NASA-CR-204634

Report No. TP-95-101

Report No. LTCF-97-102

Instantaneous Structure of Turbulent Separated Boundary Layers

RONALD J. ADRIAN

Department of Theoretical and Applied Mechanics
University of Illinois at Urbana-Champaign
216 Talbot Laboratory
104 South Wright Street
Urbana, IL 61801

Report of Research Performed During the Period
August 3, 1995 to August 20, 1996

with support of

Grant Number NAG 1 1530
National Aeronautics and Space Administration
Langley Research Center

June 9, 1997

Instantaneous Structure of Turbulent Boundary Layers with Stereoscopic Particle Image Velocimetry

The research proposed for the third year of project was to study the structure of turbulence in boundary layers on aerodynamic surfaces and to develop stereoscopic PIV methods for use in such studies (c.f. proposal dated 9/18/95)

This report describes the final results of the project, which are reported in the reference in Appendix A. This reference contains the technical details, and will constitute the main body of the report. The paper in Appendix A was first presented at the 8th International Symposium on Applications of Laser Techniques to Fluid Mechanics, July, 1996. This paper was one of 30 papers selected from more than 200 papers presented at the symposium for publication in a book by Springer -Verlag. It will become available during 1997.

A stereoscopic PIV system was constructed using two Kodak Megaplus 4.2 CCD cameras to provide the maximum resolution available with CCD's. The cameras are separated by a 30 degree angle and the separation is purely lateral. The field of view is approximately 80mm x 80mm at a working distance of 1600mm. The air flow is seeded with 1 micron oil droplets and illuminated by synchronized YAG laser with ~100 mJ per pulse. The analysis of the CCD images is performed by software from TSI, Inc called *Insight*. This software was specially modified to provide acquisition of the data from the two CCD cameras simultaneously. The cameras were registered physically to within several pixels in the image plane, then registered electronically to correct for residual offset. Once vector displacement fields were acquired, they were combined using software developed in-house. The source code for this software is given in Appendix B.

The flow investigation was performed using the stereo particle image velocimeter to provide views of the instantaneous , two-component velocity fields in streamwise-wall normal planes of a boundary layer in a wind tunnel. The results confirm the formation of near-wall shear layers. a major new insight into the structure came in the form of strong out-of-plane motion along the inclined layers, indicating strong rotation, even of the larger scales.

APPENDIX A

Structure of a Turbulent Boundary Layer Using a Stereoscopic, Large Format Video-PIV

BY

Z. C. Liu[1], R. J. Adrian[1], C. D. Meinhart[2] and W. Lai[3]

Paper first presented at the 8th Int'l symposium on Applications of Laser Techniques to Fluid Mechanics, July, 1997, Lisbon Portugal

Report No. TP-95-101

Report No. LTCF-97-102

Instantaneous Structure of Turbulent Separated Boundary Layers

RONALD J. ADRIAN

Department of Theoretical and Applied Mechanics
University of Illinois at Urbana-Champaign
216 Talbot Laboratory
104 South Wright Street
Urbana, IL 61801

Report of Research Performed During the Period
August 3, 1995 to August 20, 1996

with support of

Grant Number NAG 1 1530
National Aeronautics and Space Administration
Langley Research Center

June 9, 1997

Instantaneous Structure of Turbulent Boundary Layers with Stereoscopic Particle Image Velocimetry

The research proposed for the third year of project was to study the structure of turbulence in boundary layers on aerodynamic surfaces and to develop stereoscopic PIV methods for use in such studies (c.f. proposal dated 9/18/95)

This report describes the final results of the project, which are reported in the reference in Appendix A. This reference contains the technical details, and will constitute the main body of the report. The paper in Appendix A was first presented at the 8th International Symposium on Applications of Laser Techniques to Fluid Mechanics, July, 1996. This paper was one of 30 papers selected from more than 200 papers presented at the symposium for publication in a book by Springer -Verlag. It will become available during 1997.

A stereoscopic PIV system was constructed using two Kodak Megaplus 4.2 CCD cameras to provide the maximum resolution available with CCD's. The cameras are separated by a 30 degree angle and the separation is purely lateral. The field of view is approximately 80mm x 80mm at a working distance of 1600mm. The air flow is seeded with 1 micron oil droplets and illuminated by synchronized YAG laser with ~100 mJ per pulse. The analysis of the CCD images is performed by software from TSI, Inc called *Insight.* This software was specially modified to provide acquisition of the data from the two CCD cameras simultaneously. The cameras were registered physically to within several pixels in the image plane, then registered electronically to correct for residual offset. Once vector displacement fields were acquired, they were combined using software developed in-house. The source code for this software is given in Appendix B.

The flow investigation was performed using the stereo particle image velocimeter to provide views of the instantaneous , two-component velocity fields in streamwise-wall normal planes of a boundary layer in a wind tunnel. The results confirm the formation of near-wall shear layers. a major new insight into the structure came in the form of strong out-of-plane motion along the inclined layers, indicating strong rotation, even of the larger scales.

APPENDIX A

Structure of a Turbulent Boundary Layer Using a Stereoscopic, Large Format Video-PIV

BY

Z. C. Liu[1], R. J. Adrian[1], C. D. Meinhart[2] and W. Lai[3]

Paper first presented at the 8th Int'l symposium on Applications of Laser Techniques to Fluid Mechanics, July, 1997, Lisbon Portugal

Structure of a Turbulent Boundary Layer Using a Stereoscopic, Large Format Video-PIV

Z. C. Liu[1], R. J. Adrian[1], C. D. Meinhart[2] and W. Lai[3]

[1]Department of Theoretical and Applied Mechanics
University of Illinois, Urbana, Illinois 61801

[2]Department of Mechanical and Environmental Engineering
University of California, Santa Barbara, California 93106

[3]TSI, Inc., St. Paul, Minnesota 55164

Abstract. Development of a stereoscopic particle image velocimeter for the measurement of three-dimensional vectors on a planar domain is described. The camera is based on two large format (2k x 2k) video cameras. Experiments in a turbulent boundary layer at $Re_\theta = 2525$ demonstrate its ability to measure three-dimensional turbulent flow. In addition to the quantitative value of the out-of-plane component, it is found that having the complete three-dimensional vector also significantly improves the qualitative visualization of the flow.

Keywords. particle image velocimetry, boundary layer, three-dimensional, stereoscopic, videographic, turbulence

1 Introduction

The extension of particle image velocimetry to measurement of three-dimensional velocity vector fields is both desirable and achievable. Hinsch (1995) reviews many of the techniques used for three-dimensional measurements. In general, there are two classes of measurement techniques: those which measure three-dimensional velocity vectors on full three-dimensional domains , i.e. volumes, and those which measure three-dimensional vectors on planar domains, i.e. light sheets. Three-dimensional volumetric measurements can be performed by using holographic techniques (Barnhart et al. 1994, Meng and Hussain 1994), by using photogrammetric particle tracking techniques (Maas, Gruen and Papantonoiu 1993, Brodkey 1977, Nishino and Kasagi 1991), or by scanning a laser light sheet rapidly (Brucker 1995). Photogrammetric methods use three or four cameras to view particles from several different directions. Volumetric measurements are in general difficult and require equipment that is rather different than the standard planar PIV equipment which uses a laser light sheet and a single camera. Holographic systems require a unique set of techniques and apparatus, and three-dimensional particle tracking systems involve multiple cameras and specialized software, plus they place demands on the optical access required to view the flow field.

However, the most demanding aspect of volumetric measurements is the sheer number of velocity vectors that are obtained from the measurements. A modest 100 x 100 x 100 velocity grid measurement yields one million vectors, and if these measurements are to be repeated hundreds or even thousands of times for the purpose of statistical averaging, the amount of data is overwhelming. Thus, while volumetric measurements can be extraordinarily valuable for studying instantan-eous flow field structures, they are often too rich in data to be used on a routine basis with present equipment.

An intermediate approach to three-dimensional volumetric measurements with PIV is to perform measurements of a three-dimensional velocity vector filed on a planar domain. In this way, the number of vectors is the same as in ordinary two-dimensional planar PIV, but the third component contributes significantly to the experimenter's capability to visualize the flow and it is valuable for the purposes of quantitative analysis of the flow. Further, full measurement of the three-dimensional vectors eliminates the perspective error that is inherent in monocular PIV systems (Adrian 1991, Prasad and Adrian 1993). This error can be quite significant if the out-of-plane velocity component has non-negligible magnitude relative to the in-plane components and/or the angular field-of-view is not small.

