

Fluorescent Image Tracking Velocimetry algorithms for quantitative flow analysis in artificial organ devices

Ramanand Singh*, Franklin Shaffer† and Harvey Borovetz‡

ABSTRACT

A Fluorescent Image Tracking Velocimetry (FITV) system for producing two-dimensional velocity maps of flow fields is presented. This system is capable of measurements at flow boundaries, such as the blood-biomaterial interfaces in artificial cardiac organs (in vitro only). Three pulse-coding schemes — a single-pulse code, a dash-dot pulse code, and a constant-frequency pulse code — and associated image analysis algorithms are described. These algorithms were applied to analyze flow in three types of artificial cardiac organs: the Novacor Left Ventricular Assist System, the Nimbus AxiPump, and the Hattler Intravenous Membrane Oxygenator. Results are presented and discussed in terms of image recognition. Despite the drawback of time-direction ambiguity, a constant-frequency pulse with a hybrid of constant-frequency and single-pulse analyses was found to provide superior results for these applications.

1. INTRODUCTION

Fluorescent Image Tracking Velocimetry (FITV) is a multiple-exposure optical imaging technique. A transparent fluid is seeded with small (30 microns for this study) neutrally buoyant fluorescent particles. The particles are excited with an acousto-optically modulated argon-ion laser. Background scattering of argon excitation beam by flow boundaries is filtered so that only fluorescent wavelengths are imaged. This allows particles to be imaged close to light-scattering surfaces. Image acquisition and analysis is performed with a Dage-MTI 81 camera and an ANDROX multiprocessor parallel DSP board housed in a SUN SPARCserver 670MP. A detailed discussion of the FITV technique is presented in Shaffer et al. [1], and example applications are discussed in references [1,2,3,4,5,6]. A schematic of the FITV system developed at the U.S. Department of Energy's (DOE) Pittsburgh Energy Technology Center (PETC) is shown in Figure 1. The laser-pulse timing can be controlled to produce coded-pulse particle-image displacement patterns. The input frame is usually a 1024x1024 pixel gray scale image with 256 gray levels.

The image analysis techniques and their corresponding algorithms are discussed in Section 2. The algorithms have been implemented in a general-purpose image analysis software, Pattern Recognition and Image Processing System (PRIPS), using the C programming language, XView and SunVision toolkits, and a digital signal processing library from the manufacturer of the ANDROX DSP board. The software is driven by experimental parameters, and a user can input these parameters online through an advanced graphical user interface. Various applications of this software are discussed in Section 3. The results from the experiments and a comparative performance analysis of the three algorithms are presented in Section 4. Section 5 presents concluding remarks of the study.

2. CODING SCHEMES & IMAGE ANALYSIS ALGORITHMS

The three pulse-coding schemes mentioned in Section 1 are discussed in this section. A corresponding image analysis algorithm is presented for each of the pulse-coding schemes discussed. Each of the image analysis techniques discussed in this paper operate on binarized frames. A binarized frame can be produced with various processing algorithms, but, for this study, a median filter and fixed threshold were used.

* Science Applications International Corporation, P.O. Box 18288, Pittsburgh, PA 15236, U.S.A.

† U.S. Department of Energy, P.O. Box 10940, Pittsburgh, PA 15236, U.S.A.

‡ University of Pittsburgh, Department of Surgery, Pittsburgh, PA 15260, U.S.A.

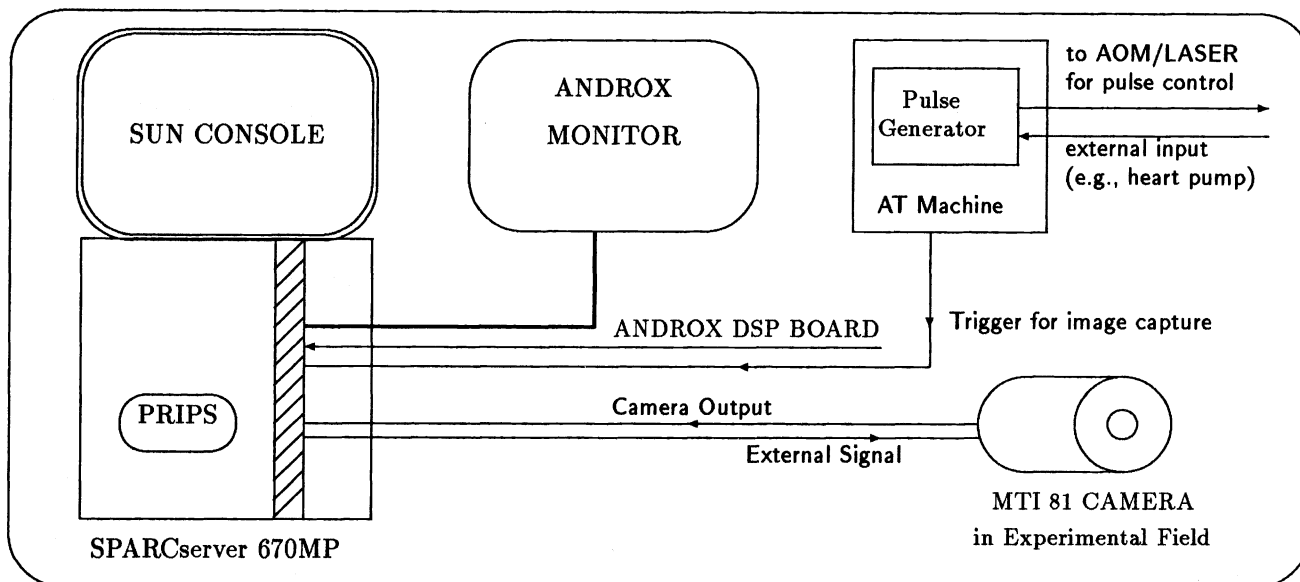


Figure 1: Fluorescent Image Tracking Velocimetry system setup at U.S. DOE's PETC

2.1 Single-Pulse Code

In this simplest pulse-coding scheme, the field of view is illuminated with a single laser pulse of fixed duration. Since the particles move during the pulse, streaks appear in the captured frame. A streak represents the particle motion over the duration of the pulse. The particle trajectory and velocity magnitude are extracted from the coordinates of the streak and the illumination time. However, the time-direction of the motion is unknown. The image analysis for the single-pulse code method is a three-step process: image thinning, line tracking, and velocity computation.

