IOP Publishing Measurement Science and Technology

Meas. Sci. Technol. **19** (2008) 075403 (10pp) doi:10.1088/0957-0233/19/7/075403

Performance prediction of point-based three-dimensional volumetric measurement systems

E C Graff[1] **and M Gharib**

California Institute of Technology, Pasadena, CA 91125, USA

E-mail: pelesl@its.caltech.edu

Received 14 December 2007, in final form 11 May 2008
Published 12 June 2008
Online at stacks.iop.org/MST/19/075403

Abstract

Point-based three-dimensional volumetric measurement systems are defined as multi-view vision systems which reconstruct a three-dimensional scene by first identifying key points on the views and then performing the reconstruction. Examples of these are defocusing digital particle image velocimetry (DDPIV) (Pereira *et al* 2000 *Exp. Fluids* **29** S78–84) and 3D particle tracking velocimetry (3DPTV) (Papantoniou and Maas 1990 *5th Int. Symp. on the Application of Laser Techniques in Fluid Mechanics*) which reconstruct clouds of flow tracers in order to estimate flow velocities. The reconstruction algorithms in these systems are variations of an epipolar line search. This paper presents a generalized error analysis of such methods, both in reconstruction precision (error in the reconstructed scene) and reconstruction quality (number of ambiguities or 'ghosts' produced).

Keywords: defocusing, PTV, PIV, photogrammetry, reconstruction

1. Introduction

The development of the digital camera has brought about several methods for imaging a three-dimensional scene by using multiple two-dimensional pixel arrays. Cross-correlation-based methods such as that in Kanade *et al* (1996) generate depth maps of scenes without specifically identifying any features in them. By far, the most prominent method is sometimes referred to as 3D particle tracking velocimetry (Papantoniou and Maas 1990) and has found wide use in fluid mechanics measurements, but its underlying principles are the foundation of many methods employed in computer vision. Defocusing digital particle image velocimetry (Pereira *et al* 2000) is another method developed for use in fluid mechanics. Recently, tomographic particle image velocimetry (Elsinga *et al* 2006) has introduced a method which yields an analog variant to the results of methods like 3DPTV and DDPIV.

Methods such as 3DPTV and DDPIV are referred to in this paper as 'point-based methods' because the reconstruction of the scene involves first identifying features in the images (in fluid mechanics applications, these would be the images of seeding particles) and then reconstructing their three-dimensional positions by some variation of the epipolar line search methodology. This can include a two-dimensional epipolar line search as analyzed in Maas (1992), a three-dimensional epipolar line search, or backwards ray tracing, as in Chen *et al* (1994), or variations thereof, such as the rectified epipolar search proposed in Ariyawansa and Clarke (1998). The result of these algorithms is a list of discrete three-dimensional points representing the original features in the images. In Tomo-PIV, a three-dimensional bitmap of the scene is obtained through a MART algorithm. This result is like an analog version of the typical epipolar line search methods because along with the feature locations the output contains the intensity of said features—however, the method never identifies features on the image and never performs a search.

This paper is a theoretical study of the reconstruction performance of point-based three-dimensional imaging methods. The reconstruction can be qualified in two ways: its error and quality. The error of the reconstruction is the difference between the reconstructed three-dimensional locations of the features and the actual location of these features. The quality can be thought of as a signal-to-

[1] Currently at Measurement Science Enterprise, Pasadena, CA 91105, USA.

0957-0233/08/075403+10$30.00

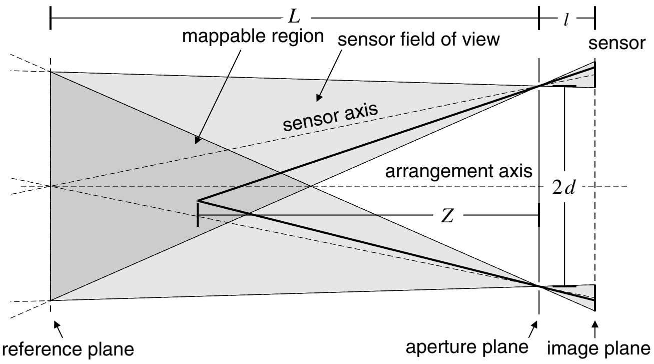


Figure 1. Pinhole-optics diagram of a two-sensor defocusing arrangement.

noise parameter; it quantifies the number of features in the reconstruction which did not really exist. In the development of epipolar search algorithms, these are referred to as 'ambiguities'; in DDPIV they are normally referred to as 'ghost' or 'noise' points.

The analysis presented here is centered around the methodology of DDPIV (similar in concept to the rectified epipolar search in Ariyawansa and Clarke (1998)), but should be applicable to similar methods which calculate directly the reconstructed coordinates (as opposed to, for example, using a least-squares approach; see Maas *et al* (1993)).

2. Pinhole-optics ray tracing

Figure 1 shows a stereoscopic photogrammetric setup. For simplicity, we show only two cameras in a two-dimensional arrangement. We present a setup in which the sensors are parallel, and convergence of the fields of view is established by shifting the sensors relative to the lens axes and all reconstructed points are to be between the *reference plane* and the cameras. The reference plane is defined as the plane at which the fields of view of the cameras are identical. We further assume that the lenses and sensors are identical. Any point within the mappable region can be reconstructed since it is visible in both cameras. Because there are no lenses *per se* we define the sensor axis as the line connecting the center of the sensor to its corresponding pinhole. In a typical 3DPTV setup, this would be coincident with the lens axis; both the lens plane and the sensor behind it would be perpendicular to this line. The defocusing arrangement is that as pictured. From the viewpoint of pinhole optics, the only difference between the two arrangements is the angle of the sensors.

L is the distance from the aperture plane to the reference plane, *l* is the distance between the aperture plane and the image plane, 2*d* is the distance between the two apertures, and *Z* is the distance between a space point and the aperture plane. In this depiction the apertures are centered about the optical axis of the arrangement so that the distance between an aperture and the center of the aperture plane is *d*. The parameters have been chosen to match the derivations presented in Willert and Gharib (1992) and Pereira and Gharib (2002). Here we expand these analyses into an arbitrary camera arrangement in 3D space. A single aperture–sensor pair from such a setup is shown in figure 2.