Systems operating with this capability can be designed using various aspects of image correlation (Robinson and Rockwell 1993), or using the change in image magnification (Willert and Gharib 1992). However, the most common technique is the well-known stereoscopic method whereby one obtains a pair of image planes, each image plane viewing the particles in the illuminated light sheet plane from a different direction. Displacements of the particle images in the two different views differ because of the different viewing angles, and measurements of these displacements can be used to solve for the full three-dimensional displacement of particles. This technique can be applied to individual particles, as in particle tracking systems, or it can be used in combination with correlation techniques that measure the displacements of groups of particles, or even continuous grey level patterns. The correlation approach is desirable because the three-dimensional velocity vectors can be obtained on uniform grids, and the problem of matching individual pairs is solved automatically. The limitation of the stereo technique is that to assure good measurements of the out-of-plane component of velocity, the viewing angles between the stereo lenses must be substantial, of the order of thirty degrees, and the time between exposures of the images must be small enough that the particles remain within the thickness of the light sheet. These requirements have been explored amply and shown to be achievable and not inconvenient in practice (Prasad and Adrian 1993, Troy and Adrian 1996)).

Stereographic systems of conventional form have been used by Arroyo and Greated (1991), Prasad and Adrian (1993), and Troy and Adrian (1996). These experiments each used two photographic recordings, interrogated the recordings to obtain displacements and then solved the stereo equations to obtain three-dimensional velocity vectors. The difficulty with using photographic film is the problem of registering the two pieces of film with respect to each other so that measurements of displacements locations in the first image can be identified precisely with locations in the second. This problem must be faced for each pair of photographic exposures that are taken in a stereo-photographic system. The registration process is more laborious and time consuming than developing the film in itself, and it places a fundamental constraint on the utility of this method.

To make feasible extensive quantitative analysis of three-dimensional vector fields, it is necessary to make the image acquisition analysis process easy enough and fast enough to permit thousands of images to be taken for averaging purposes. To this end, the photographic recording must be eliminated, and replaced by videographic recording. Video cameras in a stereo system can be registered during the construction of the stereo camera and by this means they can be aligned once and for all, thereby eliminating the need for any future efforts to register the two images. The registration can be accomplished by means of mechanical alignment during construction, or by electronic image processing alignment during the experimental stage. The electronic alignment is necessary in situations where an aberrating medium is placed between the object plane and the cameras, so that the registration that was originally achieved in the construction of the camera is distorted by the medium. In this case, a calibration procedure can be used to correct for aberrations, provided they are not so severe as to completely destroy the images of the individual particles.

In this paper, the design and construction of a stereographic camera based on video recording is described, and some of the procedures used in the aberration calibration and correction process are explained. The system is applied to the visualization of three-dimensional vector field structure in a low Reynolds number turbulent boundary layer.

2 Stereoscopic PIV System

The stereoscopic camera is shown schematically in Fig. 1. The camera lenses L_1 and L_2 lie in a plane parallel to the object plane defined by the laser light sheet. They are offset laterally by d_{L1} and d_{L2}, respectively. Since the camera arrays also lie in a plane parallel to the object plane, the magnification of each lens is constant, equal to

$$M_O = d_i / d_o \tag{1}$$

where d_i and d_o are the image and object distances, respectively. Within the angular field indicated by the solid lines each lens images point objects with resolution and distortion that are within the specification of the lens. In the object plane the size of the region that can be imaged with good resolution by both lenses, the 'joint field', is determined by the intersection of the respective angular fields. The lateral offsets d_{L1} and $d_{L2}$are chosen to be the largest values that still permit the angular fields to overlap over the desired region in the object plane, i.e. to make the joint field as large as possible while still keeping the angle between the lenses large enough to achieve small measurement error for the z-component of velocity. Nominally, d_{L1} and d_{L2} are equal, and for the present system the angle between the axes of the two lenses is 24.4 degrees. According to Prasad and Adrian (1993) the error in the z-component of velocity is about twice the error in the in-plane components of velocity when the angle between the axes of the lenses exceeds 30 degrees.

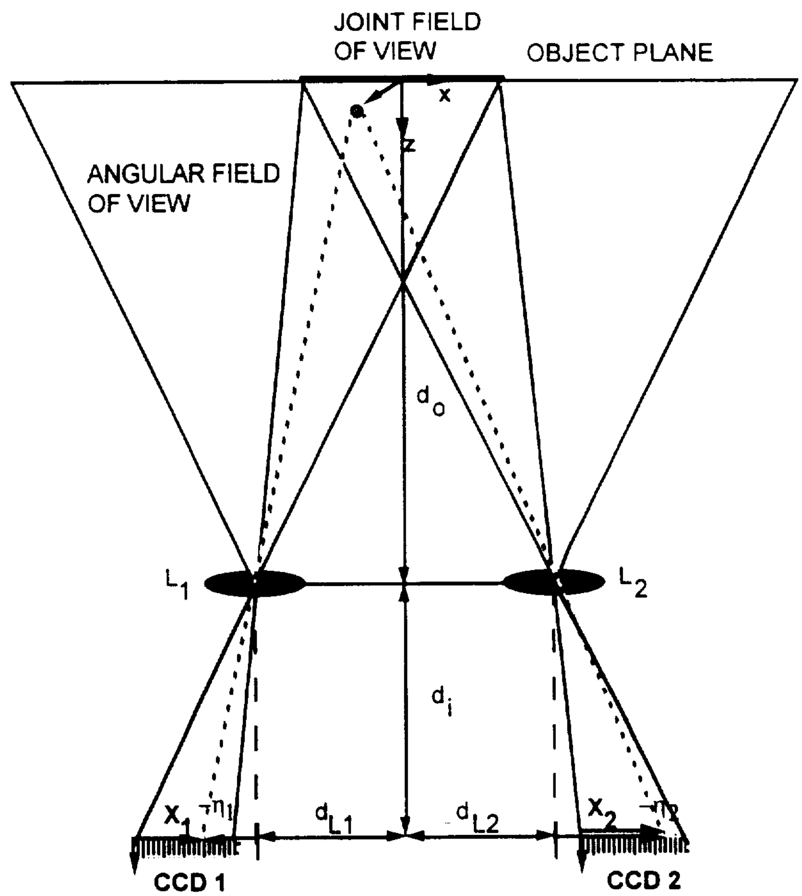

Figure 1. Schematic of the stereoscopic camera.

The field of view of each CCD array is simply the area of the array mapped onto the object plane. As in Prasad and Adrian (1993) and Troy and Adrian (1996), offsetting the cameras laterally in the image plane overlaps their respective fields of view. Each camera sees nearly the same region in the flow, thereby maximizing the size of the joint field of view. This design is relatively simple to lay out, but care must be taken to ensure that tolerances in the focal lengths of the lenses and in the orientations of the individual elements of the system are taken into account, either mechanically or by careful calibration.

The joint field of view increases by more than a factor of two if the lenses are tilted towards each other so as to completely overlay their angular fields. This method, called 'angular displacement', has been discussed by Gauthier and Riethmuller (1988) and more recently by Prasad and Jensen (1996). Its principle advantage is the larger joint field of view, and its principal disadvantage is the more complicated alignment that is needed, which also involves using magnification that varies across the field of view. Only the simpler lateral offset method will be considered here, although angular offset can be easily accommodated by the present camera.

The camera uses two Kodak *Megaplus 4.2* CCD video cameras, each with 2029 x 2044 pixels resolution. Since the camera is intended to image a 100-300 mm field of view in the flow, and the CCD chip dimension is only 18 mm, the magnification of the system is substantially less than unity. M_O= 0.23 for the present system. Two Nikon *EL-Nikkor* 302.5 mm F/5.6 enlarging lenses that are optimized for normal operation at large magnification are mounted in the reverse direction to provide the appropriate small magnification with little aberration, minimal distortion and large aperture (53.6 mm). The angular field of view of this lens is 57 degrees. Images are normally acquired at F/8. The video cameras and two lenses are mounted on a 3-D translation stage. The stage provides the six degrees of freedom needed to position and focus the cameras and to accurately register them with respect to each other. For the wind tunnel experiments, d_o = 1617 mm and d_i = 372mm.

The rest of the stereoscopic system is shown in Fig. 2. Illumination of the flow field is provided by two Continuum Nd:YAG with a wavelength of 532 nm at pulse frequency of 10 Hz. Each laser delivers up to 200 mJ/pulse of energy with a pulse duration of 8 nsec. The laser beam is formed into a sheet of about 1 mm thickness in the test section. To resolve directional ambiguity, image shifting is normally needed. However image shifting was not used for the present measurements because of the strong mean flow in comparison with the turbulence intensity in the boundary layer. If directional ambiguity were a problem the cameras could be replaced by 1k x 1k cross-correlating cameras. Timing for the sequence of laser pulses and the image capture with the cameras is provided by a TSI, Inc. *Laser Pulse* synchronizer box.

