STEVEN D. DIAMOND

DETERMINATION OF FLUID VELOCITY FIELDS WITH PARTICLE DISPLACEMENT VELOCIMETRY

The determination of fluid velocities in experiments is fundamental to the theoretical advances and engineering applications of fluid dynamics. Two-dimensional velocity fields can be noninvasively measured by particle displacement velocimetry, in which small particles are introduced into the flow and correlation analysis is performed on multiply exposed images of the particles.

INTRODUCTION

The fundamental problem of fluid dynamics is to understand the change over time of the velocity of a moving fluid at each point in space. This problem is so complex that enormous amounts of analytical, experimental, and, most recently, computational effort have been expended without producing anything like a complete solution. The problem can be concisely summarized in the Navier–Stokes equations, a set of coupled, second-order, nonlinear partial differential equations derived by applying Newton's laws of motion to a typical fluid volume element, together with certain basic assumptions about the nature of fluid stresses and about viscosity coefficients for shearing and longitudinal stresses. Various simplifying assumptions, such as incompressibility, negligible viscosity, and lack of vorticity, can be made to allow restricted analytical solutions of the Navier–Stokes equations, and indeed many early engineering advances in fluid flow and aerodynamics resulted from this approach. Current engineering applications seldom derive from a strictly analytical base, however.

Computational fluid dynamics (CFD), enabled by modern supercomputers, allows prediction and analysis of high-velocity turbulent flows important in many applications. Tuning of CFD codes to particular applications, as well as further development of computational methods, still heavily depends on comparisons with experiments. In addition, the recent explosion of interest in finite-dimensional nonlinear dynamics, together with the suggestive nature of solutions to even low-dimensional chaotic systems, has led to the hope that nontraditional approaches may give new life to the analytical study of fluid flows.[1] These approaches must first survive confrontation with experiment. The measurement of fluid velocities in an experimental context is thus essential to further development of fluid mechanics from several viewpoints.

The Laboratory's Milton S. Eisenhower Research Center is pursuing both the development of computational methods and the application of chaos theory[2] to fluids as a result of an initiative in its most recent strategic plan, *Research Center 2000*. An experimental component of this effort (involving, besides the author, Research Center staff members Harold E. Gilreath, Charles H. Hoshall, and John C. Sommerer, and Submarine Technology Department staff member Joseph E. Hopkins) depends on the ability to measure fluid velocities; fluid velocity measurement is also important in the work of several other APL departments.

FLUID VELOCITY MEASUREMENT APPROACHES

Experimental fluid velocity measurement has a long history, and many distinct approaches have been developed. Perhaps the earliest method is hot-wire anemometry, where current-carrying thermistor wires are introduced into the moving fluid. Determination of the fluid velocity at the probe is possible using the overall heat transfer to the fluid (from the electrical current passed), temperature of the fluid, probe geometry, and heat transfer theory. This method, although instrumentally simple, perturbs the flow that it is designed to measure, particularly if the fluid velocity is needed at a whole field of spatial locations, as is usually desirable.

At the other end of the spectrum is the noninvasive technique, laser Doppler velocimetry, where the fluid velocity at a point in the fluid is inferred from the Doppler shift in the laser light scattered from the fluid itself. The instrumentation for this method, however, is relatively complicated and expensive, so measuring the velocity at many points in the fluid simultaneously is impractical (time multiplexing can partially solve this constraint but still requires duplication of the precisely configured beam-routing optics).

A newer technique, which is largely noninvasive but allows the determination of velocity simultaneously at many points in the fluid, is particle displacement velocimetry (PDV). In this technique, tiny density-matched particles are introduced into the fluid and are carried passively by the flow field. The motion of the particles is assumed to reflect that of the surrounding fluid and allows determination of the fluid velocity, in principle, wherever there is a particle. Particle displacement velocimetry has been applied successfully in a number of forms, including streak analysis[3] (where a time exposure reveals the particle trajectories as streaks), holography[4] (which allows full three-dimensional information to be obtained), and correlation analysis,[5] which is the approach we pursued.

In correlation analysis, a multiply exposed picture of the particle-laden fluid shows displaced pairs of particle images. The correct correlation of particle images allows determination of the displacement of a particle over the interval between short, nonstreaking exposures, and hence determination of the particle's velocity, as well as that of the surrounding fluid.