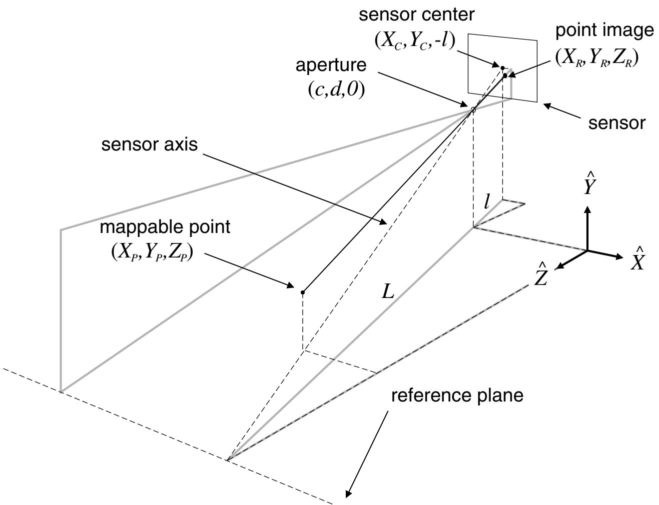


Figure 2. Schematic for 3D pinhole optics, showing a point in space (X_P, Y_P, Z_P), an aperture at $(c, d, 0)$, the point's image on the sensor at (X_R, Y_R, Z_R), and the sensor's center $(X_C, Y_C, -l)$. Two sets of similar triangles are marked: one allows for the calculation of the lateral offset of the sensor relative to the aperture, and the other one can be used to calculate the field of view at the reference plane.

We have chosen to focus our sensors at the reference plane. This is not necessary, but if the assumption is made, and we place an imaginary lens of focal length f at the aperture, we can relate l, L and f together by the thin lens equation

$$\frac{1}{L} + \frac{1}{l} = \frac{1}{f}. \tag{1}$$

If we call the center of the sensor point $\boldsymbol{C} = (X_C, Y_C, Z_C)$, then based on similar triangles (see figure 2) we arrive at the coordinates of the sensor behind the aperture at $(c, d, 0)$:

$$X_C = \frac{cL}{L-f}$$
$$Y_C = \frac{dL}{L-f} \tag{2}$$
$$Z_C = -\frac{fL}{L-f}$$

with the origin of the aperture coordinate system being the intersection of the arrangement axis with the aperture plane.

The image coordinates of a mappable point $\boldsymbol{P} = (X_P, Y_P, Z_P)$ can be obtained through several methods, including similar triangle calculations, a coordinate transformation (as frequently found in the literature) or by intersecting three-dimensional planes. For a defocusing arrangement, these reduce to

$$x = \frac{f}{L-f}\frac{c(L-Z_P) - LX_P}{Z_P}$$
$$y = \frac{f}{L-f}\frac{d(L-Z_P) - LY_P}{Z_P} \tag{3}$$

with the origin being the center of the sensor. The quantity $\frac{f}{L-f}$ is the optical magnification M of the system. It is the proportion between any two corresponding sides of the similar triangles.

Because all the sensors are parallel to the aperture plane, the epipolar lines are all parallel in the defocusing arrangement. This is not necessarily true in a typical 3DPTV arrangement. Thus in DDPIV, if the images of two cameras are superimposed, the distance between the pixel coordinates of two corresponding point images (x_i, y_i) and (x_j, y_j) (where i and j are the different cameras) is only a function of the Z coordinate of the point—by making the sensors parallel to each other, the magnification is only a function of Z. In 3DPTV, the same distance is a function of X and Y along the directions in which the sensor axes are not parallel since any rotation to the sensor equates to a variable magnification for a given Z across the image space.

The distance between the images of the same point on two different cameras for a given point location $\boldsymbol{P}$ is

$$\begin{aligned} b_{ij} &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ &= M\frac{L - Z_P}{Z_P}\sqrt{(c_i - c_j)^2 + (d_i - d_j)^2}. \end{aligned} \tag{4}$$

We define the aperture separation s_{ij} as the physical distance between the two cameras and solve for Z_P to obtain

$$Z_{P_{(ij)}} = \frac{L}{1 + \frac{b_{ij}}{M}\frac{1}{s_{ij}}}. \tag{5}$$

For featureless objects (such as particles) two cameras are not enough for the matching step; at least three must be used—otherwise there is an intolerable number of ghost particles even at very low seeding densities[2]. Z_P in practice is calculated using the average separations $\bar{b}$ and $\bar{s}$, which is equivalent to averaging the individual values of Z_P. With Z_P known, the values for X_P and Y_P can be calculated from equation (3):

$$\begin{aligned} X_P &= \bar{c}\left(1 - \frac{Z_P}{L}\right) - \bar{x}M\frac{Z_P}{L} \\ Y_P &= \bar{d}\left(1 - \frac{Z_P}{L}\right) - \bar{y}M\frac{Z_P}{L} \end{aligned} \tag{6}$$

where the bar indicates the arithmetic mean.

The sensitivity of a pair of cameras to the third dimension is defined in Willert and Gharib (1992) as the derivative of b with respect to Z_P. Here we define it as the negative of that derivative, so as to clarify that a larger magnitude means 'more sensitive':

$$-\frac{\partial b_{ij}}{\partial Z_P} = \frac{ML}{Z_P^2}s_{ij}. \tag{7}$$

Sensitivity increases as the focal lengths of the lenses increase, the distance to the reference plane decreases, and the camera separation increases—it is exactly analogous to the numerical aperture of a lens at a given working distance.

As a guide in characterizing camera arrangements we can take the arrangement-specific part of equation (7) and call it the sensitivity coefficient B:

$$B_{ij} = Ms_{ij}. \tag{8}$$

Note that the sensitivity coefficient could be different for each pair of cameras. For equilateral triangle arrangements, the average sensitivity coefficient $\bar{B}$ is the same as any of the three B_{ij}'s. For non-equilateral triangle arrangements (with more than three cameras), the B_{ij}'s can be maximized—thus maximizing the magnitude of the sensitivity—by pairing cameras that are farthest apart together. As it turns out, a detailed error analysis reveals that the exact opposite should be done (see section 3.1).