3 Image Analysis

The image analysis procedure is described in Figure 3. The capture and analysis of images is controlled by *INSIGHT*, a Windows-based software package supplied by

TSI, Inc. The software controls the simultaneous image capture by the video cameras. Image analysis to determine the velocity vectors on each image plane is performed by the autocorrelation technique. The outcome of the analysis is a pair of displacement vector fields, $\Delta\mathbf{X}_1$ and $\Delta\mathbf{X}_2$ that are functions of the camera coordinates, $\mathbf{X}_1$ and $\mathbf{X}_2$. The vectors in these fields are tested for validity using procedures described by Meinhart, *et al.* (1994), and invalid vectors are either replaced by second or third choice vectors from the autocorrelation interrogation, replaced by interpolated values, or simply removed. Typically the fraction of invalid vectors is less than 2%. Corrections for the mis-registration of the images, and the variation of the magnification factors are all performed by post-processing software. Effectively, this step maps the displacement vectors from a grid in each of the CCD array image planes onto a common grid in the object plane. This step is necessary in general if there is any distorting medium between the object plane and the camera, such as a water-air interface (Prasad and Adrian 1993). In the present experiment the 6.35 mm thick glass windows of the wind tunnel cause the images to be displaced laterally by 0.55 mm. Since this shift is nearly constant, it is most easily accounted for by mechanical alignment in the initial layout of the cameras. In more general cases of nonlinear distortion it is better to calibrate the distortion and correct for it in software. The procedures for doing this will be discussed in another paper.

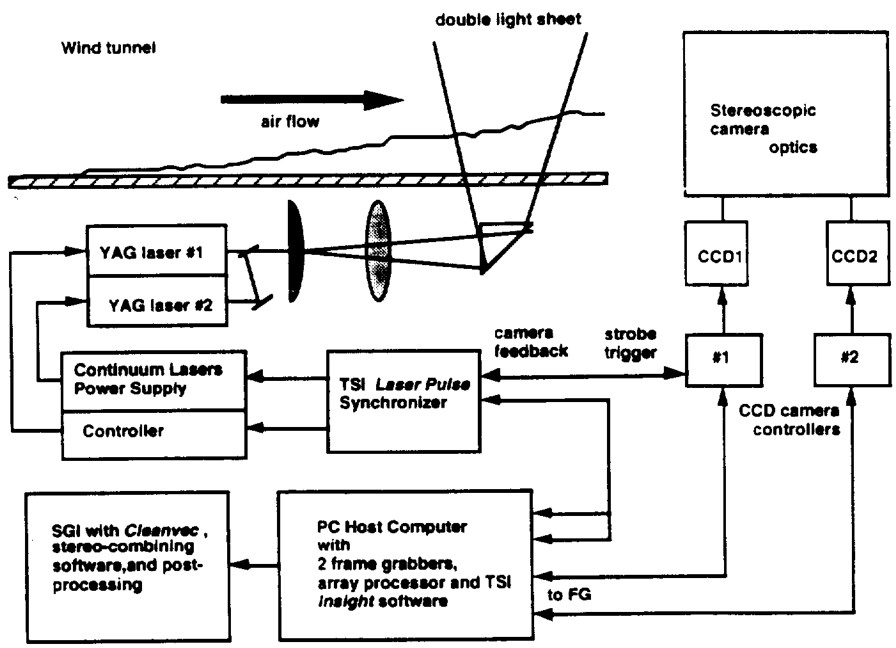

Figure 2. Stereoscopic system.

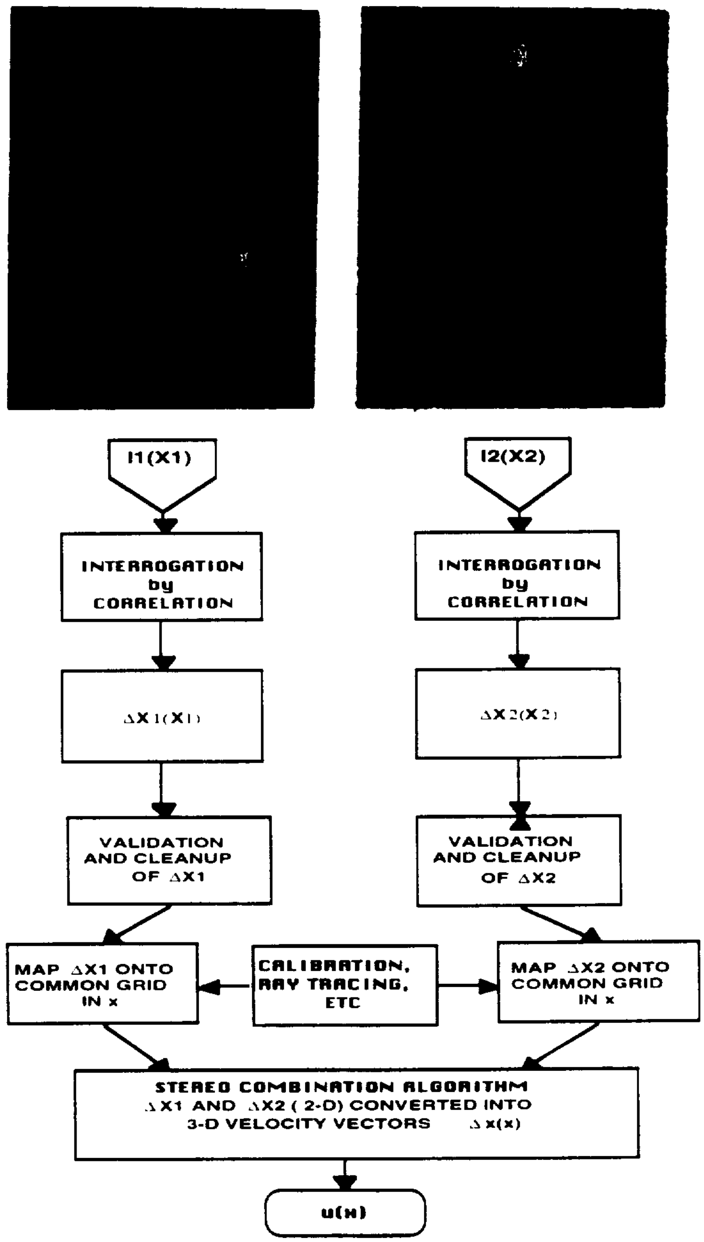

Figure 3. Image analysis procedure.

Lastly, the stereoscopic equations are applied to compute the third-component of the velocity vectors. For the ideal camera system shown in Fig. 1 the image of a point particle at $\mathbf{x}$ is mapped by lens L_ℓ onto a location η_ℓ in the $\mathbf{X}_\ell$-plane on camera ℓ given by

$$-\eta_1 = \frac{-d_i}{d_o - z}\left[\left(x + d_{L1}\right)\hat{\mathbf{x}} + y\hat{\mathbf{y}}\right], \tag{2a}$$

$$-\eta_2 = \frac{-d_i}{d_o - z}\left[\left(x - d_{L2}\right)\hat{\mathbf{x}} + y\hat{\mathbf{y}}\right] \tag{2b}$$

where carats denote unit vectors. Three-dimensional displacement of an image by an amount $\Delta\mathbf{x}$ results in two-dimensional displacements $\Delta\eta_1$ and $\Delta\eta_2$ that are found by taking the increment of (2). The results depend on the z-location of the particle at the time of the first exposure. However, PIV correlation analysis gives an estimate of the *volume average* of the displacements of the particles that lie in the measurement volume defined by the thickness of the light sheet and the area of the interrogation spot. By integrating the equation for the image displacement over such a volume and by making use of the normally large ratio of the object distance to the light sheet thickness, Prasad and Adrian (1993) have shown that the volume averaged displacements (denoted by overbars) are given by

$$\Delta\overline{\eta}_1 = M_o\left[\overline{\Delta x} + \frac{\left(x+d_{L1}\right)}{d_o}\overline{\Delta z}\right]\hat{\mathbf{x}} + M_o\left[\overline{\Delta y} + \frac{y}{d_o}\overline{\Delta z}\right]\hat{\mathbf{y}} \tag{3a}$$

$$\Delta\overline{\eta}_2 = M_o\left[\overline{\Delta x} + \frac{\left(x-d_{L2}\right)}{d_o}\overline{\Delta z}\right]\hat{\mathbf{x}} + M_o\left[\overline{\Delta y} + \frac{y}{d_o}\overline{\Delta z}\right]\hat{\mathbf{y}} \tag{3b}$$

Given the two two-dimensional vector displacements $\overline{\Delta\eta_1}, \overline{\Delta\eta_2}$ three of the equations in (3) can be solved for $\overline{\Delta x}, \overline{\Delta y}, \overline{\Delta z}$, and the fourth can be used to obtain a second estimate of $\overline{\Delta y}$. Alternatively, the two $\overline{\Delta y}$-equations can be added together to obtain a slightly more robust equation. The three-dimensional velocity vector is, of course, found by dividing three dimensional displacement by the time between exposures.

Figures 4a and 4b show the displacement fields that result on cameras 1 and 2, respectively when the object field undergoes a pure, uniform displacement in the z-direction. Solving equation (3) for the three-dimensional particle displacement yields the field shown in Fig. 4c.