Correlation analysis has been applied using fine-grained photographic media, allowing extremely high precision in velocity determination, at the cost of a great deal of analog-to-digital conversion and subsequent computation. Simply replacing the photographic media with a digital imager removes the analog-to-digital conversion, replacing it with a prohibitive data storage problem. We chose to pursue a lower-accuracy, near-real-time, all-digital approach.

The experiments being conducted in the Research Center (a two-dimensional jet in a stratified fluid and a cylinder towed perpendicular to its symmetry axis in a stratified fluid) produce essentially two-dimensional velocity fields. The analysis of these velocity fields is beyond the scope of this article, but it requires many individual instances that collectively capture the overall temporal statistical characteristics of the flows while preserving spatial relationships between distant points in the fluid. Thus, PDV presented the only real option for an experimental technique. However, the need for many samples would create a serious data problem in either analog or digital form if the approach were to collect images for subsequent analysis. The specific analysis planned for these data is not particularly sensitive to small velocity errors, thus allowing us to envision a low-accuracy real-time system, where we only need to store the reduced velocity field rather than raw imagery, with a consequent huge reduction in required data storage capacity. This approach appeared to be marginally feasible at the project's start and might not have been warranted for our applications alone. We realized, however, that the exponential improvement in computational resources would soon allow for the generalization of our system to higher-accuracy requirements, and felt that we could therefore begin something with significant general utility and much greater convenience than available systems, in addition to meeting our present, somewhat specialized needs.

The following sections describe the acquisition of particle displacement data, the algorithm on which our PDV system is based, our approach to developing and testing it, and some considerations regarding its implementation.

TYPICAL EXPERIMENT

A towed cylinder experimental setup is shown in Fig. 1. A platform containing a charge-coupled device (CCD) camera is attached to the top of the vertical cylinder in the tank. To the side of the tank are a light source and a shutter that interrupts and disperses the light. The light passes through the shutter, reflects off the particles in their density-matched fluid layer, and impinges on the camera. The camera's field of view is roughly centered on the cylinder and is sufficiently broad to include all salient flow locations.

Once the particles have been introduced into the stratified tank and the fluid has stabilized, the camera is prepared for image acquisition. The cylinder is towed through the tank to create the desired velocity flow, the camera is triggered, and the shutter is opened twice to generate the double-exposure image in the camera. This image is transferred to the computer for analysis.

A portion of a typical image is shown in Fig. 2. The brighter white dots represent particles illuminated in the

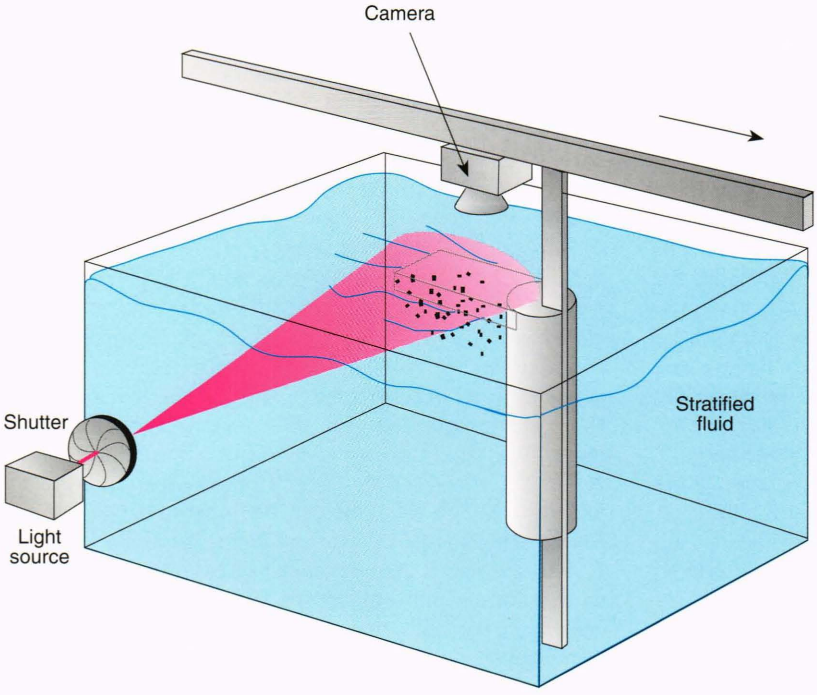


Figure 1. Towed cylinder experimental setup.