3. Reconstruction error

Returning to equation (4), we can estimate the contribution of pixel error in the algorithm which locates the point images to the determination of the coordinates of the point. Without assuming the errors in x and y are independent for a single-point image, then for a given camera pair i, j we have

$$\begin{aligned} \sigma_{b_{ij}}^2 &= \sigma_{x_i}^2\left(\frac{\partial b_{ij}}{\partial x_i}\right)^2 + \sigma_{x_j}^2\left(\frac{\partial b_{ij}}{\partial x_j}\right)^2 \\ &\quad + \sigma_{y_i}^2\left(\frac{\partial b_{ij}}{\partial y_i}\right)^2 + \sigma_{y_j}^2\left(\frac{\partial b_{ij}}{\partial y_j}\right)^2 \\ &\quad + \sigma_{x_i y_i}^2\left(\frac{\partial b_{ij}}{\partial x_i}\frac{\partial b_{ij}}{\partial y_i}\right) + \sigma_{x_j y_j}^2\left(\frac{\partial b_{ij}}{\partial x_j}\frac{\partial b_{ij}}{\partial y_j}\right). \end{aligned} \tag{9}$$

If we assume that all the errors are independent, and that the sensors and lenses are all equal, so that the magnitudes of the errors in x and y in each camera are the same, and we set the error magnitudes to be equal such that $\sigma_{x_i} = \sigma_{x_j} = \sigma_{y_i} = \sigma_{y_j} = \sigma_{\text{img}}$, then the error reduces to

$$\sigma_{b_{ij}} = \sqrt{2}\sigma_{\text{img}}. \tag{10}$$

For a given point, we can rewrite equation (5) as

$$Z_P = \frac{LM\bar{s}}{M\bar{s} + \bar{b}} \tag{11}$$

so that the quantity that depends on software, $\bar{b}$, is easy to isolate. According to our assumptions, $\sigma_{b_{ij}}$ depends only on the error of the point image location σ_{img}, so for a setup with N cameras,

$$\sigma_{\bar{b}} = \sqrt{\frac{2}{N}}\sigma_{\text{img}}, \tag{12}$$

and thus the error in Z_P is

$$\sigma_{Z_P} = \sigma_{\text{img}}\sqrt{\frac{2}{N}}\frac{LM\bar{s}}{(\bar{b} + M\bar{s})^2}. \tag{13}$$

We can simplify this further using equation (4) to substitute for $\bar{b}$ to arrive at[3]

$$\sigma_{Z_P} = \sqrt{\frac{2}{N}}\frac{Z_P^2}{LM\bar{s}}\sigma_{\text{img}}. \tag{14}$$

For the in-plane (X and Y) errors, we start by rewriting X_P from equation (6) as

$$X_P = \bar{c} - \frac{Z_P}{L}\left(\bar{c} + M\bar{x}\right). \tag{15}$$

[2] An exception is in cases where the depth of the measurement volume is specifically limited so that the epipolar bands can be made very short, such as in Bown *et al* (2006).

[3] Note that we can substitute after taking the derivative, because the partial derivative is in fact the partial derivative *evaluated* at a particular point. The distinction is subtle but important.

The two quantities that have measurement error are Z_P and $\bar{x}$. Thus we have

$$\sigma_{X_P}^2 = \left(\frac{\partial X_P}{\partial \bar{x}}\right)^2 \sigma_{\bar{x}}^2 + \left(\frac{\partial X_P}{\partial Z_P}\right)^2 \sigma_{Z_P}^2. \tag{16}$$

The first derivative equals

$$\left(\frac{\partial X_P}{\partial \bar{x}}\right)^2 = \frac{M^2 Z_P^2}{L^2} \tag{17}$$

and the second

$$\left(\frac{\partial X_P}{\partial Z_P}\right)^2 = \frac{(\bar{c} + M\bar{x})^2}{L^2}. \tag{18}$$

We know the error in Z from equation (14), and since $\bar{x}$ is just the average of the individual point image x-coordinates,

$$\sigma_{\bar{x}} = \sqrt{\frac{1}{N}}\sigma_{\text{img}}. \tag{19}$$

Applying this also to Y_P, and replacing $\bar{x}$ and $\bar{y}$ according to equation (3), we arrive at expressions for the error in the reconstructed coordinates as a function of the camera arrangement and the point's real coordinates:

$$\begin{aligned}\sigma_{X_P} &= \sigma_{\text{img}}\frac{1}{\sqrt{N}}\frac{Z_P}{L^2}\frac{1}{\bar{B}}\\ &\quad\times[L^2M^2\bar{B}^2 + 2(LM^2X_P + \bar{c}(M^2(Z_P - L) - Z_P))^2]^{\frac{1}{2}},\\ \sigma_{Y_P} &= \sigma_{\text{img}}\frac{1}{\sqrt{N}}\frac{Z_P}{L^2}\frac{1}{\bar{B}}\\ &\quad\times[L^2M^2\bar{B}^2 + 2(LM^2Y_P + \bar{d}(M^2(Z_P - L) - Z_P))^2]^{\frac{1}{2}},\\ \sigma_{Z_P} &= \sigma_{\text{img}}\sqrt{\frac{2}{N}}\frac{Z_P^2}{L\bar{B}}.\end{aligned} \tag{20}$$

Note that for a symmetric arrangement and a point on the arrangement axis ($X_P = Y_P = 0$) the square-bracketed term in the in-plane error expressions reduces to $LM\bar{B}$ and thus the sensitivity coefficient drops out of the equations. It still remains in the expression for the out-of-plane error.

We can use equation (7) to see how the errors vary with the camera's sensitivity, which is often cited in the DDPIV literature as a benchmark of the performance of a system:

$$\begin{aligned}\sigma_{X_P} &= \sigma_{\text{img}}\frac{1}{\sqrt{N}}\left(\frac{-\partial\bar{b}}{\partial Z_P}\right)^{-1}\frac{1}{Z_PL}\\ &\quad\times[L^2M^2\bar{B}^2 + 2(LM^2X_P + \bar{c}(M^2(Z_P - L) - Z_P))^2]^{\frac{1}{2}},\\ \sigma_{Y_P} &= \sigma_{\text{img}}\frac{1}{\sqrt{N}}\left(\frac{-\partial\bar{b}}{\partial Z_P}\right)^{-1}\frac{1}{Z_PL}\\ &\quad\times[L^2M^2\bar{B}^2 + 2(LM^2Y_P + \bar{d}(M^2(Z_P - L) - Z_P))^2]^{\frac{1}{2}},\\ \sigma_{Z_P} &= \sigma_{\text{img}}\sqrt{\frac{2}{N}}\left(\frac{-\partial\bar{b}}{\partial Z_P}\right)^{-1}.\end{aligned} \tag{21}$$

It is interesting to note then that the ratio $\sigma_{Z_P}/\sigma_{X_P}$ (or $\sigma_{Z_P}/\sigma_{Y_P}$) does not depend on the number of cameras or the sensitivity specifically, but still decreases, all other things constant, as the sensitivity coefficient increases. Moreover, the error in X and Y is minimized for symmetric camera arrangements (where $\bar{c}$ and $\bar{d}$ are zero), and for points centered on the XY plane (on the arrangement axis). The latter agrees with the results presented in Pereira and Gharib (2002), but the effect of arrangement symmetry has been revealed here. The error also decreases as cameras are added (more measurements contribute to the calculation) and the point in question approaches the cameras (the 'numerical aperture' increases).