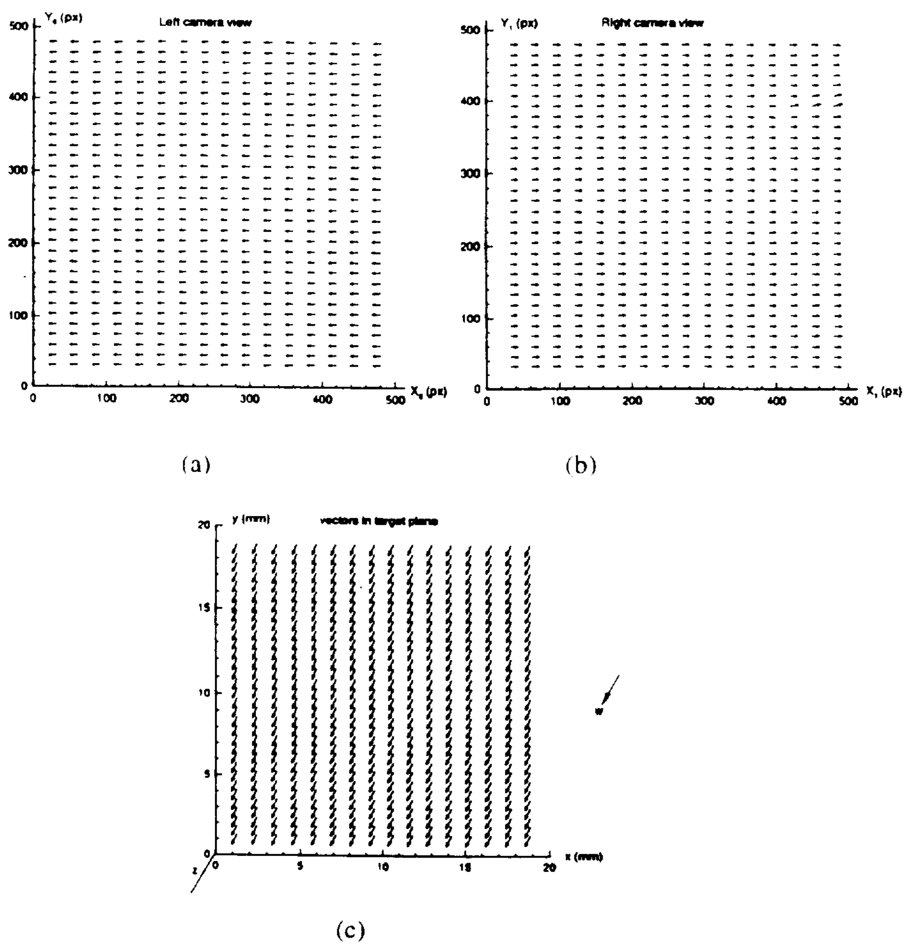

Figure 4. Image displacement fields due to a pure out-of-plane translation. (a) Camera 1, $\Delta \mathbf{X}_1(\mathbf{X}_1)$; (b) $\Delta \mathbf{X}_2(\mathbf{X}_2)$; (c) Three-dimensional displacement field $\Delta \mathbf{x}(\mathbf{X})$.

4. Turbulent boundary layer measurements

The turbulent boundary layer measurements were performed in an eiffel-type low-turbulence boundary layer wind tunnel with a working section 914 mm wide x 457 mm high x 6.096 m long. The free stream turbulence intensity at the test section inlet is less than 0.2% for free stream velocities less than 10 ms^{-1}. The turbulent boundary layer was produced on a flat plate placed 100 mm above the floor of the test section. The boundary layer was tripped by the a 4.7 mm diameter wire which spanned the boundary layer plate just downstream of the leading edge. Optical access to the boundary layer was provided from the side by float glass windows, and from below by 610 mm wide x 2.748 m long float-glass windows embedded in the boundary layer plate. The stereo PIV measurements presented here were performed at a free stream velocity of 3.4 ms^{-1}, which produced a Reynolds number based on the momentum thickness θ of $Re_\theta = 2575$ at the location of the measurements. All measurements were made with the light sheet in the x-y plane of the flow, where x is streamwise and y is normal to the wall.

Figure 5 shows the (u,v) components of one realization of the flow field. A constant convection velocity of 3000 mm s^{-1} has been subtracted to make the

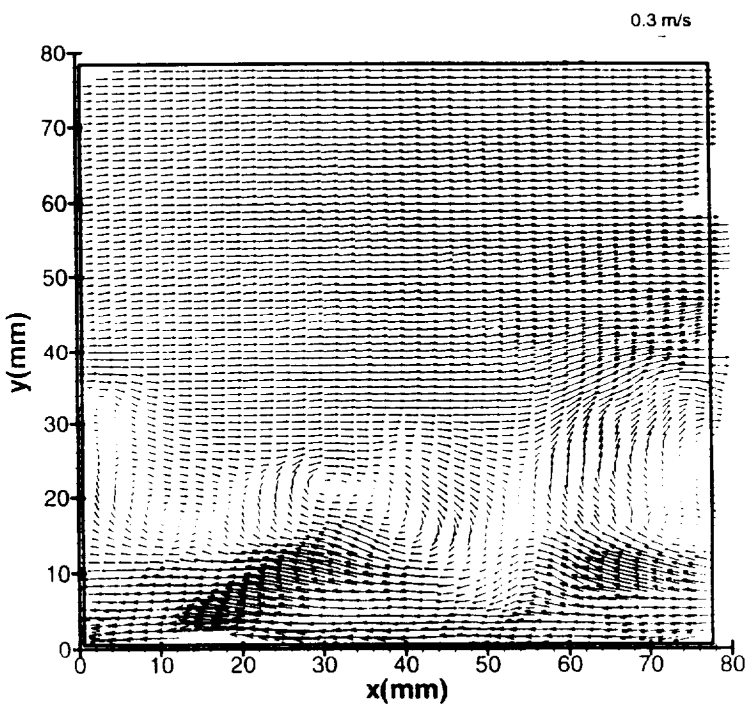

Figure 5. Sample realization of the streamwise-wall normal $(x$-$y)$ vector field of the turbulent boundary layer. $Re_\theta = 2525$. The free stream velocity is 3.4 ms^{-1}, and a constant convection velocity of 3.0 ms^{-1} has been subtracted to reveal the eddy structure more clearly. The vectors are the projection of the three-dimensional vectors onto the x-y plane.

fluctuations more visible. The edge of the turbulent boundary layer lies about 50 mm above the wall in this realization. As observed in earlier work, there is an internal layer character close to the wall that grows in a manner similar to that of an ordinary boundary layer. This internal layer, seen as a region of uniformly low momentum that extends from the wall up to about 20 mm, is capped by a collection of several intense spanwise vortices.

The w-component of velocity is shown in Figure 6 by plotting its contours in gray-level form. It is immediately clear that the values of the w-component are frequently as large or larger than the other components. There is a tendency for the w-contours to align along the same ~30 degree inclination that is observed in the u-v field. There is also a tendency for the sign of the w-component to be positive on one side of a 30 degree line and negative on the other, indicating a large-scale rotation about an inclined axis. The strength of the w-component and its coherent organization shows that one must be very cautious in interpreting two-dimensional projections of the three-dimensional vectors such as the field in Figure 5.

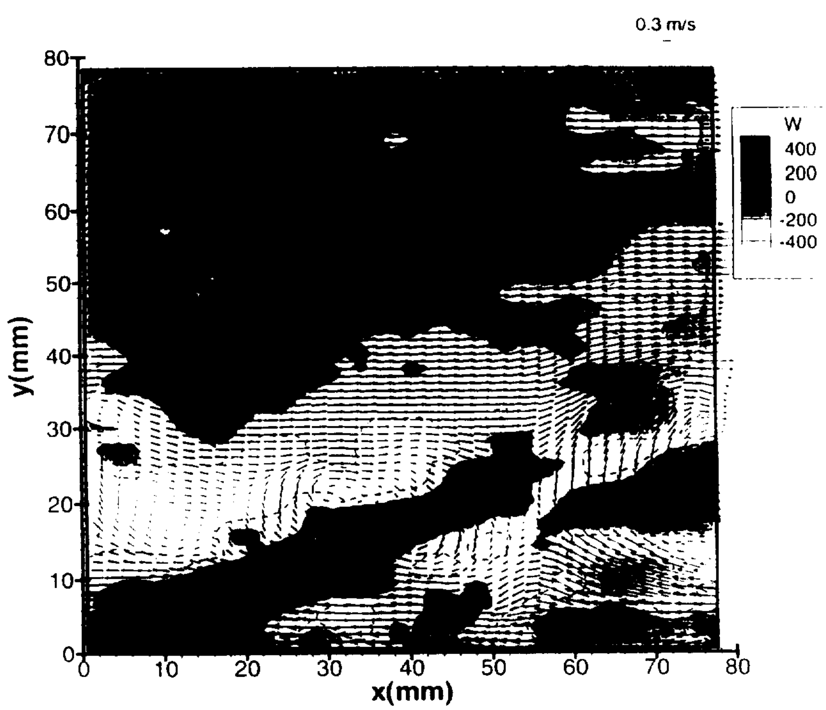

Figure 6. Sample realization of the streamwise-wall normal $(x$-$y)$ vector field from Fig. 5 with the field of the spanwise turbulent velocity superimposed on it in the form of grey-level contours. The heavy line denotes the zero value of the w-component, and dashed lines denote negative values.

