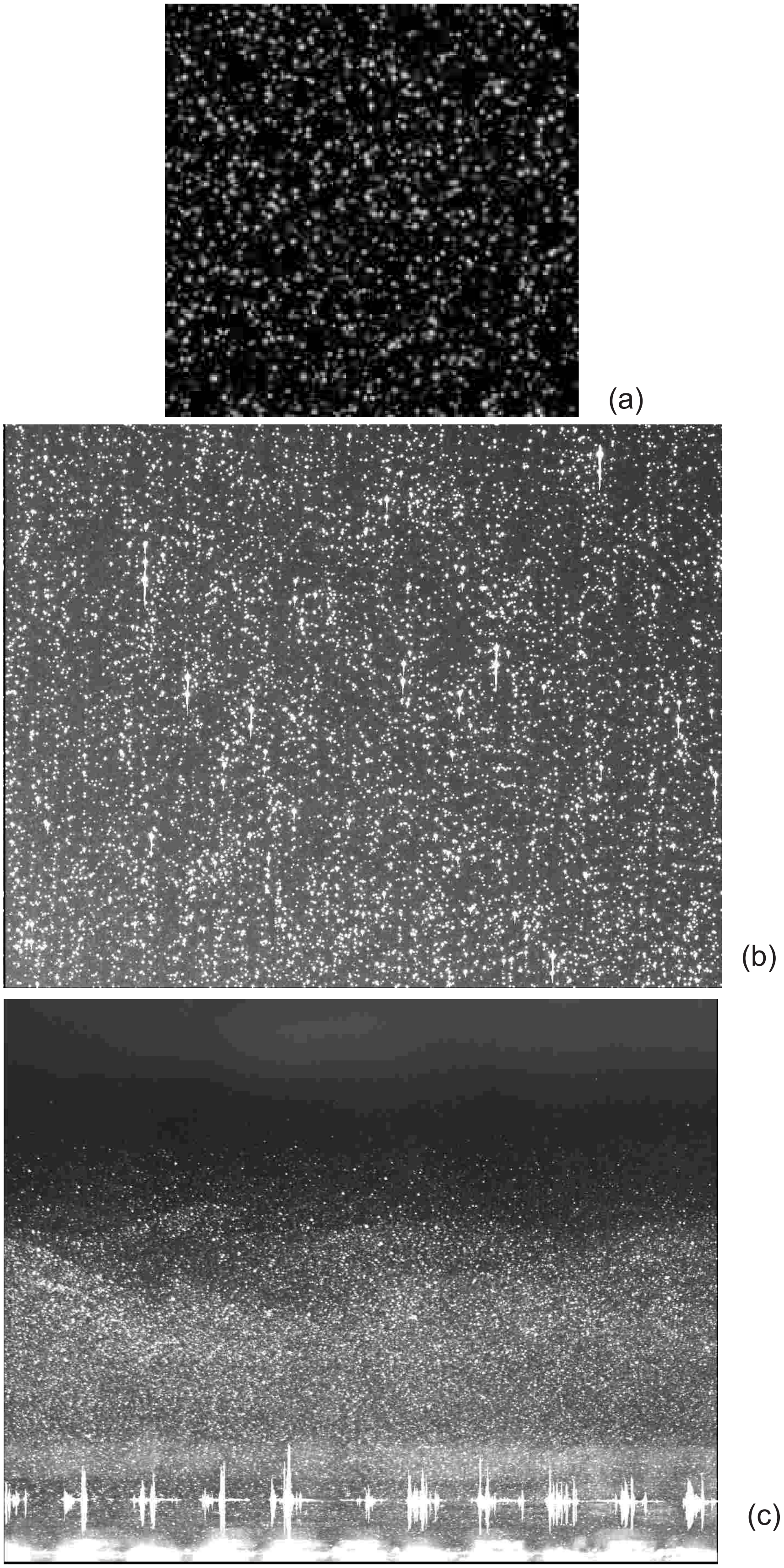

Figure 10. JPEG Compressed Images - Compression Level 5.
(a) VSJ Standard Sequence
(b) LaRC Normal Incidence Impedance Tube
(c) LaRC Separated Flow