3.1. Consequences of the actual measurement of the image separation

However, there is a subtle point that has been omitted in this derivation. To balance the errors in the detection of the image points, the average image separation must be defined as

$$\bar{b} = \frac{1}{N}\left(\sum_{i=2}^{N} b_{(i-1)i} + b_{N,1}\right) \tag{22}$$

which, for a three-camera arrangement, is

$$\bar{b}_{(3)} = \tfrac{1}{3}(b_{12} + b_{23} + b_{31}) \tag{23}$$

so that each image point is used twice. To calculate the error in this quantity directly, we take its derivative with respect to the image coordinates. For an N-camera setup, these derivative quantities are of the form

$$\left(\frac{\partial\bar{b}}{\partial x_i}\right)^2 = \frac{1}{N^2}\left(\frac{x_i - x_u}{b_{iu}} - \frac{x_v - x_i}{b_{vi}}\right)^2 \tag{24}$$

with u and v being the other two indices used with index i in equation (22). For example, for $N = 3$, if $i = 1$, then $u = 2$ (from b_{12}) and $v = 3$ (from b_{31}). For N cameras, there are $2N$ such quantities; N pertaining to the image x coordinate and the other N to y. If we assume that the error in the image coordinates is independent and of equal magnitude, then

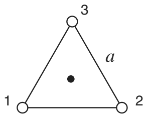
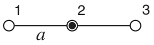
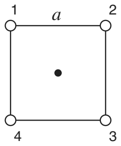
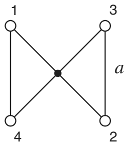
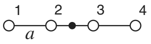
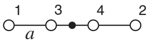
$$\sigma_{\bar{b}} = \sigma_{\text{img}}\sqrt{\sum_{i=1}^{N}\left[\left(\frac{\partial\bar{b}}{\partial x_i}\right)^2 + \left(\frac{\partial\bar{b}}{\partial y_i}\right)^2\right]} = \sigma_{\text{img}}E \tag{25}$$

is the true error in the measurement of $\bar{b}$. The square root quantity E we will refer to here as the error factor. Recall that in our previous, oversimplified derivation (equation (12)) the error factor was $\sqrt{2/N}$. The difference comes from the fact that equation (12) can be thought of as the *average* of the error in the image separation whereas here we have derived the error of the *average* image separation.

If we rewrite equation (20) to include this new error factor, we find that the in-plane errors have NE^2 replacing the multiplicative factor of 2 for the second term inside the square brackets, and that in the out-of-plane error, E simply replaces $\sqrt{2/N}$. Thus for the errors in X and Y its effect is nullified for symmetric arrangements and points on axis, but in Z it appears in the form of $E/\bar{B}$. From this we can see that minimizing E is certainly beneficial for the in-plane errors but may be contradictory in the out-of-plane error because a decrease in E will likely also decrease $\bar{B}$.

The consequences of changing E can be seen by applying it to a few examples. First we write it only as a function of

Table 1. Table of error factors E (equation (25)) for different three- and four-camera setups. The lists under Δc and Δd are the differences in X and Y positions (respectively) for paired cameras as necessitated by equation (26). The values for s are depicted in the figures along with the camera pairing. In the figures, the white circles are camera locations and the black circle is the location of the setup axis (the centroid of the layout pattern).

Setup	Δc	Δd	$\bar{B}/M$	E
	$-a,\ a/2,\ a/2$	$0,\ -a\sqrt{3}/2,$ $a\sqrt{3}/2$	a	1
	$a,\ a,\ 2a$	$0,\ 0,\ 0$	$4a/3$	$2\sqrt{2}/3 \approx 0.94$
	$-a,\ 0,\ a,\ 0$	$0,\ a,\ 0,\ -a$	a	$1/\sqrt{2} \approx 0.71$
	$-a,\ 0,\ a,\ 0$	$a,\ -a,\ a,\ -a$	$(1+\sqrt{2})a/2$	≈ 0.92
	$-a,\ -a,\ -a,\ 3a$	$0,\ 0,\ 0,\ 0$	$3a/2$	$1/\sqrt{2} \approx 0.71$
	$-3a,\ 2a,\ -a,\ 2a$	$0,\ 0,\ 0,\ 0$	$2a$	1

camera arrangement parameters, which is done by substituting for x_i and y_i:

$$\left(\frac{x_i - x_u}{b_{iu}} - \frac{x_v - x_i}{b_{vi}}\right)^2 = \left[\frac{(c_v - c_i)s_{iu} - (c_i - c_u)s_{vi}}{s_{iu}s_{vi}}\right]^2 \qquad (26)$$

(where the appropriate expression involving y and d has been omitted for conciseness).

The camera separations s_{iu} and s_{vi} are never zero, since that would imply a camera is paired with itself. For the same reason, the difference in the X coordinate of paired cameras ($c_v - c_i$ and $c_i - c_u$) cannot be zero concurrently with the appropriate difference in the Y coordinate ($d_v - d_i$ and $d_i - d_u$). One choice to minimize this quantity is to choose an arrangement where *one* of the coordinate differences is zero—that is, a camera arrangement with the cameras distributed along a line. If the arrangement is fixed, then camera pairing should be chosen such that each s_{ij} is minimized—in other words, $\bar{B}$ should be minimized. But $\bar{B}$ is also in the denominator of the Z error (equation (20)) so it is not that simple. These opposing conditions propose that a certain value of $\bar{B}$ should be used if the ratio of error factors is more favorable than the ratio of sensitivity coefficients. In other words, if $\bar{B}_{\mathrm{new}}$ is such that

$$\frac{E_{\mathrm{new}}}{E_{\mathrm{old}}} < \frac{\bar{B}_{\mathrm{new}}}{\bar{B}_{\mathrm{old}}} \qquad (27)$$

(where E_{xxx} is the error factor corresponding to the pairing that produces $\bar{B}_{xxx}$), then the pairing of $\bar{B}_{\mathrm{new}}$ should be used. The error reduction factor with the new choice is simply $(E_{\mathrm{new}}\bar{B}_{\mathrm{old}})/(E_{\mathrm{old}}\bar{B}_{\mathrm{new}})$. Table 1 shows some basic camera arrangements and their values for $\bar{B}$ and E.

For the three-camera case (used almost exclusively in DDPIV systems), there are not many options. The case of the equilateral triangle arrangement has the largest error factor, and the linear arrangement has the lowest. As long as the inter-camera distance in the linear arrangement is greater than $1/\sqrt{2}$ times the side length of the triangle, the linear arrangement will outperform the triangle. As shown, the linear arrangement would have some 30% less error in Z than the triangular arrangement. For a DDPIV system the comparison is not really just since the triangular arrangement requires a smaller off-axis shift of the sensors (and thus the lenses perform better in that region), but in 3DPTV, where there is no shift between the lenses and the sensors, it is equally as easy to implement one arrangement or the other.