2.1.1 Image Thinning

The image analysis process starts with the transformation of a binarized frame into a set of pixels that still retains significant information about the particle images in the original frame. This transformation can be accomplished by using either an edge-detection filter or a thinning (skeletonization) process. Edge-detection filters are faster than image thinning algorithms and are often used to extract lines or edges in an image [7]. The particle displacement computation in a typical edge-detection method is based on the image boundary length. This computation is likely to produce different results for images of different image diameters. Hence, these filters are not suitable for this study due to considerable variations in the streak diameter and brightness.

The thinning algorithm, on the other hand, reduces the streaks to particle paths and performs for particles of biaxially symmetric (e.g., circle, square, diamond) shapes. Several thinning algorithms have been reported in the literature [8]. Image thinning in this study is accomplished by executing a parallel thinning algorithm [9] on the binary frame. This algorithm was developed by Zhang et al. [9] and was further refined by Lu and Wang [10]. The essence of the algorithm is to iteratively shrink an image boundary by one pixel at a time symmetrically from all directions until the boundary itself vanishes and only a skeleton of unitary thickness of the original image remains. The execution of this algorithm is usually a CPU intensive process that often takes more than 10 seconds per frame, especially for larger particle images. However, under good image conditions, this algorithm produces the desirable result of continuous tracking of the path of the particle centroids over the period of exposure, irrespective of the image size and curvature.

2.1.2 Line Tracking

The line-tracking process attempts to construct lines (particle trajectories) from the output of an image-thinning step. The process can be summarized as: Given an input array, $I(i, j)$, the line tracking process produces shapes $S(n)$, where $i = 1 \dots 1024$, $j = 1 \dots 1024$, and $n = 1 \dots N$; N being the number of shapes in the frame. Each of the shapes in $S(n)$

is an ordered list of pixels such that all pixels in the list are 8-connected neighbors of their adjacent pixels. Two pixels $P_i(x_i, y_i)$ and $P_j(x_j, y_j)$ are said to be 8-connected neighbors if $|x_i - x_j| \leq 1$ and $|y_i - y_j| \leq 1$, x and y being the two coordinates of a pixel.

Shapes are formed by creating lists of all nonzero pixels from the given binarized image data. Starting at a nonzero originator pixel, the process scans its 8-connected neighbors. The nonzero pixels, if found, are added to the aforementioned list and deleted from the original image. The process is repeated with the newly added pixel. The list expansion stops when no nonzero pixels are found. The algorithm is repeated at the originator pixel to address the situations where the originator pixel was in the middle of a shape, creating an alternate list of nonzero pixels. All such lists are then merged. In the event of intersecting trajectories, two criteria were tested. The algorithm selects branches such that (i) either the final list has the maximum length, or (ii) the angle of deviation at the intersection is minimized. The first choice was found to be more efficient than the latter for our application.

2.1.3 Velocity Computation

As a result of a prior field-of-view calibration, scaling factors are established to convert pixel displacements into length units. The mean particle speed along each path is simply determined by the path length divided by the pulse duration. The time-direction of the particle motion, however, is ambiguous with the single-pulse method. To derive the direction of particle motion, a dash-dot coded pulse scheme and corresponding algorithm were developed.

2.2 Dash-Dot Pulse Code

To determine the direction of the particle flow, an asymmetrical pulse pattern can be used. The simplest is the dash-dot coded pulse pattern, as shown in Figure 2. This pulse pattern consists of a long pulse (T_l) followed by a short pulse (T_s). The width of the dark period is represented by T_d . The goal of the dash-dot coded-pulse method is to associate the dots with the dashes and to determine the *magnitude* and *time-direction* of the velocity from these images. This is a three-step process: shape classification, dot location, and dash-dot association.

2.2.1 Shape Classification

The shapes in the frame are extracted using the thinning algorithm (Section 2.1.1) and the line-tracking algorithm (Section 2.1.2). To classify the shapes as dashes and dots, two assumptions are made concerning the dynamic range of velocities and the resulting pulse patterns. The dynamic range of velocity is defined as a ratio of the largest to the smallest velocities in the flow field. The two assumptions are the following:

- the particle must move at least N_l particle diameters during the longer pulse period, T_l (where, N_l times the particle diameter equals the minimum velocity times T_l), and
- the particle must move a maximum of N_s particle diameters during the shorter pulse period, T_s (where, N_s times the particle diameter equals the maximum velocity times T_s).

The pulse durations, T_l and T_s , are adjusted to meet these assumptions. Shapes of lengths less than N_s particle diameters are classified as dots and shapes of lengths more than N_l particle diameters are classified as dashes.

2.2.2 Dot Location

Once all the streaks are identified, two probable locations of dots associated with every streak are computed. The locations of these dots are estimated by extrapolating a streak curve in both directions. The process of dot location computation consists of three steps: curve fitting, arc-length determination, and extrapolation.

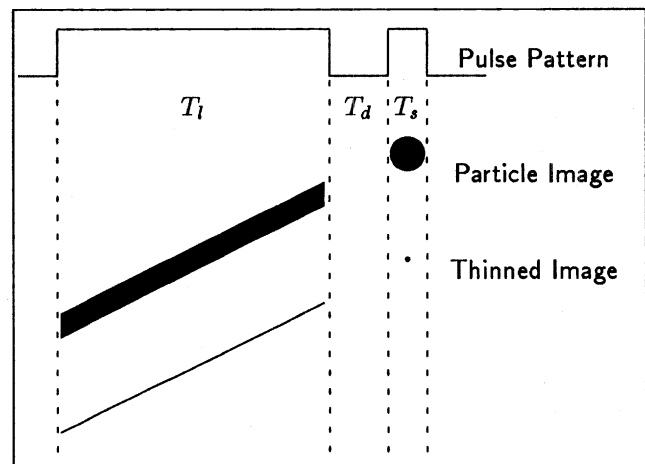


Figure 2: Pulse Pattern and Resulting Dash and Dot

2.2.2.1 Curve Fitting: The curve-fitting process finds a function $g(x, y)$ through a set of given pixel coordinates $P(x_i, y_i)$. For this application, the curve must be in the form $y = f(x)$ or $x = f(y)$ to facilitate the computation. Such a function is $y = \Psi_a(x) = \sum_{j=0}^m a_j \phi_j(x)$, where m is the number of terms in the function expansion, a_j are the coefficients of expansion, and ϕ_j are the fixed functions of x , called the basis functions.