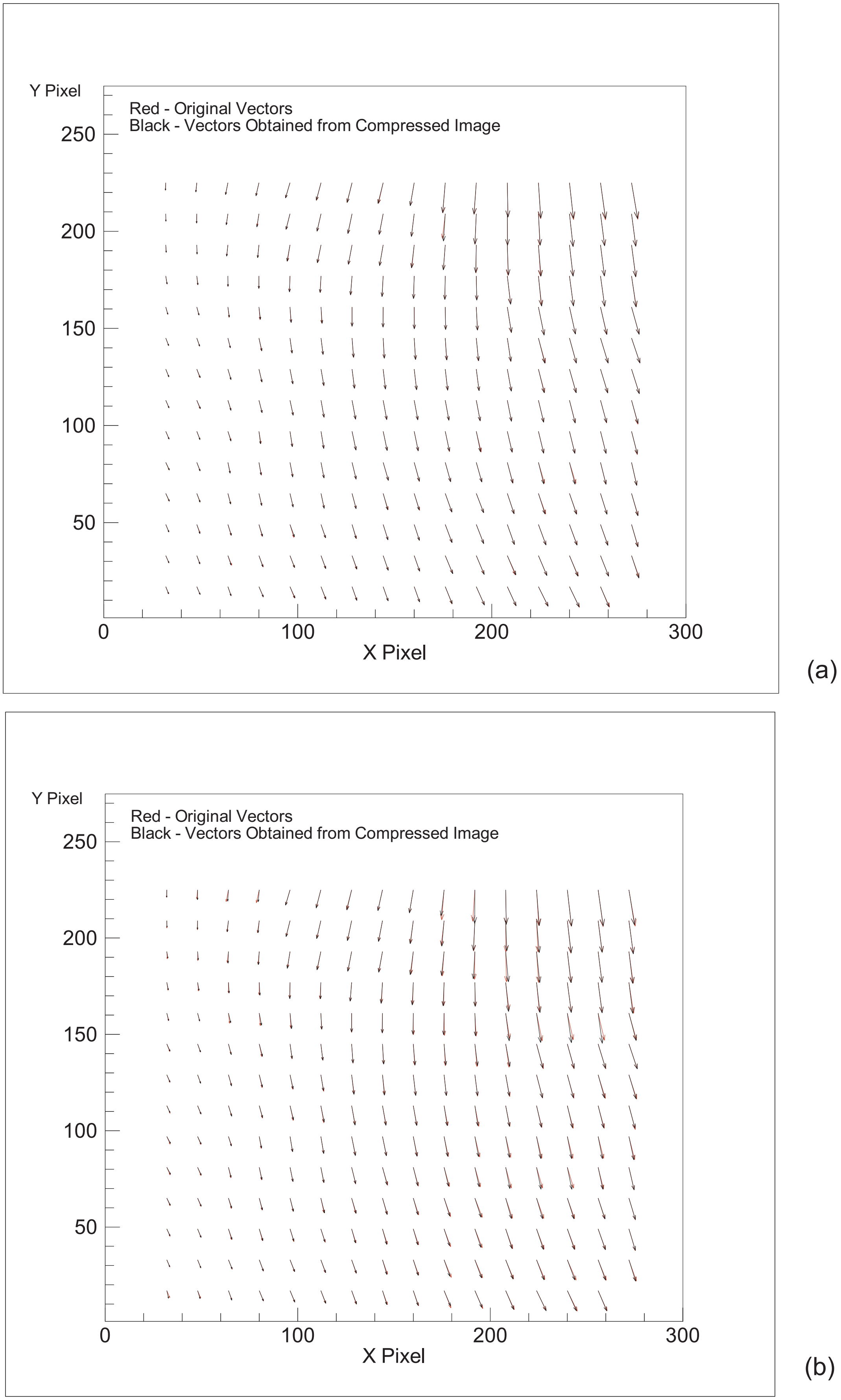

Figure 11. JPEG Displacement Vector Maps - VSJ Sequence.
(a) Compression Level 1
(b) Compression Level 5

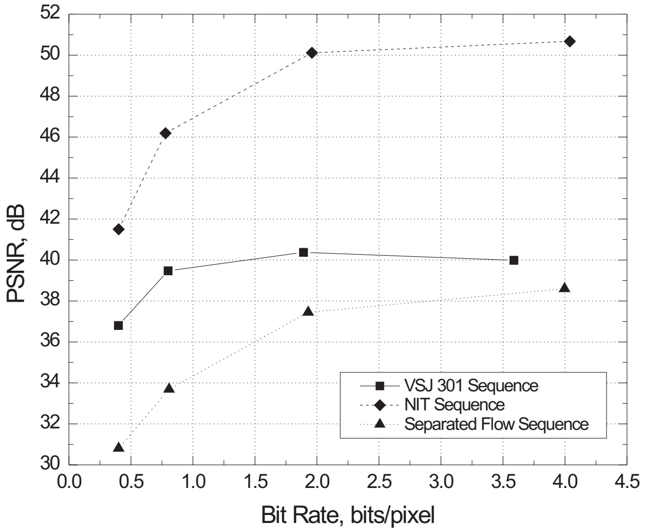

Figure 12. PSNR vs Bit Rate - Wavelet Compression.