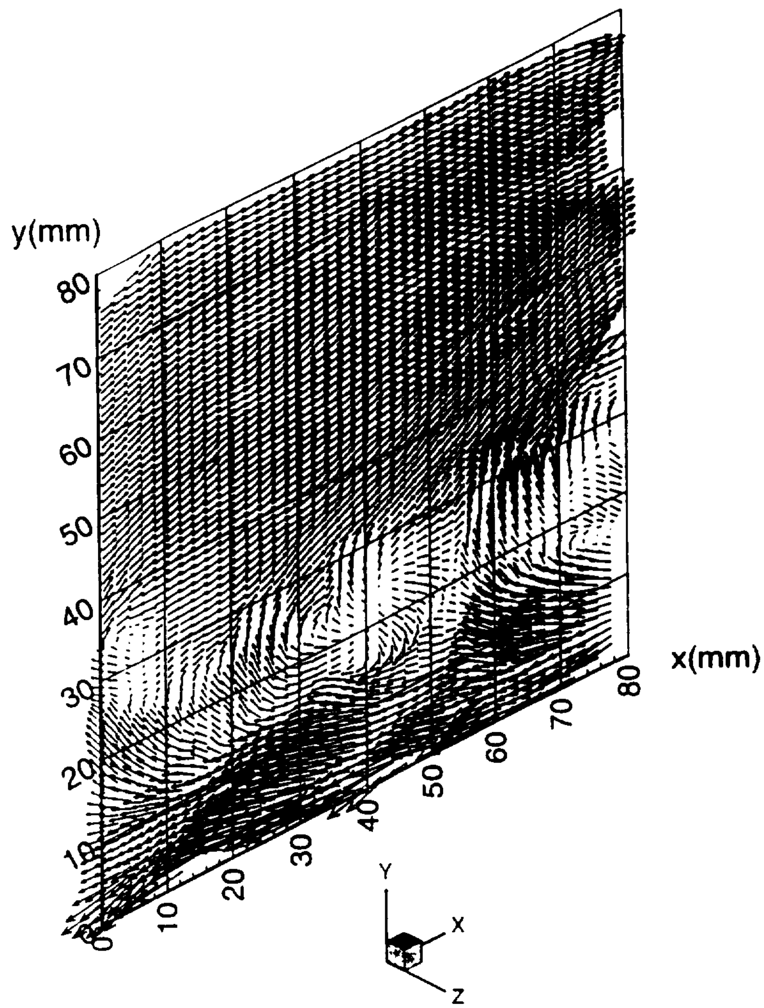

Figure 7. Three-dimensional vectors on the x-y plane in oblique view. All parameters as in Fig. 5.

The entire plane of three-dimensional vectors is shown in oblique view in Figure 7. The growth of an internal low-momentum layer is very evident. The pattern is attributed to the alignment of inclined hairpin vortices (c.f. Meinhart and Adrian 1997, Zhou, Adrian and Balachandar 1997) In Figure 8 profiles of the v-w components of the velocity vector are plotted for several different x-locations.

These vectors demonstrate very clearly the oscillation in the sign of the w-component with increasing distance above the wall, and the inclination of the pattern at about thirty degrees with respect to the wall. The oscillation is consistent with the presence of inclined, hairpin-like vortices, as observed by Meinhart (1994) and Meinhart and Adrian (1997).

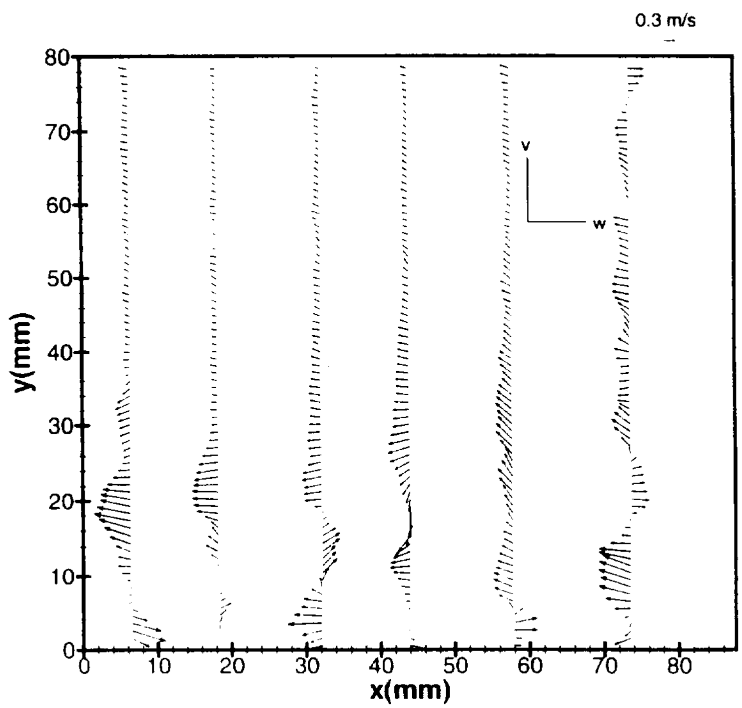

Figure 8. Profiles of $(v(x,y),\ w(x,y))$ at several x-locations from the vector field in Fig. 7.

5 Summary

Stereoscopic PIV corrects for perspective error as well as providing the out-of-plane component. It requires a special camera, but otherwise it uses essentially the same hardware as a monocular PIV. Videographic recording increases the ease of use substantially by eliminating the necessity of registering photographic images. The present system has been used to measure thousands of frames in experiments that required only a few days to set up. Although the video-based PIV system has

The present system has been used to measure thousands of frames in experiments that required only a few days to set up. Although the video-based PIV system has less resolution than a photographic system, it is encouraging that it is still able to resolve many of the important features of the turbulent boundary layer that have only recently been discovered by photographic PIV. The results are very similar to those obtained by photographic PIV. Thus, while the video system clearly has a role to play in the acquisition of data for analysis by statistical averaging, it also provides enough resolution to be of considerable value in visualizing the flow.

Acknowledgments

This work was supported by NASA Grant NAG 1 1530, the United States Office of Naval Research, and TSI, Inc.

References

Adrian, R. J. 1991, Particle-imaging techniques for experimental fluid mechanics, *Ann. Rev. Fluid Mech.*, vol. 23, pp. 261-304.

Arroyo, M.P. and Greater, C.A., 1991, Stereoscopic particle image velocimetry, *Meas. Sci. Tech.* Vol. 2, pp. 1181-1186.

Barnhart, D.H., Adrian, R.J. and Papen, G.C., 1994, Phase-conjugate holographic system for high resolution particle image velocimetry, *Appl. Opt.*, Vol. 33, pp. 7159-7170.

Brodkey, R.S., 1977, Stereoscopic visual studies of complex turbulence shear flows, *Flow Visualization*, (Ed. Asanuma, T.), New York: McGraw-Hill, pp. 117-122.

Brucker, C., 1995, 3D-PIV using stereoscopy and a scanning light-sheet: application to the 3D unsteady sphere wake flow. *Flow Visualization VII*, (Ed. Crowder, J.P.), New York: Begell House, pp. 715-720.

Gauthier, V. and Riethmuller, M.R., 1988, Application of PIDV to complex flows: measurements of the third component. *Von Karman Institute Lecture Series on Particle Image Velocimetry.* Rhodes-Saint-Genese: von Karman Institute for Fluid Dymanics.

Hinsch, K.D., 1995, Three-dimensional particle velocimetry, *Meas, Sci. Tech.* Vol. 6, pp. 741-753.

Meinhart, C.D., 1994, Investigation of turbulent boundary-layer using particle image velocimetry, Ph. D. thesis, University of Illinois.

Meinhart, C.D. and Adrian, R.J., 1997, The coherent structure of the overlap region in a turbulennt boundary layer, (manuscript in preparation).

Meng, H. and Hussain, F., 1995, In-line recording and off-axis viewing (IROV) technique for holographic particle velocimetry, *Appl. Opt.* Vol. 34. pp. 1827-1840.

Nishino, Y., Kasagi, S. and Hirata, M., 1989, Three-dimensional particle tracking velocimetry based on automated digital image processing. *ASME J.* Vol. 111, pp. 384-391.

Prasad, A.K., and Adrian, R.J., 1993, Stereoscopic particle image velocimetry applied to liquid flows, *Exp. Fluids*, Vol. 15, pp. 49-60.