With four cameras we have a choice of pairing. As long as the pairing is chosen to minimize $\bar{B}$, regular shapes have an error factor of $1/\sqrt{2}$, which, coincidentally, is the error factor predicted by the simplified expression for the error (equation (12)) for $N = 4$. These include the square and linear arrangements (shown) as well as a rectangle and a sheared rectangle (or rhombus; neither shown). The shapes shown in table 1 with the $\bar{B}$ minimized have an overall advantage in error over the alternative even when factoring in the decrease in $\bar{B}$.

In the linear arrangements, whenever one camera is moved, the error factor does not change, and neither does

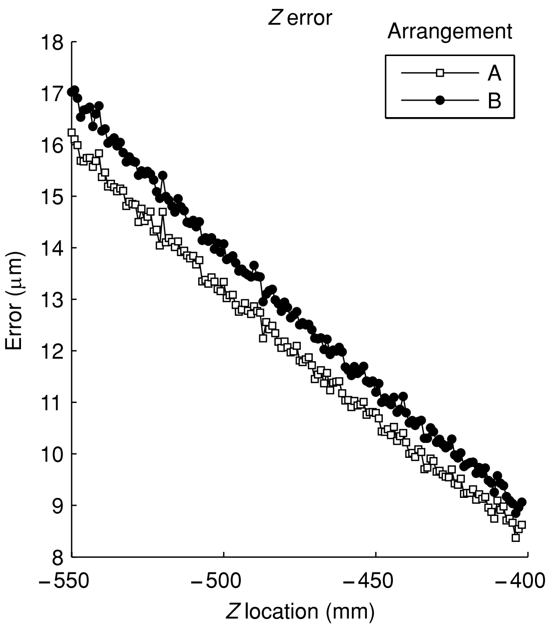


Figure 3. Simulation of the *Z* error in the reconstruction of identical eight-camera setups with identical images. The difference is that 'A' has $\bar{B} = 6.362$ and $E = 0.271$, and 'B' has $\bar{B} = 14.88$ and $E = 0.669$. The reduction in error by using the pairing that produces the shorter $\bar{B}$ matches the prediction exactly. The 'images' were lists of random point image locations in an volume 30 by 20 by 0.001 mm at *Z* increments of 1 mm with artificial 0.20 pixel error added.

the sensitivity coefficient, since one inter-camera distance will increase by the same amount another one is decreasing. Thus it is possible to implement the extremely asymmetric arrangements suggested by Maas (1992) to reduce the number of ghosts without penalty. However, whereas the error factor affects the error in *Z* most directly, the arrangement symmetry affects the errors in *X* and *Y* (as seen by the presence of $\bar{c}$ and $\bar{d}$ in equation (21)). Although the difference is small, it is certainly observable, and if utmost precision is necessary, there is no reason not to employ a symmetric strategy.

The last important point regarding the error factor is that, for the cases where camera pairing is an option ($N > 3$), the error factor can be reduced solely by *software*. That is, given a camera arrangement, the same images can be processed with different camera pairings and different errors will occur. This is shown in figure 3, where simulated images for an eight-camera regular octagon arrangement were processed, first by pairing cameras so as to minimize $\bar{B}$ and then to maximize it.

In practice there is yet another subtlety we have not considered in this analysis. The point image from which the epipolar line search originates is assumed to be perfect, that is, any tolerances in the matching process are relative to this location since the absolute error is unkown. Furthermore, tests of two-dimensional Gaussian fitting algorithms to establish the sub-pixel location of the point image center (as an alternative to center-of-mass methods) show that the *x* error is not necessarily independent of the *y* error for low-noise images. It should be noted that this correlation could be dependent on the actual implementation of the least-squares method used.

4. Reconstruction quality

As discussed above, only two cameras are necessary to calculate the three-dimensional position of a point in space. However, with featureless points such as particles, only the relative position of the images can be used for matching them between cameras, so three is the minimum number of cameras with a tolerable number of mismatches when the measurement volume is deep. The *quality* of the reconstruction refers to how many ghost, or mismatched, points are added to the reconstructed cloud. These are also referred to as 'ambiguities' in the literature (Maas 1992).

The matching process is the second step (after point image identification) in the reconstruction of the scene from two-dimensional images. In theory the epipolar lines are infinitely thin lines, but in practice, a tolerance must be added, and thus they become epipolar bands. Typically they are limited in length by the user, for example, by imaging only a thick slab of the mappable region as frequently done in 3DPTV and Tomo-PIV. In DDPIV, the search is limited only to points in front of the reference plane where the sensitivity is highest.

Ghosts exist in two types: the first is due to one or more point images being within a matching tolerance of another in a single camera image; the second occurs from random chance that the distribution of points in the clouds generates false matches within the matching tolerance. Maas (1992) proposes three types of ghosts can exist; here we classify his ambiguity of the second and third kinds as 'clump' ghosts and the first kind as 'random' ghosts. Clump ghosts' occurence grows as the measurement domain shrinks in depth, and is a result of low relative image shift between clumps of points causing repeating clusters in the images from all the cameras. Random ghosts occur purely by coincidence.

4.1. Clump ghosts

The first type of ghost is the *clump ghost*, and is extremely important as the number of cameras increases and as the depth of the domain decreases. We take the limit of a thin domain—a sheet of points on a particular *XY* plane, in which the relative offset between point images is zero (that is, all the images are identical except for a constant shift in the epipolar directions to account for the sheet's *Z* coordinate). Given a point seeding density ρ in point images per pixels squared of randomly distributed point images, we can model the probability distribution of *n* ghosts with a Poisson distribution of the form

$$P(k, \lambda) = \frac{\mathrm{e}^{-\lambda}\lambda^k}{k!}. \tag{28}$$

We will follow exactly the very famous example of Clarke (1946). First, because our point cloud is essentially a plane in *XY*, and if we assume that it is completely within the field of view of all the cameras, then we can say that ρ is the same for all cameras and is equal to

$$\rho = \frac{m}{A} \tag{29}$$

with *m* denoting the number of points in the cloud and *A* the pixel area which they cover at this given *Z* on the image. Our

Table 2. Table of typical values for the Poisson probability of finding k point images in a small region of the image as described in equation (33). δ is the matching tolerance which defines $a = \pi\delta^2$ and is set to 0.5 pixels. m is chosen so that A is constant. The point image density ρ is in point images per 100 pixels squared.