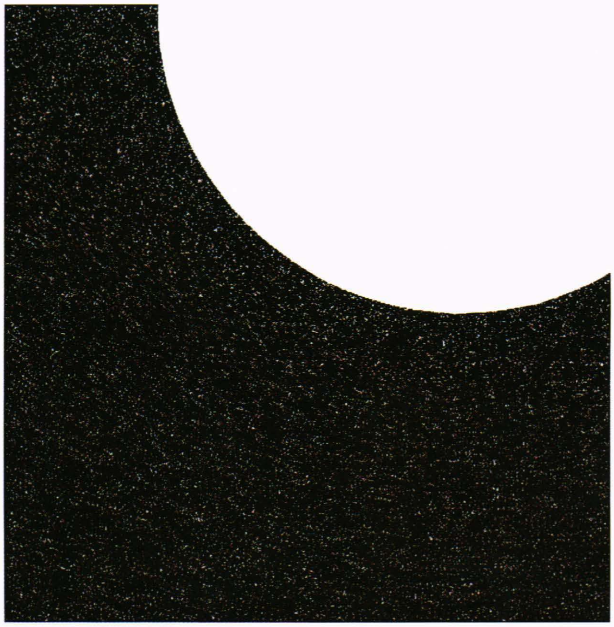


Figure 2. A simulated velocity flow in a fluid, corresponding to the experimental setup in Fig. 1. For clarity, only the lower left quadrant of the image is shown. The upper right of the image is a side view of the simulated cylinder, moving to the left. The white dots are double exposures of particles in the liquid, whose small position displacements between exposures are caused by the movement of the cylinder. Approximately 10% of the image pixels contain particles.

first exposure. Adjacent to most of the bright dots is a dimmer dot, representing the second shorter exposure of the associated particle. The cylinder is shown in the upper right portion of the figure. The rest of the image (background) is nearly black (zero intensity); nonzero intensities result from reflections and noise-related effects. From the time interval between exposures and the separations between bright and dim dots, one can calculate the average velocity of the particles between exposures. This velocity calculation is performed at regularly spaced locations throughout the velocity flow.

PDV ALGORITHM

The camera produces a digitized image of 1280 × 1024 pixels. Within the attached computer, this image is stored as a large, two-dimensional integer array. We can conceptualize the correlation analysis as follows: Initially, for simplicity, assume that a particle is completely contained within a pixel and that no more than one particle will lie within a pixel, i.e., a given pixel will either have a low intensity (no enclosed particle), a high intensity (an enclosed particle at its initial position), or an intermediate intensity (an enclosed particle at its final position).

The complete image is divided into subimages. One velocity estimate will be calculated for each subimage, and we assume that the velocity of all particles in the subimage is the same. Each subimage can be compared with copies of itself that have been shifted horizontally and/or vertically by all possible amounts (up to half the subimage size). With the appropriate shift, the bright dots of one copy will align with the dim dots of the other copy of the subimage. This shift is proportional to the average particle velocity in the subimage. A software program could perform this alignment comparison by multiplying each subimage pixel (intensity) by the corresponding pixel in the shifted subimage and accumulating the products. For a shift of r rows and c columns in any $N \times N$ subimage, the sum is

$$\mathrm{Sum}(r,c) = \sum_{i=0}^{N} \sum_{j=0}^{N} S(i,j)S(i+r,j+c),$$

where $S(i, j)$ is the pixel intensity at row i, column j. The shift corresponding to the greatest sum represents a displacement corresponding to the average particle velocity in the subimage.

This technique is very simple but extremely inefficient. A subimage of size $N \times N$ pixels requires N^4 multiplications and additions (operations), or approximately 2.2×10^{10} operations for the total image, assuming a subimage size of 64×64 pixels, with adjacent subimages overlapped by 32 pixels. Even at a high computation rate of 50 million floating point operations per second, the basic correlation calculation requires 8 min, very far from our goal of near-real-time analysis unless the flows to be measured are ridiculously slow.

To speed the calculations, we use the well-known two-dimensional fast Fourier transform (2D FFT) technique. In that approach, correlation is performed via the convolution theorem, by taking the 2D FFT of a subimage, multiplying the transform by its complex conjugate, and then taking the inverse 2D FFT of the product. The 2D FFT approach requires approximately $2N^2\log_2 N$ operations for an $N \times N$ subimage, and is therefore potentially 600 times faster than the brute force approach for the image parameters already discussed. This comparison is oversimplified, ignoring the extensive additional processing that must be performed to produce the ultimate correlation results; the actual improvement for the entire correlation process is approximately a factor of 20.