This curve fit uses singular value decomposition of the matrices arising from minimum error conditions. The coefficients of expansion a_j are used to calculate the error in the curve fit with the expression $E_x^2 = \sum_i (y_i - \Psi_a(x_i))^2$, where the summation is taken over the entire set of data points. This mode of fitting y versus x is called the X-Mode. The above process is repeated for the Y-Mode fit by exchanging the variables x and y and replacing a_j with b_j . The error in the Y-Mode fit, E_y , is then estimated. The mode producing the better fit is then selected, based on the minimum of the two estimated errors, E_x and E_y . This algorithm is described in detail in **Numerical Recipes in C** by Press et al. [11].

The choice of X-Mode and Y-Mode fitting is essential to address curvature in both x and y directions. This algorithm automatically switches the curve-fitting mode based on curvature. The curve-fitting algorithm accepts a list of pixel coordinate locations belonging to a single shape and produces the mode of the fit and the coefficients of the curve fit applicable for that particular shape.

2.2.2.2 Arc-Length Determination: Since an analytical curve passing through the set of data points is derived from the previous step, the arc length can be easily computed through numerical integration. A ten-point Gauss-Legendre quadrature was used to perform the above numerical integration of the fitted function to compute the streak length [11].

2.2.2.3 Extrapolation: The two possible dot locations on either side of a streak are estimated by extrapolation of the streak curve fit. The distance of the extrapolation, d , (i.e., the distance a particle would be expected to travel before a dot is formed) is determined by $d = \bar{v} \times (T_d + T_s/2)$, where $\bar{v}$ is the mean particle speed defined as $\bar{v} = L_l/T_l$, L_l being the streak length.

The probable dot coordinates can be calculated from the intersection of the analytical curve and a circle of radius d centered at the end point of the streak. A combination of Newton-Raphson and bisection methods is used to solve the corresponding system of nonlinear equations for x and y . The hybrid algorithm takes a bisection step whenever Newton-Raphson would take the solution out of bounds, or whenever Newton-Raphson is not reducing the size of the brackets rapidly enough [11]. The roots x and y are refined iteratively until they are within a specified accuracy limit.

2.2.3 Dash-Dot Association

For each streak, a list of dot coordinates is scanned and the nearest dot to the predicted location is assigned to the streak, subject to a user-specified tolerance circle. To account for dynamic variations, this tolerance is specified as a percentage of the streak length. The assigned dot is made unavailable for any further assignments. If no acceptable dot coordinates are found within the specified tolerance, the streak is discarded. If, on the other hand, more than one dot is found within the specified tolerance circle, the one closest to the predicted location is selected.

2.3 Constant-Frequency Pulse Code

In this scheme, a pulse train of constant frequency produces several images of each particle along its path. The centroids of these images are used to construct the particle trajectory and the velocity along the trajectory is determined from the particle image displacements. This process is graphically illustrated in Figure 3 and is described in the following subsections. The image analysis technique presented here is an enhancement of the technique described in [12].

First, the binarized frame is processed to extract blobs (contiguous regions). This algorithm has constant-time complexity, irrespective of the size of the blobs in the image, and is discussed by Ballard and Brown [13]. This process produces a list of centroid coordinates of blobs sorted from top to bottom and from left to right. This ordering helps in the later search process by arranging the centroid coordinates into trajectories, thereby corresponding to the general

sense of particle motion in the flow field (except in case of Novacor LVAS). The following assumptions were made to identify the valid trajectories:

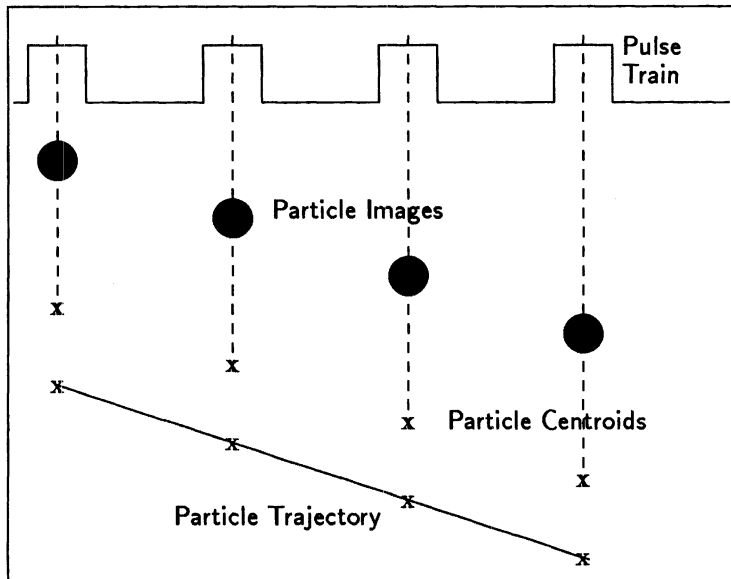


Figure 3: Constant Frequency Pulse Pattern & Particle Images

- The number of particle centroids along a valid trajectory, N , is in the range $N_{min} < N < N_{max}$. N_{max} is known from the number of pulses delivered during the frame. N_{min} (a user-specified parameter value) is less than N_{max} allowing for missing images along a trajectory. Missing images may be caused by particles passing in and out of the laser sheet or by dim particle images.
- The successive images of a particle are not separated by more than a specified number of pixels. The maximum separation can be determined if the maximum velocities in the flow field can be estimated.
- Particle accelerations are such that the distance between a pair of consecutive centroids of a particle does not change by more than a specified fixed percentage of the distance between the previous centroid pair.

The algorithm is designed to handle situations such as intersecting trajectories and trajectories with missing images. Particle images may be missing due to excessive variations in brightness levels in the original frame. The trajectory formation is a three-step process: trajectory inception, trajectory expansion, and trajectory acceptance.

2.3.1 Trajectory Inception

The process of trajectory formation starts with a search for an eligible, initial particle centroid. This initial centroid is referred to as a progenitor and is assumed to be the first image of the particle along the particle trajectory. Once the progenitor is chosen, the list of particle centroids is searched for a secondary initiator within the initial search area. The initial search area is defined as a circular area, surrounding the progenitor, with a radius equal to the distance the particle would travel at the maximum expected velocity during a single pulse period. In case of multiple secondary initiators, one is selected for immediate processing and the remainder are marked as alternate secondary initiators. On completion of the search process with one secondary initiator, the next secondary initiator is selected from the list. The process is repeated until all the alternate secondary centroids are tested. Conversely, if no secondary initiator is found, the corresponding progenitor is marked ineligible to be on any trajectory and the whole trajectory inception process repeats with a search for a new progenitor. When all the available centroids have been examined for their eligibility as progenitors, the trajectory formation process is terminated.