Prasad, A.K., Jensen, K., 1995, Scheimpflug stereocamera for particle image velocimetry in liquid flows, *Appl. Opt.*, Vol. 34, pp. 7092-7099.

Robinson, O., and Rockwell, D., 1993, Construction of three-dimensional images of flow structure via particle tracking techniques, *Exp. Fluids*, Vol. 14, pp. 257-270.

Troy, V. and Adrian R. J., 1996, Investigation of turbulent penetrative convection, 1996, *Eighth International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, pp.23.1.1-23.1.8.15

Willert, C.E. and Gharib, M., 1991, Digital particle image velocimetry, *Exp. Fluids*, Vol. 10, pp. 181-193.

Zhou J., Adrian R. J., Balachandar S. and Kendall T., 1997, Hairpin vortices in near-wall turbulence and their regeneration mechanisms, (manuscript in preparation).

APPENDIX B

Source Code for Combination of the Two-dimensional Displacement Fields from Stereoscopic PIV Images into Three-dimensional Vectors

Z. C. Liu

```
      program stereo

c*****************************************************************************
c       This program is for combining the measurements of particle
c       displacements from two CCD cameras and to
c       calculate three components of velocity field.
c       The calculations are for the stereo-system with no window and no change
c       in the index of refraction in the optical paths.
c       The .p file assigns the basic parameters:
c               xregion, yregion: the mm size of the camera views
c               imax, jmax: the grid numbers in x and y directions
c               cmag: the average magnification of the two cameras
c               xshift, yshift: the px shifts of view1(right) to view0(left)
c               dt: the time separation of two exposures in seond
c               infile0, infile1: the .s files of camera0 and camera1,
c                       from cleanvec output, --- x,y,u,v in px
c               outvec: output file of three velocity components in mm/s
c               interpolation: guassian interpolation
c*****************************************************************************

        include 'stereo.h'

        open (7,file='stereo.p',status='unknown')

        read (7,'(a)') infile0
        read (7,'(a)') infile1
        read (7,'(a)') outvec
        read (7,'(a)') aorbin
        read (7,*) convect
        read (7,*) imax,jmax
        read (7,*) xshift,yshift
        read (7,*) cmag
        read (7,*) dt

        close (7)

        do=f*(1.+1./cmag)
        di=f*(1.+cmag)
        ms=cmag*s

        call data_in

c       xs and ys are in px, to convert them into grid numbers:

        xs=xshift/float(imax)
        ys=yshift/float(jmax)

c       To find the shifts of view1 to view0 in a grid unit:

        dx=xs-int(xs)
        dy=ys-int(ys)

        call gaussian
        call combine
        call vec_out

 2000   stop
        end
```

```
      subroutine data_in

      include 'stereo.h'

      real xkern,ykern,choice

c     the input data are displacemants in px
c     multiply pxsize to convert px to mm in camera planes
c     the input data files are the .s files from cleanvec

      open (10,file=infile0,status='unknown',form='unformatted')

      read (10) imax,jmax,xkern,ykern
      read (10) choice,choice,choice,choice,choice

      do 200 j=1,jmax
        do 100 i=1,imax

              read (10) x0(i,j),y0(i,j),u0(i,j),v0(i,j)
c             write (*,*) x0(i,j),y0(i,j),u0(i,j),v0(i,j)

c     The position data in .s file have already changed to mm, but the
c     velocity data are still pixels, not converted to mm not mm/s

                    x0(i,j)=-x0(i,j)
                    y0(i,j)=-y0(i,j)

c     The bad vectors in .s file are given 9.e+6 values, and now changed
c     to zeros here
c     Also, if desire to look at flow field from a moving frame (ccd cameras)
c     with a convection velocity, subtract convect from u. The resulting three
c     components of velocity contain the difference only in u, while the v and
c     w do not change.

              if (u0(i,j).eq.9.e+6) then
                u0(i,j)=0.0
              else
                u0(i,j)=-u0(i,j)*pxsize
              endif

              if (v0(i,j).eq.9.e+6) then
                v0(i,j)=0.0
              else
                v0(i,j)=-v0(i,j)*pxsize
              endif

 100  continue
 200  continue

      close (10)
      write (*,'(a)') infile0,'has been read.'

      open (15,file=infile1,status='unknown',form='unformatted')

      read (15) imax,jmax,xkern,ykern
      read (15) choice,choice,choice,choice,choice

      do 400 j=1,jmax
        do 300 i=1,imax
```

```fortran
            read (15) x1(i,j),y1(i,j),u1(i,j),v1(i,j)
c           write (*,*) x1(i,j),y1(i,j),u1(i,j),v1(i,j)

                    x1(i,j)=-x1(i,j)
                    y1(i,j)=-y1(i,j)

            if (u1(i,j).eq.9.e+6) then
              u1(i,j)=0.0
            else
              u1(i,j)=-u1(i,j)*pxsize
            endif

            if (v1(i,j).eq.9.e+6) then
              v1(i,j)=0.0
            else
              v1(i,j)=-v1(i,j)*pxsize
            endif

 300    continue
 400    continue

        close (15)

        write (*,'(a)') infile1,'has been read.'
        write (*,*) 'data input is done!'

        return
        end
```

```fortran
      subroutine gaussian

      include 'stereo.h'

c             the 2D-gaussian interpolation:

c          nw *----------------* ne
c             |      a     |  b  |
c             |            |     |
c             |--------(i,j)---|
c             |            |     |
c             |      c     |  d  |
c             |            |     |
c             |            |     |
c          sw *----------------* se

c       the values of x,y,u,v at (i,j) - the regular grid as the same as
c       the view0 - are the interpolated values at 4 points around (i,j).

c       The interpolation is conducted by convolving the data with a gaussian
c       function.

        deltax=x0(2,1)-x0(1,1)
        deltay=y0(1,2)-y0(1,1)

        do 200 j=1,jmax
          do 100 i=1,imax

              inw=i+int(xs)
              ine=i+int(xs+1.)
              isw=inw
              ise=ine

              jnw=j+int(ys)