The size of the actual subimages is a compromise among several factors. Because a randomly positioned particle can appear in several image pixels, a more accurate velocity estimate is obtained when more particle images are included in the subimage (as for a larger subimage or higher particle density) and when the average movement of a particle traverses a large number of pixels (which requires a large subimage). However, both exposures of a high-velocity particle are less likely to be contained in a subimage. Real velocity fields are also not constant over arbitrarily large subimages; the maximum fluid acceleration between the two exposures limits the maximum subimage size. For our simulated velocity fields, subimages of 64 or 128 pixels squared were selected, although this size can obviously change for different experiments or as computers improve.

Particle size is also a compromise, but between velocity resolution and total image size. The location of the center of a particle can be more accurately determined for a multipixel particle. However, large particles are more likely to overlap, requiring a reduced particle density (fewer particles in a fixed area), which reduces velocity accuracy. Excessively large particles also perturb the underlying flow. One disadvantage of small particles that are less than a pixel in diameter is that they produce a lower-intensity pixel. This reduced intensity can be compensated for by higher illumination and a longer exposure that is still short enough to avoid streaking. The exact location of subpixel particles within a pixel is also frequently unknown. For example, a particle whose diameter is $^{1}/_{4}$ of a pixel diameter can move $^{3}/_{8}$ of a pixel horizontally and/or vertically from a centered position without changing the pixel intensity. When a particle is near the edge of a pixel, its intensity is divided between that pixel and the adjacent pixel. By comparing the relative intensities of these pixels, the location of the particle center can be approximately calculated. This subpixel accuracy would apply to about 40% of the small particles.

The result of an $N \times N$ subimage correlation calculation is an $N \times N$ two-dimensional array A, with row and column indices ranging from 0 to $N - 1$. The value at row r and column c [$A(r, c)$] represents the "goodness" of correlation, assuming that the particles all moved r pixels vertically and c pixels horizontally between exposures. A maximum correlation occurs at (0, 0) and is ignored (an image correlates perfectly with an unshifted version of itself). The location of the *second* highest correlation (SHC) represents the average particle velocity.

Particles move horizontally or vertically in positive or negative directions. In the array A, row and column values from 0 to $N/2 - 1$ represent progressively higher positive velocities. Row and column values from $N/2$ to $N - 1$ represent progressively smaller (in absolute value) negative velocities. This velocity discontinuity at $N/2$ complicates subsequent analysis, so the correlation values are shuffled to produce a continuous velocity from 0 to $N - 1$, and the zero velocity point is at the location $(N/2, N/2)$ in the "center" of the array.

The approach just discussed must be modified in practice for several reasons. Various noise sources in the CCD camera and associated electronics, as well as stray light in the experiment, produce nonzero pixel values (where zero represents a totally black pixel), even when viewing a particle-free image. This noise, combined with the particle position uncertainty and particle overlaps, produces substantial correlation values at arbitrary locations throughout the image. A high particle density also exacerbates this problem. For example, if 10% of the image pixels contain particles, there is a 10% probability that a particle pixel in an original subimage will coincide with an unrelated particle pixel in an arbitrarily shifted subimage. Thus, on average, each correlation value $A(r, c)$ will be 10% of the maximum correlation value $A(0, 0)$. Fortuitous excursions to two or three times this average value are not unusual. The SHC value (the one we are looking for) would be about 60% of the maximum if it were contained in just $A(r_{\mathrm{SHC}}, c_{\mathrm{SHC}})$. However, because of the effects just mentioned, the SHC value is divided among $A(r_{\mathrm{SHC}}, c_{\mathrm{SHC}})$ and neighboring array elements. This situation may be visualized by displaying the correlation array in three-dimensional space, where the row and column are the coordinates on the two-dimensional plane, and the correlation value is the height above the plane. Correlation of a simulated image under ideal conditions produces a sharp peak at (0, 0) (deleted for clarity) and radially symmetric peaks at the SHC location (Fig. 3). Most of the other pixels have nonzero values from overlaps of particles in the original subimage, with unrelated particles in the shifted subimage. In a correlation array based on experimental data, the sharp peaks become more rounded and must be carefully discriminated from the hills and peaks produced by the overlaps, noise, and other factors.