ρ	k	$\mu P(k, \lambda)$	Measured
0.48	1	747	749
0.48	2	1	1
0.48	3	0	0
1.44	1	2225	2237
1.44	2	13	13
1.44	3	0	0
2.87	1	4400	4453
2.87	2	50	47
2.87	3	0	0

area of interest is the area enclosed by the matching tolerance δ, which, for simplicity, will be treated as a square:

$$a = 4\delta^2 \tag{30}$$

though in software it is usually a circle. Thus our image space is divided into μ regions with

$$\mu = \frac{A}{a} = \frac{m}{\rho a} \tag{31}$$

and our parameter λ is

$$\lambda = \rho a \tag{32}$$

such that the expected number of regions containing k point images equals

$$\begin{aligned}\mu P(k, \lambda) &= \frac{m}{\lambda}\frac{\mathrm{e}^{-\lambda}\lambda^k}{k!}\\ &= m\frac{\mathrm{e}^{-\lambda}\lambda^{k-1}}{k!}.\end{aligned} \tag{33}$$

Some values of this are shown in table 2 compared to measurements taken on simulated clouds. It should be noted that the comparison between calculation and measurement is not completely equivalent. The measurement is that of the number of point images which contain $k - 1$ neighbors within the tolerance δ, whereas the calculation is the number of regions of size a which contain k point images. Thus the probabilities in the measured quantities would be normalized by the number of points whereas the calculated probabilities are normalized by the number of areas (A/a). The discrepancy is not serious as long as the average number of point images within a remains small.

Since we are considering a case with zero relative image shift, with a setup which has N cameras, k points within a matching tolerance δ will generate g_c clump ghosts as defined by

$$g_c = k^N - k \tag{34}$$

assuming each point is guaranteed to have a correct match. This means that two point images within δ of each other in an eight-camera setup will generate 254 ghosts, and three point images will generate a whopping 6558 ghosts.

In real experiments the numbers are not as high as indicated here, since a point image takes up finite space on the image plane and thus can become indistinguishable from another if the two are close enough. The clumping effect in sheet domains is still apparent, even with just three cameras, and care must be taken to post-process these clumps.

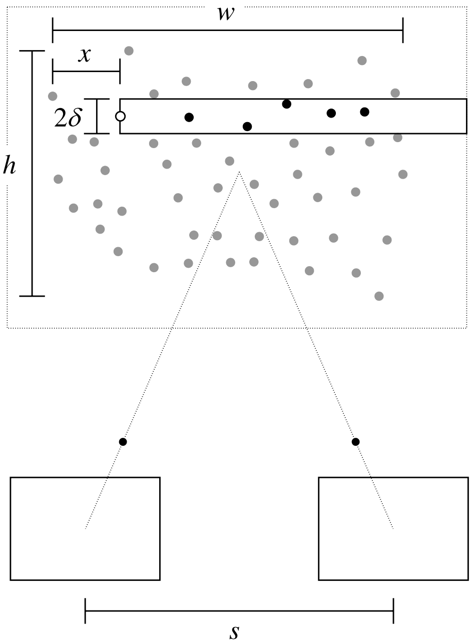


Figure 4. Schematic of the matching process for a two-camera setup. Two pinhole cameras are shown recording a sheet of points at the reference plane. With these conditions, cameras in the defocusing arrangement would record images that are identical. The white point image is that for which the match is being sought. Since there are only two cameras, all the point images within the rectangle of size $(w - x) \times 2\delta$ will be classified as matches.

4.2. *Random ghosts*

Random ghosts occur completely by chance. In volume domains, they are the predominant type of ghost. We start our analysis with the simple case of two horizontally opposed cameras, such that the search criteria for matching is simply 'any point image in image 2 that is within the tolerance of a horizontal line emanating from a point image in image 1' (see figure 4). From this it should be clear why, in practice, two-camera arrangements are unusable with featureless points without limiting the epipolar bands.

We continue our analysis with a sheet of points for simplicity. Let us assume that the sheet is at the reference plane (if it is not, then the corresponding shift between images only increases the width of the search area) so that the images from cameras 1 and 2 will coincide exactly. Moreover, let us assume that the sheet of points is entirely visible and its image has pixel dimensions of w in width and h in height. Given a point image in camera 1 with a certain x coordinate, all the points in camera 2 that match it at a tolerance δ (not including itself) are in a rectangle with dimensions 2δ by $w - x$, so that its area is

$$a(x) = 2\delta\,(w - x)\,. \tag{35}$$

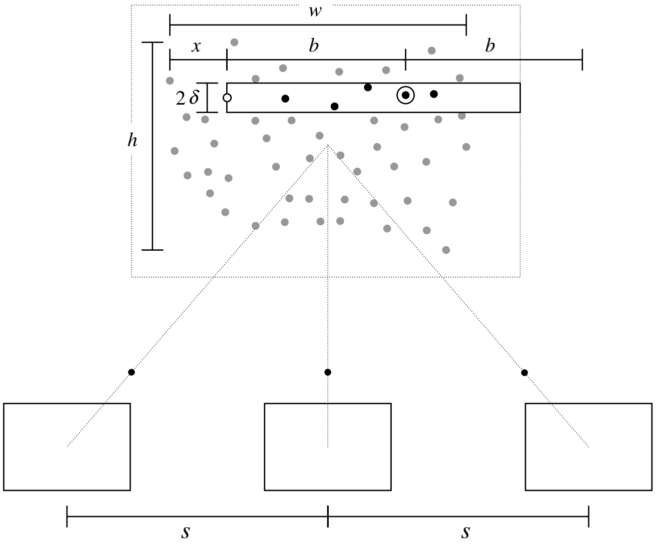


Figure 5. Schematic of the matching process for a three-camera setup. The white point image is that in camera 1 for which the match is being sought; the circled black point image is the current match in camera 2. Since the cameras are separated by identical distances, the distance between matched point images must also be equal. When a separation b drives the search area for the third point image beyond the edge of the domain, there is no match.

As before, the point images are spread randomly over an area A, so that we can expect that within this matching rectangle there are

$$g_{r,pp}(x) = m\frac{a(x)}{A} = \frac{2m\delta\,(w-x)}{wh} \tag{36}$$

ghosts per point image (in the case of a cloud at the reference plane, a point's true match is itself). We then assume that the points are evenly distributed in height, so that there are m/w point images with this x coordinate and thus the total number of ghosts is

$$\begin{aligned} g_{r,2} &= \int_0^w g_{r,pp}(x)\frac{m}{w}\,\mathrm{d}x \\ &= \int_0^w \frac{2m^2\delta}{hw^2}(w-x)\,\mathrm{d}x \\ &= \frac{m^2\delta}{h}. \end{aligned} \tag{37}$$

Once a third camera is added to the setup, the matching criterion becomes much more restrictive. Along with the first two point images lying within some tolerance along some direction, the third point image must now be along some prescribed direction and at a specified distance from camera 2 so that the distance between point images is in proportion to the distance between corresponding cameras (see figure 5). This essentially limits the probability of finding the third point image in the right place to generate a ghost to a mere:

$$P_g = \frac{4\delta^2}{wh}, \tag{38}$$

where we are interpreting the tolerance as designating a square search area rather than a circular one.