2.3.2 Trajectory Expansion

The progenitor and the secondary initiator are considered two probable consecutive images of a particle. These two centroids will be denoted as C_0 and C_1 respectively. Now, the location of the next potential centroid is predicted, based on a linear extrapolation scheme, and the list of centroid coordinates is searched for a centroid within the descendant search area. The descendant search area is a circular region centered around the predicted centroid location. Its radius is a user-defined parameter based on the magnitudes of particle acceleration (or deceleration) and the trajectory curvature. If a centroid is found within the descendant search area, it is included in the search tree and denoted as C_2 . Now, C_1 and C_2 are assigned as two initiators for the next round of searches. This round will try to find C_3 and the process repeats for the next centroid until the frame boundary is reached, the number of centroids in the trajectory exceeds its maximum value (N_{max}), or no centroid is found within the descendant search area.

If no centroid was found within the descendant search area, an imaginary centroid can be placed at the predicted location and the search process continues. The process of assuming imaginary centroids continues until a real centroid is found, or the number of successive imaginary centroids exceeds its maximum allowable value. The number of successive imaginary centroids allowed is defined by the user based on image quality. In cases where real centroids are found, the imaginary centroids are termed missing and are inducted into the trajectory; otherwise, all the imaginary centroids are discarded and the search path is terminated at the last real centroid. The case of multiple descendant centroids is dealt with in a fashion similar to the multiple secondary initiators. Figure 4 shows an example of particle images for a constant-frequency pulse coding scheme. An example search tree constructed from such a scenario (Figure 4) is shown in Figure 5. The final trajectories are highlighted.

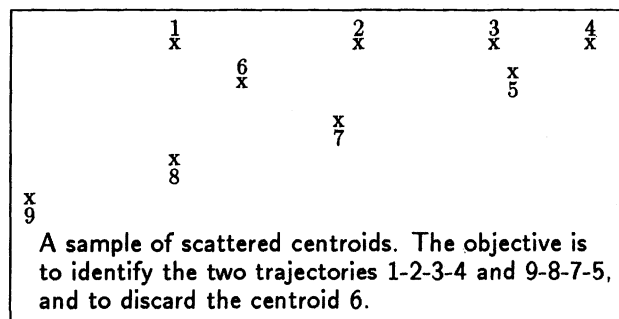


Figure 4: Example Particle Image Data

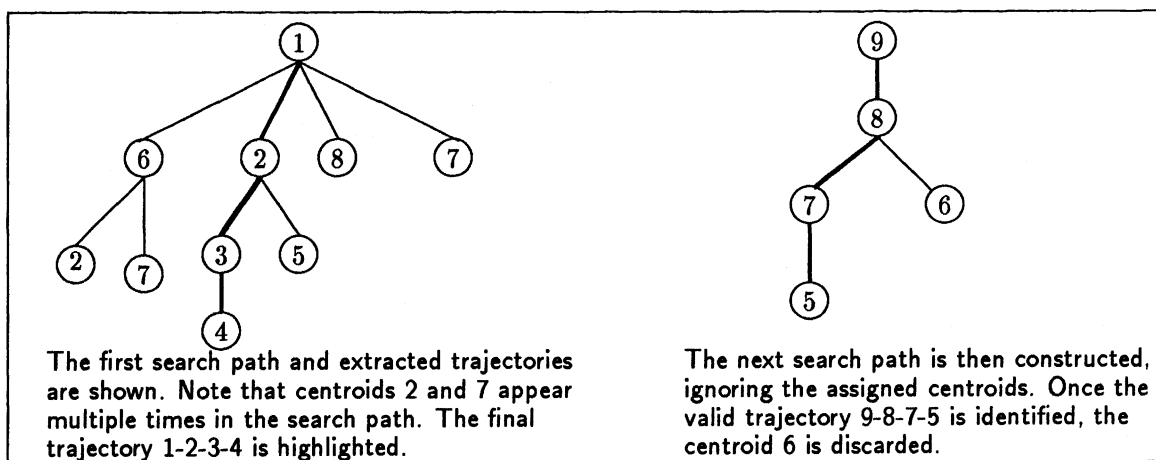


Figure 5: Search Tree from Particle Centroids

2.3.3 Trajectory Acceptance

One of the paths of the search tree constructed in the previous step is then selected as the final trajectory. In case of multiple eligible trajectories, the one with the most centroids is selected as the final particle trajectory. The centroids of the selected trajectory are termed "consumed" and are no longer available for further consideration. The other centroids in the search path (but not in the final trajectory) are eligible for assignment to other trajectories. If no real trajectories are found in the search path, all the centroids are marked eligible for assignment in future trajectory searches. However, the progenitor centroid of this search path must be marked ineligible for consideration as a progenitor again. This trajectory formation process is repeated until all centroids have been examined as progenitors or are identified in real trajectories.

In flow fields with a large dynamic range of velocities, using variable (incremental) initial search areas is beneficial. The initial search area is set to a value corresponding to a low velocity and the trajectory formation process is performed. This process removes the centroids of the valid trajectories from the image data. Next, the initial search area is incremented by a specified amount, and the trajectory formation is repeated until the initial search area reaches a value corresponding to the maximum expected velocity in the flow field.

3. APPLICATIONS

The pulse-coding schemes and their corresponding image analysis algorithms are discussed in Section 2. General purpose image analysis software called PRIPS was designed and developed at PETC to implement the three algorithms.

Since PRIPS incorporates algorithms for all three pulse-coding schemes, an individual technique or a combination of techniques was tested on different applications. The software is used to analyze simulated blood flow patterns in three artificial cardiac organs under investigation at the University of Pittsburgh Medical Center. The organs are the Novacor Left Ventricular Assist System (NLVAS), the Nimbus AxiPump, and the Hattler Intravenous Membrane Oxygenator (IMO).

3.1 Novacor Left Ventricular Assist System

The NLVAS is an artificial heart assist device (a blood flow pump) manufactured by Baxter Novacor (Oakland, CA), and is currently employed in clinical trials. The motivation for this study was to optimize the flow characteristics of the NLVAS for a given set of patient hemodynamic conditions [4]. The FITV technique was preferred because of its capability to measure flow at blood-biomaterial interfaces.