To reduce the likelihood of mistaking a randomly produced peak for the SHC peak, the ratio of the SHC value to the next highest peak must exceed a threshold. Once an SHC peak has been verified, the velocity accuracy is improved to subpixel resolution by calculating a proportional average of the SHC peak with the array values horizontally and vertically adjacent to the SHC peak. This simple algorithm produces sufficiently accurate results (typical error is less than 0.2 pixel), as shown in Table 1. Additional improvements could probably be obtained by computing a three-dimensional least-squares surface fit that best fits the SHC peak and neighboring array values. Once the coefficients of this surface were determined, the coordinates of the maximum point on the surface would correspond to the average subimage velocity.

One problem still remains with the preceding approach. At certain locations in the velocity field, such as stagnation points, the velocity may be close to zero. If, on average, a particle moves less than $^{1}/_{2}$ pixel between exposures, the SHC value will occur at (0, 0), which is also the location of the highest correlation value. Since the algorithm ignores $A(0, 0)$, the search for the SHC will actually find the *third* highest correlation at a random

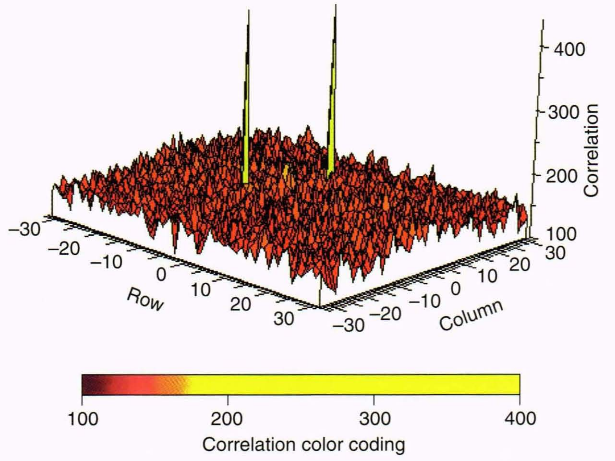


Figure 3. The results of the correlation of a representative 64 × 64 pixel subimage of the velocity flow image. The plot height at a coordinate (x_1, y_1) is proportional to the "goodness" of correlation for a subimage correlated with a copy of itself, offset by x_1 pixels horizontally and y_1 pixels vertically. This offset is proportional to the average particle velocity in the subimage. For clarity, the highest peak, which is always at coordinate (0, 0), is omitted. The input subimage has an "ideal" correlation peak at coordinate (9, 3). The plot is symmetrical; further analysis is necessary to resolve the velocity direction ambiguity.

Table 1. Comparison of simulated and calculated pixel displacements between two exposures. Pixel displacements are proportional to particle velocities.

Subimage location		Simulated movement between exposures (pixels)		Calculated movement between exposures (pixels)		Movement error magnitude (pixels)
Row	Column	Row	Column	Row	Column	
0	0	−0.82	2.32	−0.65	2.36	0.18
128	512	1.34	2.52	1.22	2.51	0.12
768	768	−0.82	2.28	−0.84	2.19	0.09

location in the subimage. This incorrect SHC problem is solved by comparing the average velocity of a subimage with the velocity of the neighboring subimages. If a significant difference is observed, the velocity is replaced by the interpolation of the neighboring subimage velocities. Care must be taken to ensure that the neighboring velocities themselves are correct. Even then, errors are possible, for example, where a velocity field symmetrically increases in magnitude on either side of a null.

The FFT-based correlation process cannot distinguish between particle movement in one direction, say, from A to B, and in the opposite direction, from B to A. This manifests itself as a radially symmetric correlation array, that is, after shuffling, $A(N/2 + r, N/2 + c) = A(N/2 - r, N/2 - c)$. To determine the correct overall velocity direction, the initial (brighter-intensity) location of a randomly chosen particle (r_i, c_i) in a subimage is identified by thresholding. Then, assuming a velocity (from correlation results) with components r_{SHC} and c_{SHC} pixels, the subimage is examined (intensities are added) in the neighborhood of locations $(r_i + r_{SHC}, c_i + c_{SHC})$ and $(r_i - r_{SHC}, c_i - c_{SHC})$. The neighborhood in the correct direction (where the second exposure was taken) should have a higher sum. The random locations of other particles will occasionally give an incorrect result, so the sums are calculated and compared for several particles to reach a consensus.