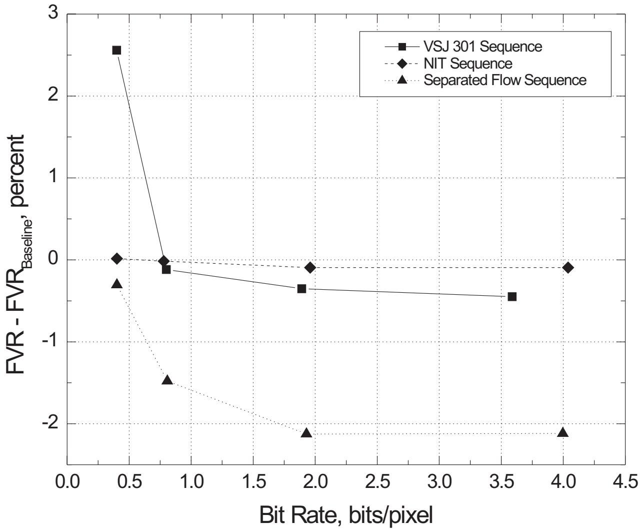

Figure 13. Change in FVR vs Bit Rate - Wavelet Compression.

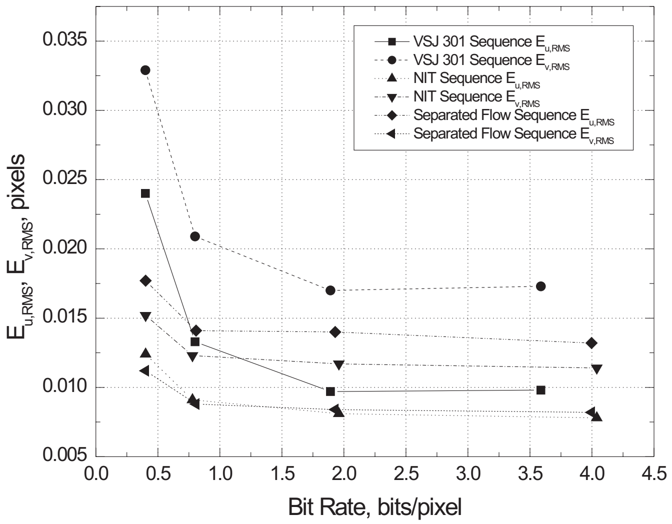

Figure 14. RMS Spatial Errors - Wavelet Compression.