              jne=j+int(ys+1.)
              jsw=jnw
              jse=jne

              a=dx*dx/deltax/deltax+dy*dy/deltay/deltay
              b=(deltax-dx)*(deltax-dx)/deltax/deltax+dy*dy/deltay/deltay
              c=dx*dx/deltax/deltax+(deltay-dy)*(deltay-dy)/deltay/deltay
              d=(deltax-dx)*(deltax-dx)/deltax/deltax+
     +                 (deltay-dy)*(deltay-dy)/deltay/deltay

              a=exp(-6.*a)
              b=exp(-6.*b)
              c=exp(-6.*c)
              d=exp(-6.*d)

         x1p(i,j)=(xshift+float(i*igrid))*pxsize
         y1p(i,j)=(yshift+float(j*jgrid))*pxsize
         u1p(i,j)=u1(inw,jnw)*a+u1(ine,jne)*b+u1(isw,jsw)*c+u1(ise,jse)*d
         v1p(i,j)=v1(inw,jnw)*a+v1(ine,jne)*b+v1(isw,jsw)*c+v1(ise,jse)*d

         u1p(i,j)=u1p(i,j)/(a+b+c+d)
         v1p(i,j)=v1p(i,j)/(a+b+c+d)

  100    continue
  200    continue
```

```
write (*,*) 'Gaussian interpolation is done!'

return
end
```

```fortran
      subroutine combine

      include 'stereo.h'

c     convert registration shifts of camera1 to camera0 into mm in camera
c     plane

      xsmm=xshift*pxsize
      ysmm=yshist*pxsize

c     calculate the image region size into mm

      xregion=(x0(imax,1)-x0(1,1))*pxsize
      yregion=(y0(1,jmax)-y0(1,1))*pxsize

c     grid points (i,j) are in target plane, while the conjugate grid points   c
c     the conversion of (x0,y0) in the image plane to (x,y) in the target
c     plane is being done in the following do loop
c     the combining formula have also taken care of the conversion of the      c

      do 450 j=1,jmax
        do 440 i=1,imax

              ip=imax-i+1
              jp=jmax-j+1

      x(i,j)=(xregion-x0(ip,jp))/cmag
      y(i,j)=(yregion-y0(ip,jp))/cmag

c     if there are bad vector in (i,j) point of .s file, do not combine
c     and set them as zeros

        if (u0(ip,jp).eq.0.0.or.v0(ip,jp).eq.0.0.or.u1p(ip,jp).eq.0.0.
     +          or.v1p(ip,jp).eq.0.0) then
        w(i,j)=0.0
        u(i,j)=0.0
        v(i,j)=0.0

        else

      w(i,j)=do/(1.-ms/(u0(ip,jp)-u1(ip,jp)))

      u(i,j)=(u1p(ip,jp)*(x0(ip,jp)-0.5*(ms+xregion))-
     +        u0(ip,jp)*(x1p(ip,jp)+0.5*(ms-xregion)-xsmm))/
     +        cmag/(ms-u0(ip,jp)+u1p(ip,jp))

      v(i,j)=w(i,j)*0.5/di*(v0(ip,jp)+v1p(ip,jp)-
     +        yregion+y0(ip,jp)+y1p(ip,jp-ysmm))-
     +        0.5/cmag*(v0(ip,jp)+v1p(ip,jp))

      u(i,j)=u(i,j)/dt
      v(i,j)=v(i,j)/dt
      w(i,j)=w(i,j)/dt

        endif

c     write (*,*) x(i,j),y(i,j),u(i,j),v(i,j),w(i,j)
 440  continue
 450  continue
      write (*,*) 'Combination of the two views has been done!'
```

```
return
end
```

```fortran
      subroutine vec_out

      include 'stereo.h'

      b=0.0
      cc=convect*pxsize/dt/cmag

c     Two kinds of output formats can be selected: tecplot ascii with
c     a subtraction of a constant convection veclocity in streamwise
c     direction; or binary without any subtraction.

 300  if (aorbin.eq.'tecplot') then
      open (30,file=outvec,status='unknown',form='formatted')

      write (30,*) 'TITLE = "BOUNDARY LAYER FLOW FIELD, Uo=3.4 m/s"'
      write (30,*) 'VARIABLES = X, Y, Z, U, V, W'
      write (30,*) 'ZONE T="CHANNEL_TEC",I=',imax,'J=',jmax,'F=POINT'

      do 600 j=jmax,1,-1
        do 500 i=imax,1,-1

              if (abs(w(i,j)).gt.2000.) then
                w(i,j)=0.0
                u(i,j)=0.0
                v(i,j)=0.0
              else
              endif

              if (u(i,j).ne.0.0) then
                u(i,j)=u(i,j)-cc
              else
              endif
              write (30,700) x(i,j),y(i,j),b,u(i,j),v(i,j),w(i,j)

 500  continue
 600  continue

      elseif (aorbin.eq.'binary') then

      open (30,file=outvec,status='unknown',form='unformatted')

              write (30) imax,jmax,xkern,ykern
              write (30) choice,choice,choice,choice,choice

      do 680 j=jmax,1,-1
        do 650 i=imax,1,-1

              if (abs(w(i,j)).gt.2000.) then
                w(i,j)=0.0
                u(i,j)=0.0
                v(i,j)=0.0
              else
              endif

              write (30) x(i,j),y(i,j),b,u(i,j),v(i,j),w(i,j)

 650  continue
 680  continue

      else
```

```
      write (*,*) 'enter output format(tecplot or binary):'
      read (*,*) aorbin
      goto 300
      endif

      write (*,*) 'Calculation of three velocity components has been done!'
      write (*,*) '    and stored in file: ', outvec
      write (*,*) 'The format of output is: ',aorbin
      write (*,*)
      write (*,*) 'The calculated velocity field is from a viewer with a
     +   constant convection velocity of:', cc, '(mm/s)'
      write (*,*)

      close (30)

700   format (6f11.4)
      return
      end
```

```fortran
c       stereo.h

        parameter (m=128,n=128)
        parameter (pxsize=0.009)
        parameter (s=700.4,f=302.3)

        common x0,y0,u0,v0,x1,y1,u1,v1,x1p,y1p,u1p,v1p,x,y,u,v,w,
     +          xshift,yshift,xregion,yregion,imax,jmax,dx,dy,dt,
     +          igrid,jgrid,cmag,cmag0,cmag1,do,di,ms,xs,ys,convect,
     +          infile0,infile1,outvec,interpolate,aorbin

        real x0(m,n),y0(m,n),u0(m,n),v0(m,n)
        real x1(m,n),y1(m,n),u1(m,n),v1(m,n)
        real x1p(m,n),y1p(m,n),u1p(m,n),v1p(m,n)
        real x(m,n),y(m,n),u(m,n),v(m,n),w(m,n)
        real cmag0(m,n),cmag1(m,n),convect

        real do,di,ms,xshift,yshift,xs,ys,dx,dy,dt
        integer imax,jmax

        character*12 infile0,infile1,outvec,aorbin
c       character *1 interpolate
```

```
wt0_01.s
wt1_01.s
wt_01
binary
0.0
40,80
25.,-9.
0.229
0.00010

stereo.p:
input .s file from cleanvec for camera0, binary
input .s file from cleanvec for camera1, binary
output file for 3 components, bin: wt_07, tecplot: wt_07.dat
output format: tecplot(ascii) or binary
convection velocity, Uc in px, 7.63 for wt_01, 11.45 for others.
        It is used only for tecplot output.
        This number was obtained from cleanvec
imax,jmax -- grid number
x and y registration shifts of camera1 to camera0, px
averaged magnification of the stereo-camera system
delta t, time separation of the two pulses, sec
        0.0001 for wt_01
        0.00015 for others.
```