TEST DATA GENERATION

Setting up the tank hardware to perform velocity flow measurements is time-consuming. Once the tank has been stratified and background motions have been allowed to damp out, it can only be used for a limited set of "runs" because the repeated movements of the cylinder through the tank mix the liquid and destratify it. It is desirable to minimize the number of runs needed just to optimize the experimental parameters such as particle density and exposure times. To assist in determining such parameters before the experiments, as well as to aid the development and testing of the PDV algorithm, we wrote a program (the Particle Displacement Velocimetry Test [PDVT]) that simulates the data that would be recorded by the CCD camera under various test conditions and incorporates the correlation analysis described previously.

The program is based on a simulated velocity field of arbitrary origin. Currently, we are using a velocity field based on a rather complicated, time-dependent analytical stream function. (In fact, the velocity field is complicated enough that we automatically generated the code to calculate it using Wolfram Research's symbolic mathematics program, Mathematica). The velocity field was constructed to have many of the expected features of interest in the towed cylinder experiment, including stagnation points, a boundary layer with strong velocity gradients, and von Karman vortices, in addition to a benignly free-streaming region.

Several user-specified parameters (e.g., field of view, particle density, and size) can be supplied to simulate experimental options. The particle positions are randomly chosen over the whole field of view, consistent with the overall particle density. These physical locations are then mapped into the appropriate image array elements on the basis of the image size. Particles at pixel boundaries will realistically map into multiple image elements. Particle intensities are set to maximum values, modified by simulated noise parameters. To simulate the movement of particles between exposures, the velocity of each particle at its first exposure position is multiplied by the time between exposures to produce a position displacement. This displacement is added to the first position to produce the second exposure position, which is mapped, as before, into image array elements, but at lower intensities. Dark pixel noise is then added to the rest of the image array elements. The image array now simulates the CCD camera data, ready for correlation.

We designed the PDVT program as a development tool so that, in addition to generating the final results (average subimage velocities), it produces various useful intermediate or more detailed results. Intermediate results include a single-exposure image, double-exposure image, and correlation array. Detailed results include statistical analysis of the correlation array and comparisons of actual correlation results with ideal results. The images are stored in either binary format for subsequent processing or the standard TIFF image format for off-line viewing. The correlation results are saved in tabular text form, in binary format, or as a new image type that is the same size as the original image; velocity vectors originate at the center of each subimage, representing the average velocity for the associated subimage. Figure 4a is a velocity vector display for the entire image; the cylinder location is superimposed, and its movement is to the left. Figure 4b, a magnified view of the velocity flow downstream from the cylinder, clearly shows the flow's turbulent nature.