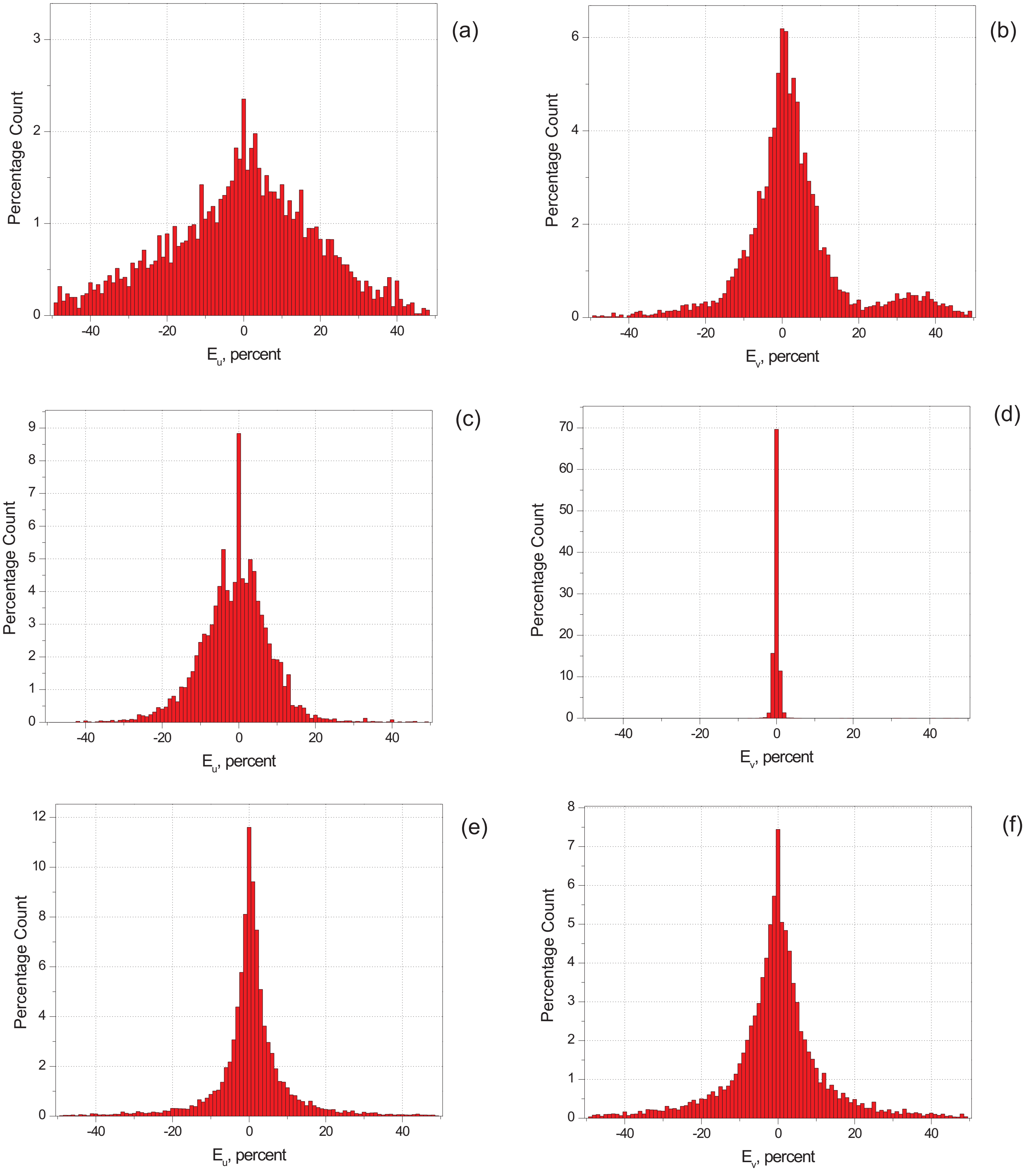

Figure 15. Representative Error Histograms - Wavelet Compression Level 4.
(a) VSJ Sequence, E_u (b) VSJ Sequence, E_v
(c) NIT Sequence, E_u (d) NIT Sequence, E_v
(e) Separated Flow Sequence, E_u (f) Separated Flow Sequence, E_v