Table 3. Comparison of predicted number of random ghosts and measured number of total ghosts using simulated images for two- and three-camera arrangements at a matching tolerance of 1 pixel ($\delta = 1.0$). The point cloud is at the reference plane, and is identical between the two cases of the same density ρ (in point images per 100 pixels squared). For all cases, $w = 433$ and $h = 361$, and only one random point cloud was used in the simulation.

ρ	m	Predicted, $N = 2$	Measured, $N = 2$	Predicted, $N = 3$	Measured $N = 3$
0.48	750	1558	1595	15	33
0.96	1500	6228	6354	120	183
1.44	2250	14 012	14 247	403	551
1.91	3000	24 901	25 122	954	1221
2.87	4500	55 997	56 212	3211	3519
3.82	6000	99 537	99 976	7609	7756

However this probability is only valid within the region A. If we assume for simplicity that the setup has three cameras horizontally arranged (so that the distance between cameras 1 and 2 is the same as that between 2 and 3), then it is easy to see that once the distance between the matching point images in images 1 and 2 exceeds $(w - x)/2$, the probability P_g is 0 because the matching criteria require that the third point image be at a distance greater than $w - x$ from the first. Thus only half of the prospective ghosts could possibly form complete mismatched triplets, so that the expected number of ghosts in a three-camera setup becomes $g_{r,2}/2 \times mP_g$.

We can extend the above argument to an N-camera arrangement to arrive at a rough estimate of the number of random ghosts generated at a given seeding density:

$$\begin{aligned} g_{r,N} &= \frac{g_{r,2}}{N-1}(mP_g)^{N-2} \\ &= \frac{1}{N-1}\left(\frac{m^2\delta}{h}\right)\left(m\frac{4\delta^2}{wh}\right)^{N-2}. \end{aligned} \tag{39}$$

Table 3 shows a comparison of the predicted number random ghosts according to equation (39) and measured number of total ghosts in simulations[4]. The point clouds used are exactly those used in table 2. It should be noted that random and clump ghosts are not distinguishable in the simulation, thus it is expected that the measured number of *total ghosts* be higher than the predicted number of *random ghosts*.

These equations yield only estimates, since they have been derived for a specific camera arrangement and for a sheet at the reference plane to facilitate the calculation of the strip dimensions. In general the widths w and h can be that of the entire image—or whatever limits are imposed by the user—and the arguments should still hold. As the Z thickness of the point cloud increases, clump and random ghost generation becomes more intertwined. In a thick volume, for example, it is possible to have an image in one camera contain a clump of point images within the matching tolerance of each other that are completely dispersed in the other two cameras' images due to different Z coordinates of the points. More ghosts would be generated due to the clumping, but the number would not grow exponentially with the number of cameras in the setup as would be the case for pure clump ghosts.

[4] See the appendix for information regarding the simulations presented here.

It should be noted that in the case of very particular measurement domain shapes, there are camera arrangements that perform better than others in terms of reconstruction quality. For example, if the domain is a thin rectangle oriented toward the long side vertically, then it would be best to have the cameras arranged in a primarily horizontal arrangement, since this would minimize the length of the search strips and thus minimize the probability of finding ghosts.

4.3. Ghosts versus number of cameras

Equation (39) is the estimated number of random ghosts for an N-camera arrangement. The estimated number of clump ghosts is the sum of the number of regions where clumping exists (equation (33)) times the number of ghosts generated in each of these regions (equation (34)) for each case of k point images clumped together. Assuming the probability of finding four clumped images is minute, the estimate is

$$\begin{aligned} g_{c,N} &= \sum_{k=2}^{3} g_c \mu P(k,\lambda) \\ &= m \sum_{k=2}^{3} \left[(k^N - k) \frac{\mathrm{e}^{-mf}(mf)^{k-1}}{k!} \right] \end{aligned} \tag{40}$$

where $f = \frac{4\delta^2}{wh}$ is the ratio of the matching tolerance area to that of the entire point image distribution. Equations (39) and (40) were used to generate figure 6. As expected, the number of random ghosts is easily diminished by adding cameras, which is becoming more and more economically feasible. The number of clump ghosts increases rapidly, but one should recall that, as formulated here, these really exist only in sheet domains, in which case they are easy to remove. In a volumetric domain, there are also ghosts of a 'mixed' type which would arise from the clumping of point images from one viewpoint only. These would appear in the volume as ghosts very close to a real point, but their density would be much lower than a pure clump ghost and thus almost impossible to remove without also altering real, valid data.

5. Conclusion and future work

The reconstruction error analysis presented here has shown some interesting results. The symmetry of the camera arrangement affects the error (equation (20)). Perhaps most surprisingly, a more thorough analysis of the propagation of pixel error (equation (22)) reveals that the camera arrangement (and, for more than three cameras, the camera pairing) affects the error in Z substantially. In the case of camera pairings, the error can be reduced by software simply by choosing the camera pairing that minimizes the average camera sensitivity coefficient $\bar{B}$, which, for all shapes examined, satisfies the condition of equation (27). This has been confirmed via simulations employing exactly identical synthetic images with identical induced error, changing only the camera order so as to compare that of the minimum and maximum sensitivity coefficients possible in a regular octagon arrangement (figure 3).

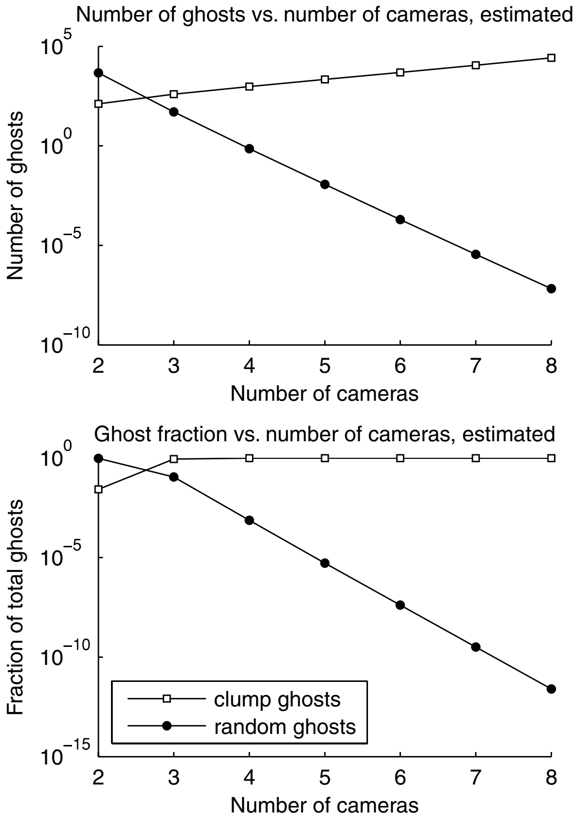


Figure 6. Estimated number of ghosts for a point image density of 0.96 point images per 100 pixels squared and a matching tolerance of 0.75 pixels as a function of number of cameras.