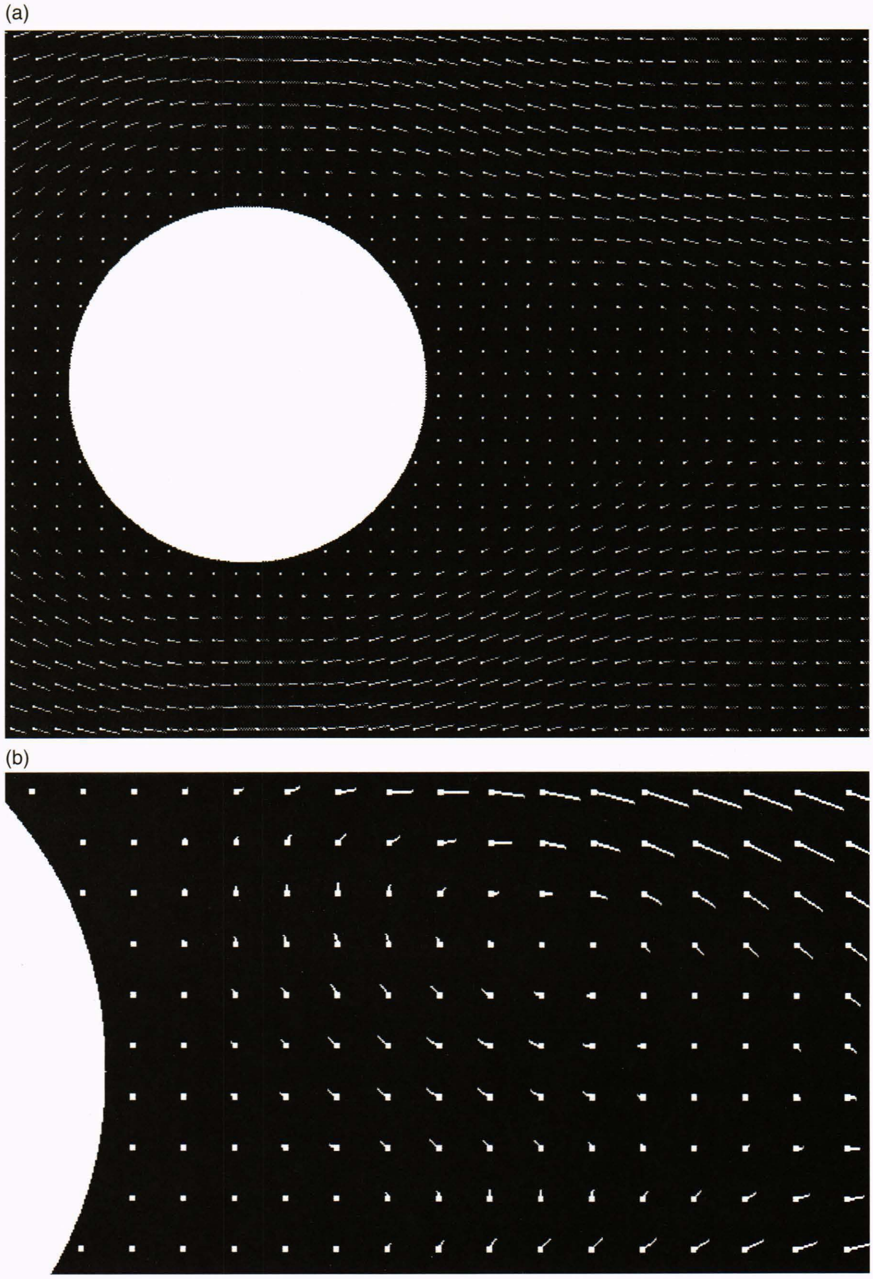


Figure 4. (a) Velocity vector display for entire image. The relative average velocity for particles in each 64 × 64 pixel subimage is shown by a vector whose origin (indicated by dot at one end) is at the center of each subimage. The cylinder position is superimposed. Cylinder movement through the fluid is to the left. (b) A magnified view of (a) with rescaled velocity vectors, showing the turbulent velocity flow to the right of (downstream from) the cylinder.

COMPUTER HARDWARE AND SOFTWARE

The PDVT program needs no special hardware and, with some limitations, can execute on a PC, Macintosh, or Unix workstation. Execution time on a PC or Macintosh for a large image is prohibitive. The program was developed on an IBM RS/6000 workstation in C language.

The computer hardware that performs the experiments consists of a 486 PC with a high-resolution display, connected to the CCD camera. A high-resolution image (1280 × 1024 × 12 bits) is sent from the camera to the PC in approximately 20 s, where it is stored in extended PC memory. These image data are stored in a data file for subsequent analysis or are submitted to correlation analysis.

The program that executes on the PC, hereafter called the PDV program, contains the correlation analysis portion of the PDVT program. In addition, it controls the camera and provides a user interface for defining the test parameters and displaying or archiving intermediate and final test results.

Since the execution time of the correlation analysis on the 486 PC is excessive, we augmented the PC with a commercial coprocessor board that contains two high-speed digital signal processing (DSP) Motorola 96002 chips. These high-speed chips are optimized for floating-point calculations and have special parallel floating-point and data movement instructions for efficiently performing FFT calculations. Table 2 compares the execution times for correlation analysis on the entire image for the coprocessor board, an IBM RS/6000 workstation, and a high-performance Alpha PC. The execution times of the board are based on relative execution times for 2D FFTs,

Table 2. Comparison of execution times of the correlation algorithm on a fixed-size image for two subimage sizes.

	Computers		DSP 96002 board	
Subimage size (pixels)	IBM RS/6000 (min)	Alpha PC (min)	Single chip (min)	Dual chip (min)
32 × 32	4.1	2.5	1.9[a]	0.95[a]
64 × 64	3.6	2.8	2.2[a]	1.1[a]

[a] Implementation of the correlation algorithm for the DSP 96002 board is not complete. Listed values are estimates based on measured execution times of 2D FFTs by the computers and board.

since parts of the correlation algorithm are still being implemented on the board. Times are shown in the table for both the current single-DSP-chip implementation and the future dual-chip version.