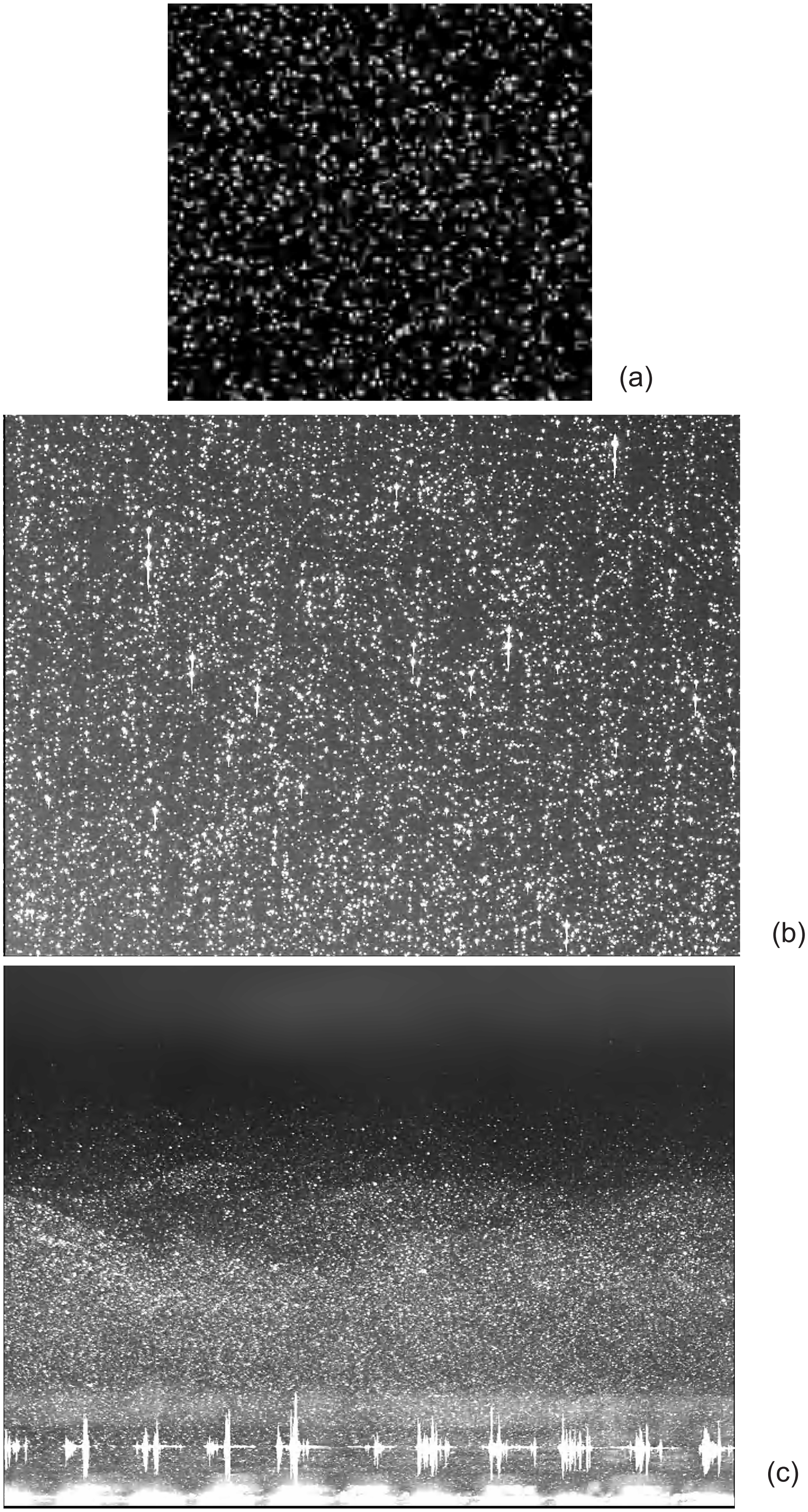

Figure 16. Wavelet Compressed Images - Compression Level 4.
(a) VSJ Standard Sequence
(b) LaRC Normal Incidence Impedance Tube
(c) LaRC Separated Flow