APPENDIX B

Source Code for Combination of the Two-dimensional Displacement Fields from Stereoscopic PIV Images into Three-dimensional Vectors

Z. C. Liu

```
      program stereo

c*************************************************************************
c     This program is for combining the measurements of particle
c     displacements from two CCD cameras and to
c     calculate three components of velocity field.
c     The calculations are for the stereo-system with no window and no change
c     in the index of refraction in the optical paths.
c     The .p file assigns the basic parameters:
c           xregion, yregion: the mm size of the camera views
c           imax, jmax: the grid numbers in x and y directions
c           cmag: the average magnification of the two cameras
c           xshift, yshift: the px shifts of view1(right) to view0(left)
c           dt: the time separation of two exposures in seond
c           infile0, infile1: the .s files of camera0 and camera1,
c                 from cleanvec output, --- x,y,u,v in px
c           outvec: output file of three velocity components in mm/s
c           interpolation: guassian interpolation
c*************************************************************************

      include 'stereo.h'

      open (7,file='stereo.p',status='unknown')

      read (7,'(a)') infile0
      read (7,'(a)') infile1
      read (7,'(a)') outvec
      read (7,'(a)') aorbin
      read (7,*) convect
      read (7,*) imax,jmax
      read (7,*) xshift,yshift
      read (7,*) cmag
      read (7,*) dt

      close (7)

      do=f*(1.+1./cmag)
      di=f*(1.+cmag)
      ms=cmag*s

      call data_in

c     xs and ys are in px, to convert them into grid numbers:

      xs=xshift/float(imax)
      ys=yshift/float(jmax)

c     To find the shifts of view1 to view0 in a grid unit:

      dx=xs-int(xs)
      dy=ys-int(ys)

      call gaussian
      call combine
      call vec_out

 2000 stop
      end
```

```fortran
      subroutine data_in

      include 'stereo.h'

      real xkern,ykern,choice

c     the input data are displacemants in px
c     multiply pxsize to convert px to mm in camera planes
c     the input data files are the .s files from cleanvec

      open (10,file=infile0,status='unknown',form='unformatted')

      read (10) imax,jmax,xkern,ykern
      read (10) choice,choice,choice,choice,choice

      do 200 j=1,jmax
        do 100 i=1,imax

              read (10) x0(i,j),y0(i,j),u0(i,j),v0(i,j)
c             write (*,*) x0(i,j),y0(i,j),u0(i,j),v0(i,j)

c     The position data in .s file have already changed to mm, but the
c     velocity data are still pixels, not converted to mm not mm/s

                    x0(i,j)=-x0(i,j)
                    y0(i,j)=-y0(i,j)

c     The bad vectors in .s file are given 9.e+6 values, and now changed
c     to zeros here
c     Also, if desire to look at flow field from a moving frame (ccd cameras)
c     with a convection velocity, subtract convect from u. The resulting three
c     components of velocity contain the difference only in u, while the v and
c     w do not change.

              if (u0(i,j).eq.9.e+6) then
                u0(i,j)=0.0
              else
                u0(i,j)=-u0(i,j)*pxsize
              endif

              if (v0(i,j).eq.9.e+6) then
                v0(i,j)=0.0
              else
                v0(i,j)=-v0(i,j)*pxsize
              endif

 100  continue
 200  continue

      close (10)
      write (*,'(a)') infile0,'has been read.'

      open (15,file=infile1,status='unknown',form='unformatted')

      read (15) imax,jmax,xkern,ykern
      read (15) choice,choice,choice,choice,choice

      do 400 j=1,jmax
        do 300 i=1,imax
```

```
                 read (15) x1(i,j),y1(i,j),u1(i,j),v1(i,j)
c                write (*,*) x1(i,j),y1(i,j),u1(i,j),v1(i,j)

                         x1(i,j)=-x1(i,j)
                         y1(i,j)=-y1(i,j)

                 if (u1(i,j).eq.9.e+6) then
                   u1(i,j)=0.0
                 else
                   u1(i,j)=-u1(i,j)*pxsize
                 endif

                 if (v1(i,j).eq.9.e+6) then
                   v1(i,j)=0.0
                 else
                   v1(i,j)=-v1(i,j)*pxsize
                 endif

 300      continue
 400      continue

          close (15)

          write (*,'(a)') infile1,'has been read.'
          write (*,*) 'data input is done!'

          return
          end
```

```
      subroutine gaussian

      include 'stereo.h'

c            the 2D-gaussian interpolation:

c        nw *-----------------* ne
c           |       a    |  b  |
c           |            |     |
c           |--------(i,j)---|
c           |            |     |
c           |       c    |  d  |
c           |            |     |
c           |            |     |
c        sw *-----------------* se

c     the values of x,y,u,v at (i,j) - the regular grid as the same as
c     the view0 - are the interpolated values at 4 points around (i,j).

c     The interpolation is conducted by convolving the data with a gaussian
c     function.

      deltax=x0(2,1)-x0(1,1)
      deltay=y0(1,2)-y0(1,1)

      do 200 j=1,jmax
        do 100 i=1,imax

            inw=i+int(xs)
            ine=i+int(xs+1.)
            isw=inw
            ise=ine

            jnw=j+int(ys)
            jne=j+int(ys+1.)
            jsw=jnw
            jse=jne

            a=dx*dx/deltax/deltax+dy*dy/deltay/deltay
            b=(deltax-dx)*(deltax-dx)/deltax/deltax+dy*dy/deltay/deltay
            c=dx*dx/deltax/deltax+(deltay-dy)*(deltay-dy)/deltay/deltay
            d=(deltax-dx)*(deltax-dx)/deltax/deltax+
     +             (deltay-dy)*(deltay-dy)/deltay/deltay

            a=exp(-6.*a)
            b=exp(-6.*b)
            c=exp(-6.*c)
            d=exp(-6.*d)

      x1p(i,j)=(xshift+float(i*igrid))*pxsize
      y1p(i,j)=(yshift+float(j*jgrid))*pxsize
      u1p(i,j)=u1(inw,jnw)*a+u1(ine,jne)*b+u1(isw,jsw)*c+u1(ise,jse)*d
      v1p(i,j)=v1(inw,jnw)*a+v1(ine,jne)*b+v1(isw,jsw)*c+v1(ise,jse)*d

      u1p(i,j)=u1p(i,j)/(a+b+c+d)
      v1p(i,j)=v1p(i,j)/(a+b+c+d)

 100  continue
 200  continue
```

```
write (*,*) 'Gaussian interpolation is done!'

return
end
```

```fortran
      subroutine combine

      include 'stereo.h'

c     convert registration shifts of camera1 to camera0 into mm in camera
c     plane

      xsmm=xshift*pxsize
      ysmm=yshist*pxsize

c     calculate the image region size into mm

      xregion=(x0(imax,1)-x0(1,1))*pxsize
      yregion=(y0(1,jmax)-y0(1,1))*pxsize

c     grid points (i,j) are in target plane, while the conjugate grid points
c     the conversion of (x0,y0) in the image plane to (x,y) in the target
c     plane is being done in the following do loop
c     the combining formula have also taken care of the conversion of the

      do 450 j=1,jmax
        do 440 i=1,imax

                ip=imax-i+1
                jp=jmax-j+1

      x(i,j)=(xregion-x0(ip,jp))/cmag
      y(i,j)=(yregion-y0(ip,jp))/cmag

c     if there are bad vector in (i,j) point of .s file, do not combine
c     and set them as zeros

        if (u0(ip,jp).eq.0.0.or.v0(ip,jp).eq.0.0.or.u1p(ip,jp).eq.0.0.
     +          or.v1p(ip,jp).eq.0.0) then
        w(i,j)=0.0
        u(i,j)=0.0
        v(i,j)=0.0

        else

      w(i,j)=do/(1.-ms/(u0(ip,jp)-u1(ip,jp)))

      u(i,j)=(u1p(ip,jp)*(x0(ip,jp)-0.5*(ms+xregion))-
     +          u0(ip,jp)*(x1p(ip,jp)+0.5*(ms-xregion)-xsmm))/
     +          cmag/(ms-u0(ip,jp)+u1p(ip,jp))

      v(i,j)=w(i,j)*0.5/di*(v0(ip,jp)+v1p(ip,jp)-
     +          yregion+y0(ip,jp)+y1p(ip,jp-ysmm))-
     +          0.5/cmag*(v0(ip,jp)+v1p(ip,jp))

      u(i,j)=u(i,j)/dt
      v(i,j)=v(i,j)/dt
      w(i,j)=w(i,j)/dt

        endif

c     write (*,*) x(i,j),y(i,j),u(i,j),v(i,j),w(i,j)
 440  continue
 450  continue
      write (*,*) 'Combination of the two views has been done!'
```

```
return
end
```

```
      subroutine vec_out

      include 'stereo.h'

      b=0.0
      cc=convect*pxsize/dt/cmag

c     Two kinds of output formats can be selected: tecplot ascii with
c     a subtraction of a constant convection veclocity in streamwise
c     direction; or binary without any subtraction.

 300  if (aorbin.eq.'tecplot') then
      open (30,file=outvec,status='unknown',form='formatted')

      write (30,*) 'TITLE = "BOUNDARY LAYER FLOW FIELD, Uo=3.4 m/s"'
      write (30,*) 'VARIABLES = X, Y, Z, U, V, W'
      write (30,*) 'ZONE T="CHANNEL_TEC",I=',imax,'J=',jmax,'F=POINT'

      do 600 j=jmax,1,-1
        do 500 i=imax,1,-1

              if (abs(w(i,j)).gt.2000.) then
                w(i,j)=0.0
                u(i,j)=0.0
                v(i,j)=0.0
              else
              endif

              if (u(i,j).ne.0.0) then
                u(i,j)=u(i,j)-cc
              else
              endif
              write (30,700) x(i,j),y(i,j),b,u(i,j),v(i,j),w(i,j)

 500  continue
 600  continue

      elseif (aorbin.eq.'binary') then

      open (30,file=outvec,status='unknown',form='unformatted')

              write (30) imax,jmax,xkern,ykern
              write (30) choice,choice,choice,choice,choice

      do 680 j=jmax,1,-1
        do 650 i=imax,1,-1

              if (abs(w(i,j)).gt.2000.) then
                w(i,j)=0.0
                u(i,j)=0.0
                v(i,j)=0.0
              else
              endif

              write (30) x(i,j),y(i,j),b,u(i,j),v(i,j),w(i,j)

 650  continue
 680  continue

      else
```

```
      write (*,*) 'enter output format(tecplot or binary):'
      read (*,*) aorbin
      goto 300
      endif

      write (*,*) 'Calculation of three velocity components has been done!'
      write (*,*) '    and stored in file: ', outvec
      write (*,*) 'The format of output is: ',aorbin
      write (*,*)
      write (*,*) 'The calculated velocity field is from a viewer with a
     +  constant convection velocity of:', cc, '(mm/s)'
      write (*,*)

      close (30)

700   format (6f11.4)
      return
      end
```

```
c       stereo.h

        parameter (m=128,n=128)
        parameter (pxsize=0.009)
        parameter (s=700.4,f=302.3)

        common x0,y0,u0,v0,x1,y1,u1,v1,x1p,y1p,u1p,v1p,x,y,u,v,w,
     +          xshift,yshift,xregion,yregion,imax,jmax,dx,dy,dt,
     +          igrid,jgrid,cmag,cmag0,cmag1,do,di,ms,xs,ys,convect,
     +          infile0,infile1,outvec,interpolate,aorbin

        real x0(m,n),y0(m,n),u0(m,n),v0(m,n)
        real x1(m,n),y1(m,n),u1(m,n),v1(m,n)
        real x1p(m,n),y1p(m,n),u1p(m,n),v1p(m,n)
        real x(m,n),y(m,n),u(m,n),v(m,n),w(m,n)
        real cmag0(m,n),cmag1(m,n),convect

        real do,di,ms,xshift,yshift,xs,ys,dx,dy,dt
        integer imax,jmax

        character*12 infile0,infile1,outvec,aorbin
c       character *1 interpolate
```

```
wt0_01.s
wt1_01.s
wt_01
binary
0.0
40,80
25.,-9.
0.229
0.00010

stereo.p:
input .s file from cleanvec for camera0, binary
input .s file from cleanvec for camera1, binary
output file for 3 components, bin: wt_07, tecplot: wt_07.dat
output format: tecplot(ascii) or binary
convection velocity, Uc in px, 7.63 for wt_01, 11.45 for others.
        It is used only for tecplot output.
        This number was obtained from cleanvec
imax,jmax -- grid number
x and y registration shifts of camera1 to camera0, px
averaged magnification of the stereo-camera system
delta t, time separation of the two pulses, sec
        0.0001 for wt_01
        0.00015 for others.
```