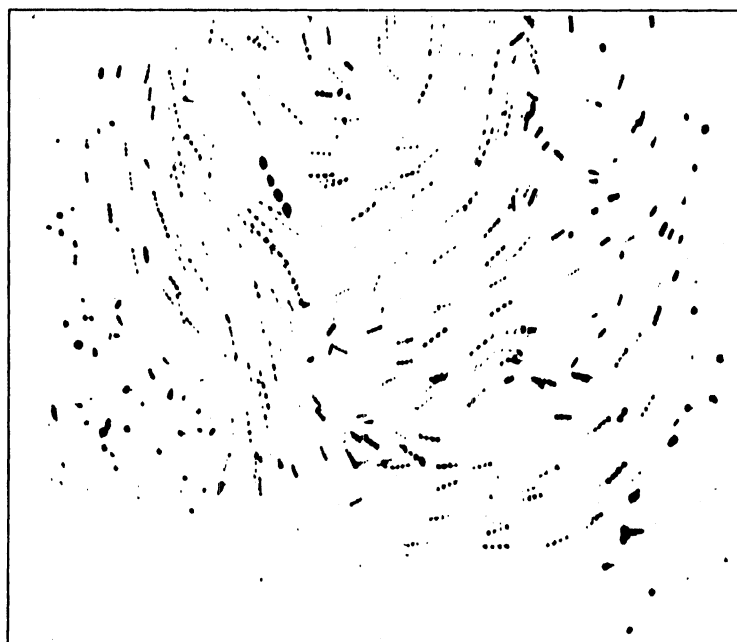


Figure 6. Example FITV image for the Novacor LVAS

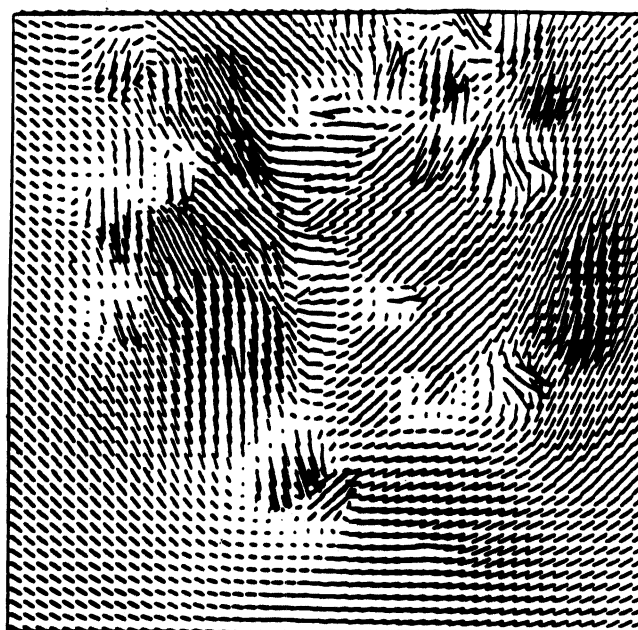


Figure 7. A velocity profile for image shown in Figure 6

The anatomic configuration of the NLVAS and the experimental setup at the PETC Flow Analysis Laboratory have been reported by Shaffer et al. [1] and Woodard et al. [4]. The experimental flow conditions simulated the full range of clinical conditions observed for cardiac transplant patients at the University of Pittsburgh Medical Center [1]. All measurements were performed in vitro using a transparent glycerol-water mixture with a viscosity close to that of blood. Six frames were acquired for each cardiac cycle with four constant-frequency pulses during each frame.

Figure 6 shows an example of a typical FITV image taken near mid-diastole when the pericardial valve of the NLVAS is fully opened and stationary. The range of velocities varies considerably throughout the cardiac cycle. Thus, to maintain reasonable spacings between consecutive particle images, the laser-pulse frequency was varied from frame-to-frame throughout the cardiac cycle. Even with the varying pulse frequency, the velocity near the valve surfaces was found to approach stagnation. This caused an overlapping of particle images resulting in a streak rather than direct particle images. Such images could be analyzed by using a hybrid of single-pulse code and constant-frequency pulse-code analyses with limited manual verification.

An example FITV data extraction, from the image of Figure 6, is shown in Figure 7. This represents a velocity profile of the example image. Flow vectors derived with constant-frequency pulses do not indicate time-direction of the motion. This ambiguity is due to the symmetric pulse coding. For this application velocity magnitude was the main parameter of interest. The flow direction can be determined if an asymmetric pulse coding, such as a dash-dot coded-pulse, is used in the illumination source.

3.2 Nimbus AxiPump

The constant frequency pulse technique was also used to study flow patterns inside an axial flow blood pump, the Nimbus AxiPump. The pump, under joint development by Nimbus, Inc. and the University of Pittsburgh, is also a Left Ventricular Assist Device [2]. Refinement of the flow geometry within the pump rotor and stator sections, along with that of the inflow cannula, is the main motivation of this study.

In the experimental setup, the axial flow blood pump was placed in a flow loop containing normal saline and operated continuously at specific speeds (8,000, 10,000, and 12,000 rpm) and pressure. A detailed physical configuration of the axial flow blood pump, its anatomic placement, and description of the experimental setup is described by Butler et al. [2]. The PRIPS software is used to track the motion of neutrally buoyant particles within the flow field and to provide frame-by-frame particle velocities. Summation of such frames produces a velocity profile of the overall flow field.