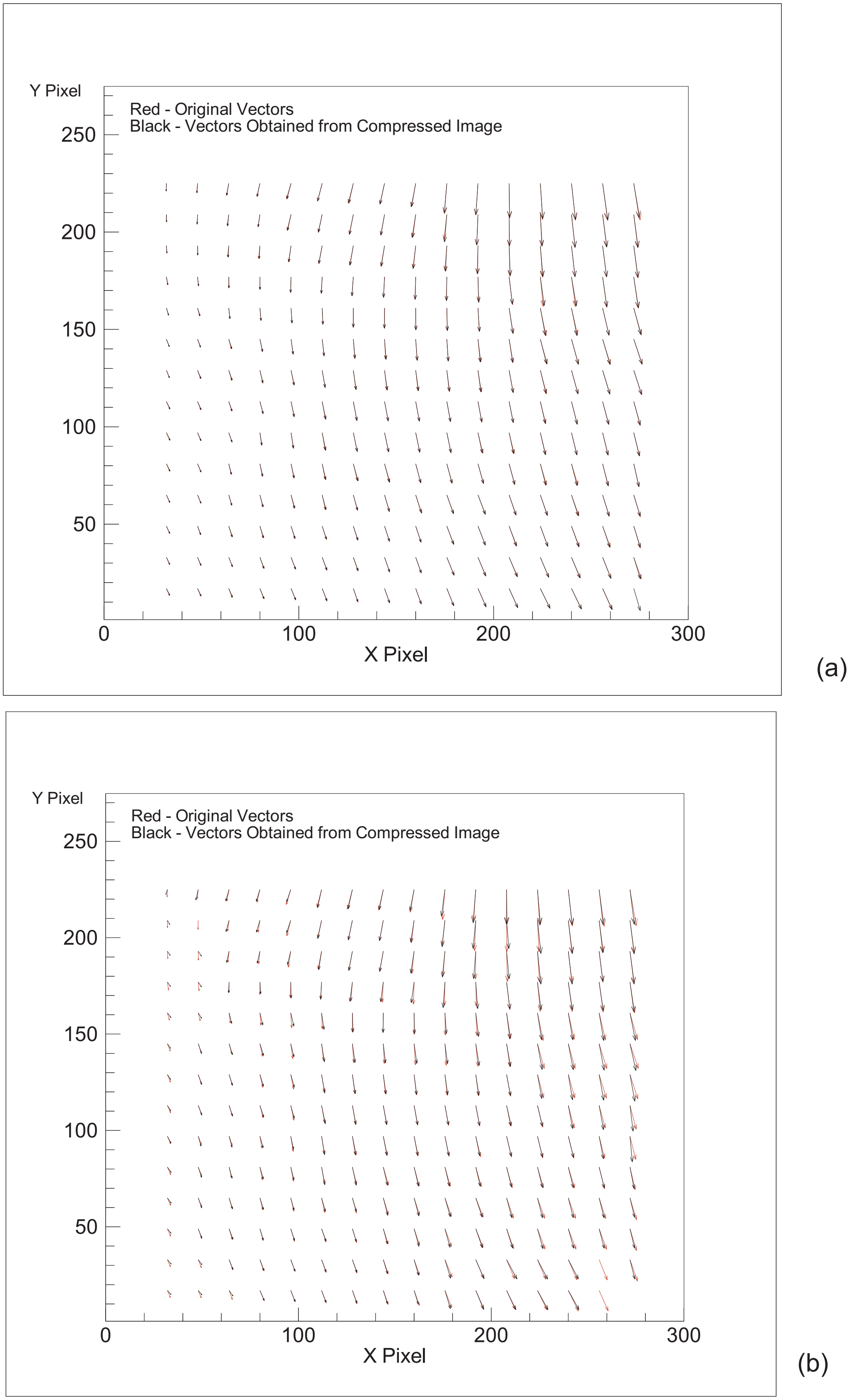

Figure 17. Wavelet Displacement Vector Maps - VSJ Sequence.
(a) Compression Level 1
(b) Compression Level 4

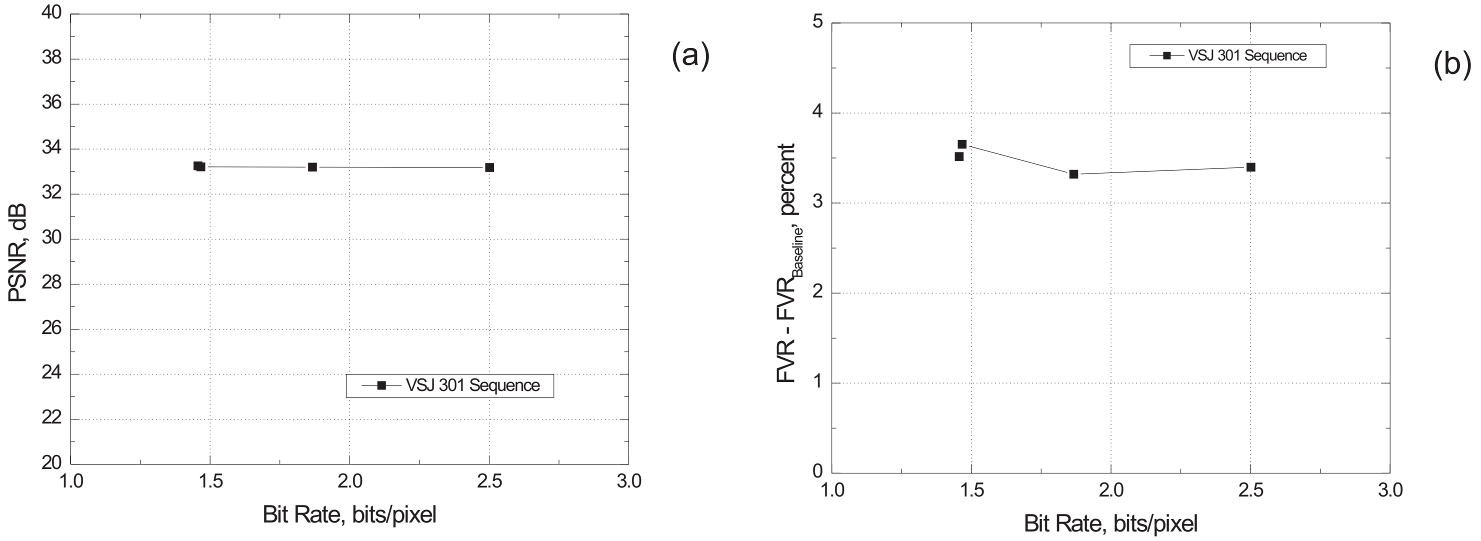

Figure 18. Fractal Compression Results.
(a) PSNR vs Bit Rate
(b) Change in FVR vs Bit Rate