Because of its high performance, the newly introduced Alpha PC was considered as a substitute for the 486 PC and coprocessor board. Although the CCD camera interface hardware is compatible with the Alpha PC, the software drivers are not, and they will not be for the foreseeable future. Nevertheless, the Alpha PC remains an attractive possibility for the future.

SCHEME OF OPERATION

Before a test begins, the tank is prepared and the various test parameters are defined with the PDV program. The experimental flow is initiated under computer control, and the camera is turned on, records the doubly exposed image, and sends the image to the PC. After the camera data have been received, values below a threshold are set to zero to remove the dark pixel noise. These data are then partitioned into subimages that are sent individually to the coprocessor board via a direct memory access channel. While the board analyzes one subimage array, another subimage is sent to an alternate buffer on the board. This double buffering scheme keeps the board continuously busy. After a subimage has been analyzed, the correlation result (the average velocity) is sent back to the PC to be displayed or archived. This process is repeated for all subimages. The camera continues to collect images for real-time analysis as long as the experimental flow can be maintained.

CURRENT STATUS

The work on the PDV project has thus far dealt primarily with the development of the PDVT program, which performs correlation analysis on simulated velocity flows. The program is fully operational; however, additional data analysis functions and enhancements to further improve accuracy will probably be included later.

The real-time experimental computer hardware consists of a PC, with a coprocessor board, attached to a high-resolution CCD camera. The transition to this system will involve converting part of the PDVT program to run on the PC, modifying camera interface software, and writing a Windows-based user interface. The correlation analysis portion of the PDVT program will be converted to run on the PC coprocessor board and will be augmented by functions that provide control and data transfer between the PC and the board. In addition, camera interface software will be extracted from stand-alone programs supplied by the camera vendor. Furthermore, Windows-based user interface software will be written to provide control of the experiment, including camera control and display and archiving of experimental results.

REFERENCES

[1] Aubry, N., Holmes, P., Lumley, J. L., and Stone, E., "Application of Dynamical System Theory to Coherent Structures in the Wall Region," *Physica D* **37**(1–3), 1–10 (Jul 1989).

[2] Sommerer, J. C., and Ott, E., "Particles Floating on a Moving Fluid: A Dynamically Comprehensible Physical Fractal," *Science* **259**, 335–339 (15 Jan 1993).

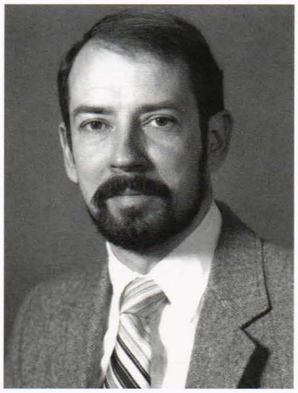
[3] Adrian, R. J., "Particle-Imaging Techniques for Experimental Fluid Mechanics," *Annual Rev. Fluid Mech.* **23**, 261–304 (1991).

[4] Schuster, P. R., and Wagner, J. W., "Holographic Velocimetry for Flow Diagnostics," *Exp. Mech.* **28**(4), 402–408 (Dec 1988).

[5] Willert, C. E., and Gharib, M., "Digital Particle Image Velocimetry," *Exp. Fluids* **10**, 181–193 (1991).

ACKNOWLEDGMENTS: The author wishes to thank Harold E. Gilreath for the opportunity to work on the PDV project, and John C. Sommerer for useful discussions during the development of the system and preparation of this article.

THE AUTHOR



STEVEN D. DIAMOND received a B.S. degree in electrical engineering from the University of Colorado in 1976 and an M.S. in computer science from The Johns Hopkins University in 1986. Before joining APL in 1983, he worked at Ball Aerospace and Storage Technology Corporation in Colorado, and Digidata Corporation in Jessup, Maryland. His first position at APL was in the Advanced Systems Design Group, where he worked on the Cooperative Engagement System. In 1990, Mr. Diamond joined the Mathematics and Information Science Group in the Milton S. Eisenhower Research Center, where he is currently involved in research in real-time neurological signal processing, particle displacement velocimetry, distributed computer systems, and database processing. He is also on the faculty of The Johns Hopkins University G.W.C. Whiting School of Engineering.