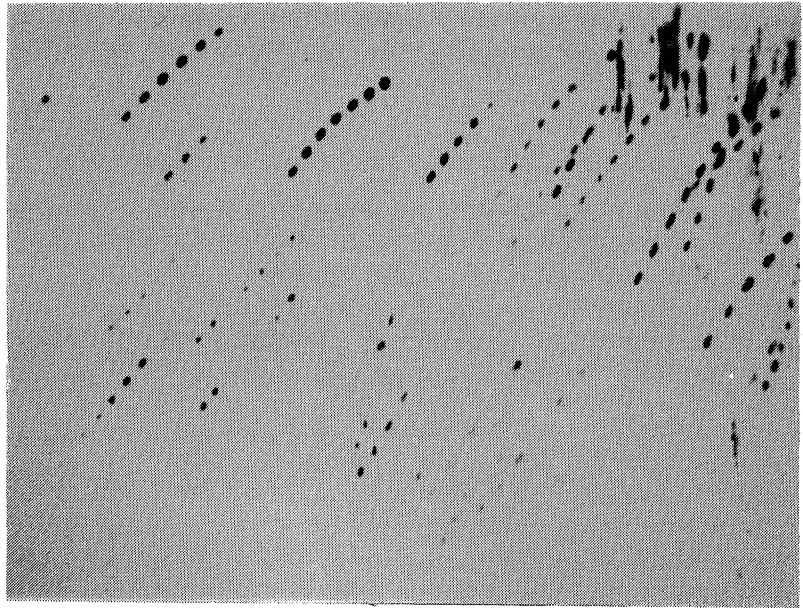


Figure 8: Original FITV Image for AxiPump

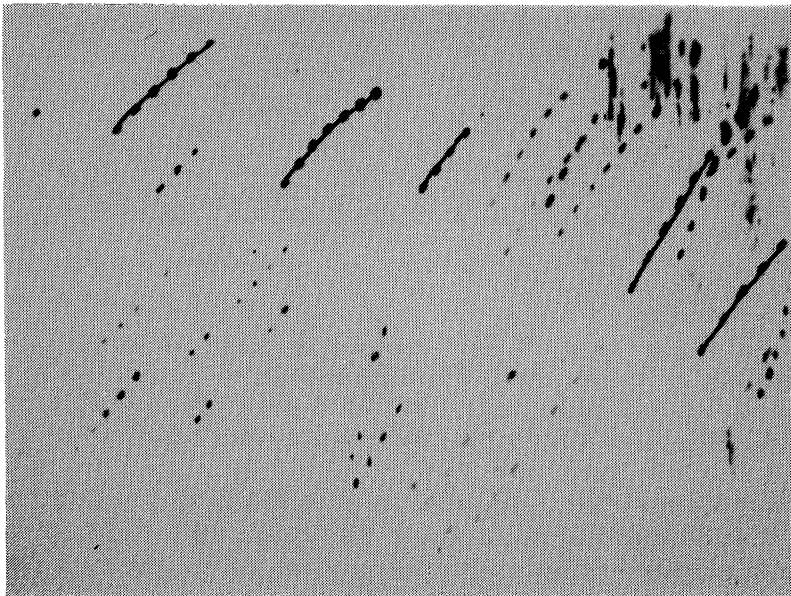


Figure 9: Analyzed Image of Figure 8

An example of a raw image captured during the experimentation is shown in Figure 8. The image data shows the particle images at consecutive intervals of time. The automatic analysis portion of the PRIPS software was used to identify particle trajectories employing the constant frequency pulse-code method. Figure 9 shows the result of the automatic analysis by connecting particle image centroids of all valid trajectories. Since the dynamic range of velocities in the AxiPump is small, particle images did not overlap as with the Novacor LVAS. As evident from the analyzed image data of Figure 9, the minimum number of centroids to form a trajectory was set to 4. Figure 10 shows a velocity profile of an FITV image data of the AxiPump rotating at 10,000 rpm.

A large number of images were captured and analyzed using the image analysis techniques discussed earlier. The flow velocities and resulting shear stresses in the AxiPump were measured.

The blade angles in the pump were properly oriented, but the pump's inducer section had high velocity impact areas.

3.3 Intravenous Membrane Oxygenator

The Intravenous Membrane Oxygenator (IMO) is placed in the vena cava and functions as an artificial lung by diffusing oxygen into the blood stream. It consists of a few hundred hollow fiber membranes arranged around a centrally positioned pulsating balloon. The development of the IMO device is motivated by a desire to find a possible cure for acute respiratory distress syndrome. The central balloon is inflated and deflated (synchronized with the cardiac system)

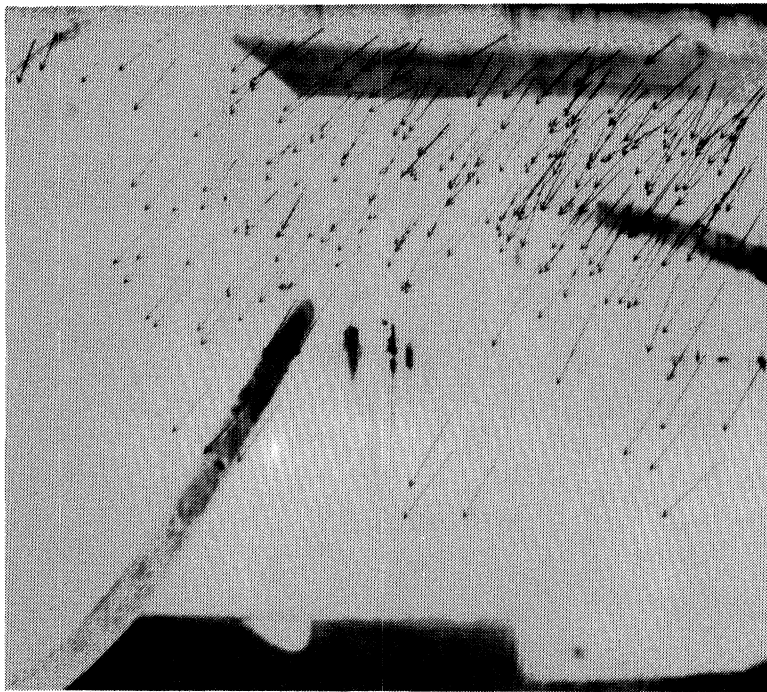


Figure 10: Velocity profile of the image shown in Figure 8

tion the velocity vectors are not parallel to the fibers. To better visualize the convective mixing effects, velocity vectors in these two figures are overlaid from numerous FITV images.

to enhance convective mixing and thus diffusion. The development of the device is still in its experimental stage. A detailed description of this device and the experimental setup can be found in Hattler's recent publication [3].

With this device, time-direction of the flow must be determined to better characterize convective mixing and flow reversal. This requires the use of the dash-dot pulse pattern. A number of image data were captured for different experimental conditions. Two such image data are shown in Figures 11 and 12 representing IMO fibers in stable configuration and in motion, respectively. Mapping flow fields around the IMO has shown that the velocity profiles adjacent to the surface of the fibers have been altered. Figure 13 shows the characteristic parallel streamlines in the flow field when the IMO is in its static configuration with no balloon pulsation (i.e., stationary fibers). As seen in Figure 13, the velocity vectors are parallel to the fibers. But, an actuation of the IMO balloon increases the dynamic range of the velocity and produces a strong mixing at the fluid fiber interface. As shown in