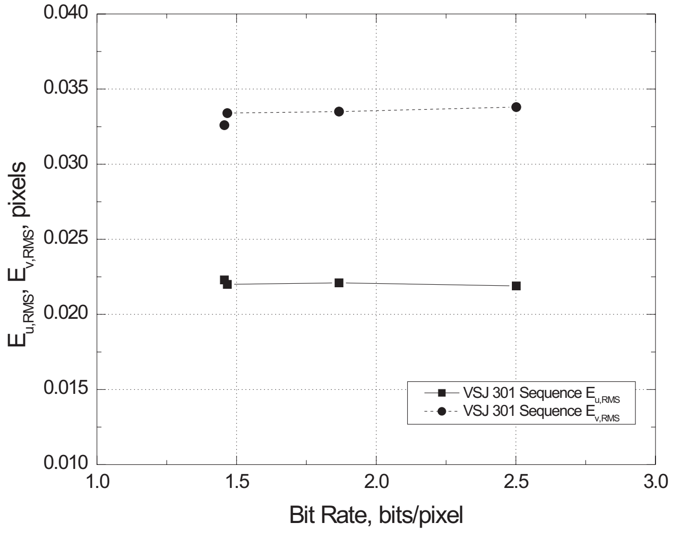

Figure 19. RMS Spatial Errors - Fractal Compression.

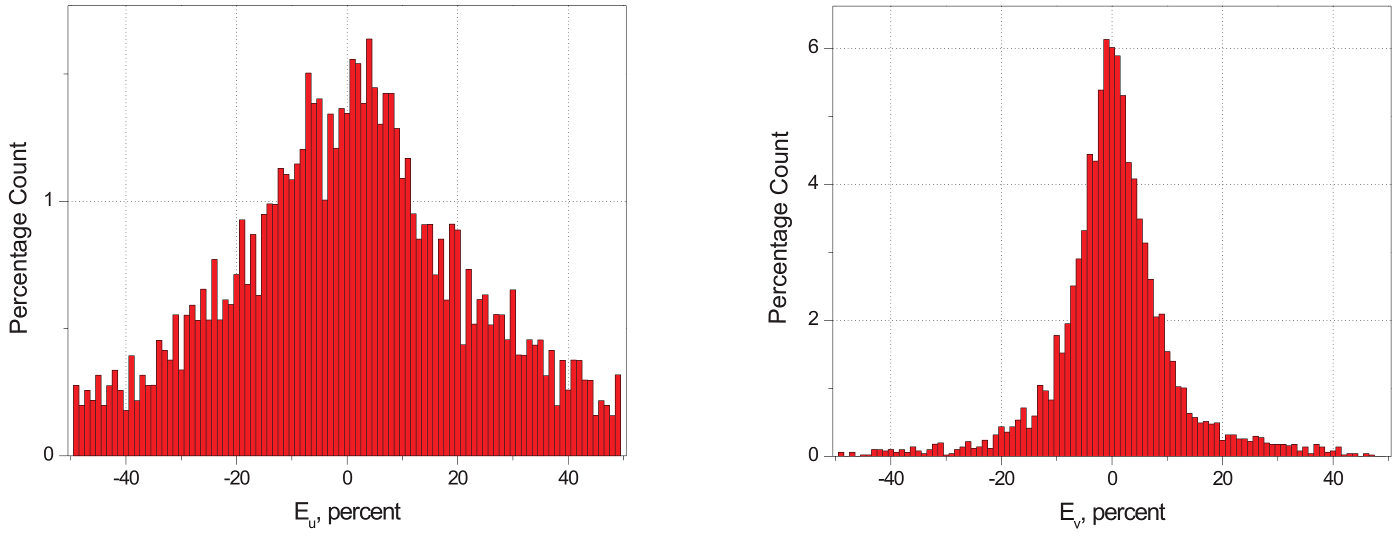

Figure 20. Representative Error Histograms - Fractal Compression Level 4.
VSJ Standard Sequence