The treatise on reconstruction quality presented here focuses on clump ghosts, which are rampant in thin, sheet-like domains, and random ghosts, which arise in domains of any proportion. The estimates match well with simulations for both clump ghosts (equation (40)) and random ghosts (equation (39)). In the case of random ghosts, the two-camera case also agrees with Maas (1992). Here the argument is extended to be a function of the number of cameras. The analysis of random ghosts presented here also employs a sheet domain but only for simplicity; the arguments should hold for volumetric domains as well. Although the order of magnitude drop in the number of random ghosts between camera arrangements of three and four apertures could be substantial at higher seeding densities, excellent results have been obtained experimentally even with three-camera DDPIV instruments (see Pereira *et al* (2006) and Graff *et al* (2006)).

Together, the two analyses can be used to guide the design of the camera arrangement possible within the constraints of the real optical requirements of a given experiment.

The work presented here does not take into account the performance of the optics, sensors or the point image identification algorithm. Although these are probably best suited to Monte Carlo simulations, it would be of interest to arrive at a comparison of overall performance as a function of sensor resolution versus number of cameras. The analyses presented here are compared with extensive simulations and real measurements taken with the latest DDPIV system built at GALCIT in an upcoming publication.

Acknowledgments

The authors would like to thank the referees, whose helpful comments served to greatly improve this paper.

Appendix. Details of the simulations shown in this paper

The simulations used here as evidence of the analyses were created and processed with the latest version of the DDPIV software from GACLIT (version 4.6.19c). To obtain full control over the error and bypass that generated from the point-identification processing step, the software is enabled to accept lists of point images rather than bitmaps as input. The lists of point images are generated by first generating a pseudo-random three-dimensional point cloud, then pinhole-ray-tracing the points through the chosen camera arrangement to obtain the lists of point images. These lists then represent perfect, aberration and error free locations of the point images in each camera's sensor. Such a list would be the output from the point-identification algorithm that runs on experimental images (except, of course, they would already include error). From there, error can be added by adding a pseudo-randomly generated offset to each point image coordinate. For example, for the results of figure 3, after ray-tracing error was added to the point image locations such that a point image was never shifted outside a square of side length 0.4 pixels centered upon the original point image location.

Inputting 'images' as point image lists has the advantage that there is no question about which original point to compare a reconstructed point. Not only are ghosts immediately identified, but also is the entire commutation from the original point-to-point image set to the reconstructed point. Thus reconstruction error statistics are not tainted by misinterpreted point correspondences.

To generate point sheets (used throughout this paper), the generated point cloud was limited to 1 μm thickness. In the case of figure 3, where this sheet was then traversed through Z 1 mm at a time, the density stated in the units of 'point images per 100 pixels squared' is as measured at the reference plane; it decreases as the cloud approaches the camera.

For all tests done exclusively at the reference plane (those in section 4), only one point cloud was used. In any case where the density is the same, the point cloud is identical (but ray traced through the appropriate camera arrangement). Matching tolerances δ are always treated as the radius of a circular area by the software, but in the analyses this was reduced to a rectangle for ease of notation.

When a particular camera arrangement was used in a simulation, the distance to the reference plane L was set to 551 mm and the focal length f to 28 mm, which makes $M \approx 0.054$. The two- and three-camera cases used in the ghost prediction (table 3) had the cameras 220 mm apart so they match the discussion exactly. The octagonal arrangement used in figure 3 had dimensions such that its arrangement fits in a square 220 by 220 mm. These values were chosen because they are related to one of the recent DDPIV systems built at GALCIT.

Simulations of this sort were used to exhaustively test these analyses and the overall performance of a DDPIV system (including calibration and point-finding algorithms) and will be presented in an upcoming publication.

References

Ariyawansa D D and Clarke T A 1998 High speed multiple view image point correspondences using rectification *Proc. SPIE* **3641** 92–103

Bown M R, MacInnes J M, Allen R W K and Zimmerman W B J 2006 Three-dimensional, three-component velocity measurements using stereoscopic micro-PIV and PTV *Meas. Sci. Technol.* **17** 2175–85

Chen J, Clarke T A, Cooper M A R and Grattan K T V G 1994 An optimised target matching based on a 3-D space intersection and a constrained search for multiple camera views *Proc. SPIE* **2350** 324–35

Clarke R 1946 An application of the poisson distribution *J. Inst. Actuaries* **72** 481

Elsinga G E, Scarano F, Wieneke B and van Oudheusden B W 2006 Tomographic particle image velocimetry *Exp. Fluids* **41** 933–47

Graff E C, Pereira F and Gharib M 2006 Visualization of bubbly propeller flow with defocusing digital particle image velocimetry *12th Int. Symp. on Flow Visualization*

Kanade T, Yoshida A, Oda K, Kano H and Tanaka M 1996 A stereo machine for video-rate dense depth mapping and its new applications *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*

Maas H-G 1992 Complexity analysis for the establishment of image correspondences of dense spatial target fields *Int. Arch. Photogramm. Remote Sens.* **XXIX-B5** 102–7

Maas H G, Gruen A and Papantoniou D A 1993 Particle tracking velocimetry in three-dimensional flows, part 1: photogrammetric determination of particle coordinates *Exp. Fluids* **15** 133–46

Papantoniou D and Maas H G 1990 Recent advances in 3-D particle tracking velocimetry *5th Int. Symp. on the Application of Laser Techniques in Fluid Mechanics*

Pereira F and Gharib M 2002 Defocusing digital particle image velocimetry and the three-dimensional characterization of two-phase flows *Meas. Sci. Technol.* **5** 683–94

Pereira F, Gharib M, Dabiri D and Modarress D 2000 Defocusing digital particle image velocimetry: a 3-component 3-dimensional DPIV measurement technique. Application to bubbly flows *Exp. Fluids* **29** S78–84

Pereira F, Stüer H, Graff E C and Gharib M 2006 Two-frame 3D particle tracking *Meas. Sci. Technol.* **17** 1680–92

Tuma J J and Walsh R A 1997 *Engineering Mathematics Handbook* 4th edn (New York: McGraw-Hill)

Willert C E and Gharib M 1992 Three-dimensional particle imaging with a single camera *Exp. Fluids* **12** 353–8