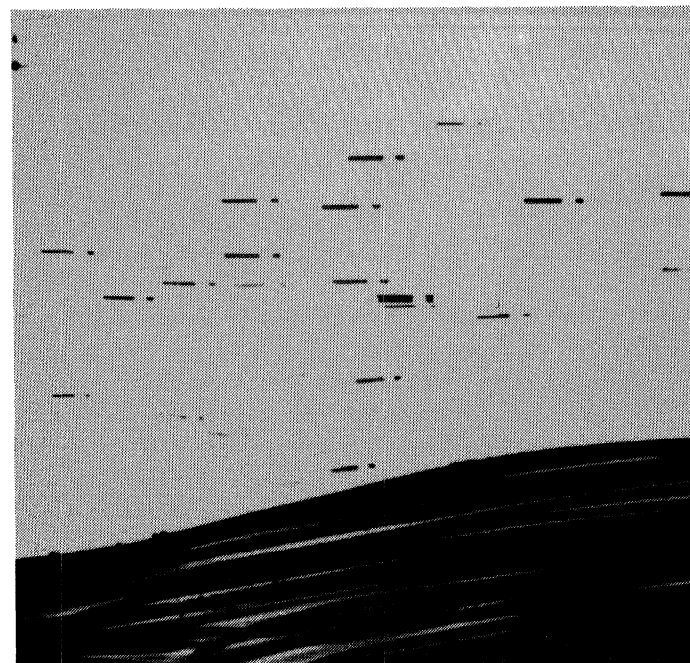


Figure 11. IMO image data for stable configuration



Figure 12. IMO image data 30 ms after start of inflation

An analysis of the captured images shows an enhanced convective mixing around the fibers as well as near the vessel wall. This implies that, with rhythmic balloon inflation/deflation, a greater number of red blood cells in the flowing

blood will be exposed to the oxygen-rich atmosphere near the fiber for a longer period of time.

4. PERFORMANCE ANALYSIS

One of the difficulties experienced during image analysis in the aforementioned applications was excessive variation in the particle image gray level. This may be caused by several experimental conditions including low-level fluorescence of flow boundaries, variations in the amount of dye in particles, and image decay on the camera sensor. The gray level variation can cause image breakups, making image analysis more difficult. However, the image breakup can be minimized in two different ways. A more robust localized variable thresholding technique [14,15] could be adopted but is very computation-intensive. An alternative is to maintain a smaller streak area either by reducing the pulse width of the illumination source or by zooming out of the flow field so that even the maximum particle velocities produce smaller streaks. But this introduces errors in the analysis for slower moving particles. An acceptable data extraction rate in this study was achieved with a single-threshold method.

The single-pulse analysis has a better success rate of data extraction with uniformly high-intensity images of streaks. Such an image usually results from a slower particle. On the other hand, the fast-moving particles tend to produce nonuniform intensities along the streaks, causing image breakups. The image breakup generates several isolated image streaks that could be misinterpreted by the single-pulse method as images of slow-moving particles. This is likely to introduce errors in the analysis. Thus, the single-pulse method is not suitable for image analysis for particles moving at higher velocities.

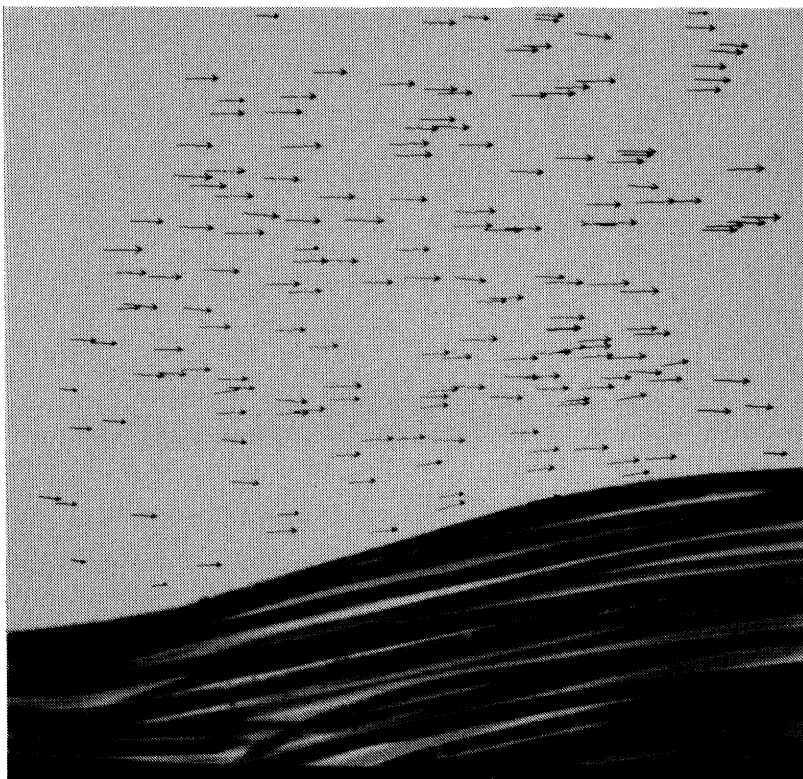


Figure 13: Velocity vectors for stationary fibers

The dash-dot pulse method reduces errors due to image breakup problems by associating a dash with a corresponding dot. This improves the data extraction over the single-pulse method. The dash-dot method discards the broken images if a corresponding dot can not be located within a specified tolerance region. However, it may identify erroneous trajectories resulting from fragmented streaks if high tolerance levels need to be maintained due to acceleration in the flow field. Setting N_t to a higher value prevents the shorter streaks (due to image breakup) from being considered for processing. This minimizes the effect of image breakup, but it has the undesirable effect of biasing the velocity field. Khalighli and Lee's dot-dash-dot-dot pulse method [7] better compensates for this problem, but a combination of high velocities, severe streak curvature, and flow field quality causes this technique to be less effective.

The constant-frequency pulse method generates particle images of significantly lower areas than streaks, minimizing the probability of gray level variation. It also has a better image analysis algorithm with the multiple-path searching technique. This helps discard the fragmented images, resulting in better data extraction, and thus the image breakup problem has significantly less effect on the constant-frequency pulse method. However, at lower velocities the individual particle images tend to overlap. Such merged images are typically discarded by this method and, as a result, bias the data toward higher velocities. However, the single-pulse method can be used to thin down the merged images and to extract the particle path.

Thus, a hybrid of constant-frequency and single-pulse methods was found to be the most effective in this study. The only shortfall with such a hybrid technique is its inability to identify merged particle images. Currently, a user verification is needed to identify such images for data extraction.

5. CONCLUSION

Three pulse-coding schemes and corresponding image-analysis algorithms were implemented in a general purpose image analysis software, PRIPS. This software was used to analyze simulated blood flow patterns in three artificial cardiac devices by applying different image analysis techniques. Representative results are presented to demonstrate the usefulness of the three techniques in handling real-world flow conditions.