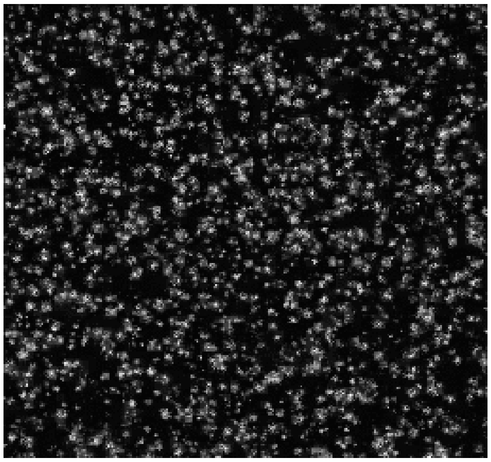

Figure 21. Fractal Compressed Image - Compression Level 4.
VSJ Sequence using DPIV Image Template

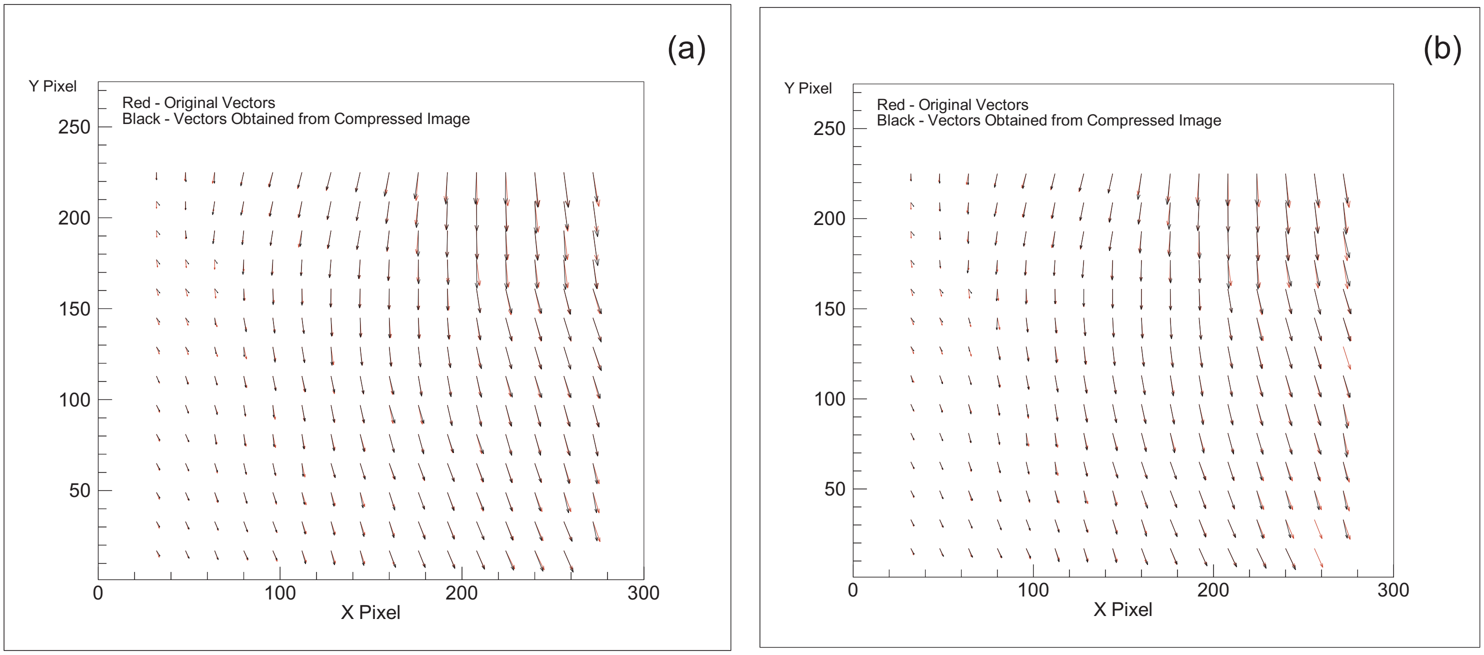

Figure 22. Fractal Displacement Vector Maps - VSJ Sequence.
(a) Compression Level 1 (b) Compression Level 4