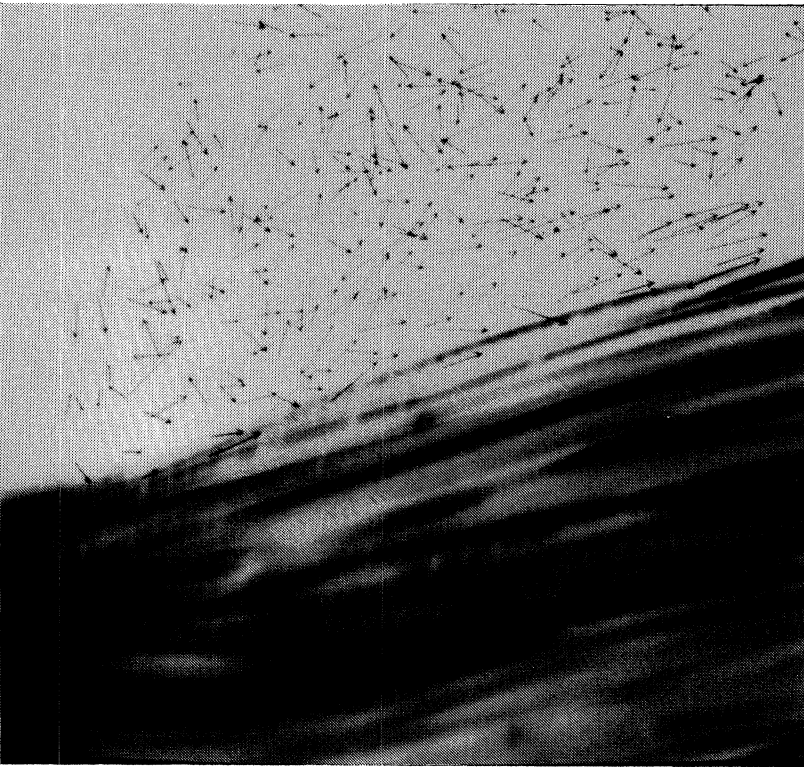


Figure 14: Velocity vectors 30 ms after start of inflation

The single-pulse method achieves a higher data extraction rate, although inconsistent results are obtained due to image breakup of longer streaks. The dash-dot method reduces the image breakup problem but yields a lower data extraction rate. It also produces time-direction data. The constant-frequency pulse method models a system fairly closely and provides the most reliable data extraction. However, both single-pulse and constant-frequency pulse methods fail, individually, to represent lower velocities. A combination of the two methods produces better results. The choice of an image analysis technique depends on the flow condition and on whether the time-direction data is needed.

The manual identification of the merged particle images in the hybrid technique requires a greater level of user interaction with the software. Development of a more

intelligent algorithm to automate this process will be considered for future work.

References

- [1] F. Shaffer, M. Mathur, J. Woodard, H. Borovetz, R. Schaub, J. Antaki, R. Kormos, B. Griffith, R. Srinivasan, R. Singh, and C. McCreary, "Fluorescent Image Tracking Velocimetry Applied to the NOVACOR Left Ventricular Assist Device," *Proceedings, Second National Fluid Dynamics Congress: Cavitation and Multiphase Flow Forum, 1992*, O. Furuga (ed.); pp. 141-147, ASME, New York, 1992.
- [2] K.C. Butler, T.R. Maher, H.S. Borovetz, R.L. Kormos, J.L. Antaki, M. Kameneva, B.P. Griffith, T. Zerbe, and F.D. Shaffer, "Development of Axial Flow Blood Pump LVAS," *ASAIO Journal*, vol. 38, pp. M296-300, 1992.
- [3] B.G. Hattler, P.C. Johnson, P.J. Sawzik, F.D. Shaffer, M. Klain, L.W. Lund, G.D. Reeder, F.R. Walters, J.S. Goode, and H.S. Borovetz, "Respiratory Dialysis: A New Concept in Pulmonary Support," *ASAIO Journal*, vol. 38, pp. M322-325, 1992.

- [4] J.C. Woodard, F.D. Shaffer, R.D. Schaub, L.D. Lund, and H.S. Borovetz, "Optimal Management of a Ventricular Assist System Contribution of Flow Visualization Studies," *ASAIO Journal*, vol. 38, pp. M216-219, 1992.
- [5] F.D. Shaffer and E.R. Ramer, "Pulsed Laser Imaging of Particle-Wall Collisions," Proceedings, International Conf. on Mechanics of Two-Phase Flows, Taipei, Taiwan, June 1989, S.L. Lee and Franz Durst (eds.), pp. 116-128, Institute of Applied Mechanics, National Taiwan University, Taipei (Taiwan), 1989.
- [6] F.D. Shaffer, J.M. Ekmann, and E.R. Ramer, "Development of Pulsed Laser Velocimetry Systems with Photoelectric Image Sensors," Proc., First National Fluid Dynamic Congress Cincinnati, Ohio (USA), July 1988, pp. 1944-1952, AIAA/ASME/ASCE/SIAM/APS, 1988.
- [7] B. Khalighi and Y.H. Lee, "Particle Tracking Velocimetry: An Automatic Image Processing Algorithm," *Applied Optics*, vol. 28, no. 20, pp. 4328-4332, October 1989.
- [8] N.J. Naccache and R. Singhal, "STPA: A proposed algorithm for thinning binary patterns," *IEEE Trans. on System, Man, and Cybernetics*, vol. 14, pp. 409-418, 1984.
- [9] T.Y. Zhang and C.Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236-239, March 1984.
- [10] H.E. Lu and P.S.P. Wang, "A Comment on the Fast Parallel Algorithm for Thinning Digital Patterns," *Communications of the ACM*, vol. 29, no. 3, pp. 239-242, March 1986.
- [11] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, Numerical Recipes in C, Cambridge University Press, New York, 1990.
- [12] E.R. Ramer and F.D. Shaffer, "Automated analysis of multiple pulse particle image velocimetry data," *J. Applied Optics*, vol. 31, no. 6, pp. 779-784, February 1992.
- [13] D. H. Ballard and C. M. Brown, Computer Vision, Prentice Hall, Englewood, New Jersey, 1990.
- [14] T.P.K. Chang, A.T. Watson, and G.B. Tatterson, "Image Processing of Tracer Particle Motions as Applied to Mixing and Turbulent Flow - I: The Technique," *Chemical Engineering Science*, vol. 40, no. 2, pp. 269-275, 1985.
- [15] A. Majumdar, P. Stroeve, J.L. Fales, V.R. Algazi, and J.E. Stewart, "Teaching Computers to Track Particles II. Auto Tracking Algorithm and Coordinate Transformation," *Journal of Imaging Science*, vol. 31, No. 5, pp. 208-219, Sept.-Oct. 1987.
- [16] A.A. Adamczyk and L. Rimai, "2-Dimensional Particle Tracking Velocimetry (PTV): Technique and image processing algorithms," *Experiments in Fluids*, vol. 6, pp. 373-380, 1988.