Experiments in Fluids 26 (1999) 371–380 © Springer-Verlag 1999

# A new tracking algorithm of PIV and removal of spurious vectors using Delaunay tessellation

**X. Song, F. Yamamoto, M. Iguchi, Y. Murai**



**Abstract** A new algorithm of Delaunay Tessellation Particle Tracking Velocimetry (*DT-PTV* in abbreviation) is proposed for tracking particles in images of a PIV system by making use of the Delaunay tessellation (*DT*). The algorithm is tested by using numerically simulated particle images. The calculation results based on DT are compared with those obtained by a conventional algorithm of Binary Image Cross-correlation method (*BICC*). The new algorithm shows higher performance of obtaining more identical particles in two consecutive images correctly with shorter computation time even if the images contain many particles. A further application of *DT* to elimination of spurious vectors is also discussed.

**List of symbols**

| | |
|---|---|
| $A$ | the label of one image |
| $B$ | the label of consecutive image after image $A$ |
| $C$ | cross-correlation coefficient |
| $N$ | the number of triangles in image $A$ |
| $N_P$ | the number of vertices of a polygon |
| $M$ | the number of triangles in image $B$ |
| $Q$ | flux of flow |
| $R$ | radius of an interrogation area |
| $Re$ | Reynolds number |
| $S$ | area of a polygon |
| **Xc** | coordinate vector of gravity center of particle image |
| $f$ | a triangle which is considered as a pattern |
| **r** | displacement vector of the paired particles |
| $tri$ | tessellated triangle set |
| **v** | velocity vector of particle |
| $u$, $v$ | velocity components in the $x$, $y$ directions, respectively |
| $x$, $y$ | coordinates |
| $\Delta t$ | time interval |
| $\rho$ | non-dimensional particle number density |

**Subscript**

| | |
|---|---|
| $i$ | index in image $A$ |
| $j$ | index in image $B$ |

## 1 Introduction

A number of methods and algorithms for particle-image-velocimetry (*PIV*) based on cross-correlation algorithms have been developed. Among these methods, it is common to determine the displacement of ensembles of particle images ("particle image pattern") by cross-correlation methods (Willert and Gharib 1991; Huang et al. 1993a, b) or tracking methods (Gui and Merzkirch 1996). The used "pattern" is the gray scale values of the pixels in a limited area. On the other hand, some researchers (Uemura et al. 1990; Yamamoto et al. 1993, 1996) developed another type of particle-tracking algorithm based on the binary image cross-correlation (*BICC*) in which the "pattern" is extracted from the locations of particles in an interrogation area, i.e., the "pattern" is not the distribution of gray values of pixels but the spatial distribution of particles. Beside the correlation method, another widely used algorithm called Particle Tracking Velocimetry (*PTV*) is based on tracking particles in four time step consecutive images, (Nishino et al. 1989; Kasagi and Nishino 1991; Malik et al. 1993). Wernet (1993, 1995) developed a technique which was based on fuzzy logic. In this method, fuzzy logic technique had been utilized to minimize the error rate in particle tracking. Further application of this method is to estimate individual velocity vector error. These methods process the locations of particles that are obtained from binarization images. Although some information will be lost in the course of binarization, the coordinates of the gravity center of particles can be obtained at a higher spatial accuracy. The merit of using the locations of particles is that it is convenient to extend these methods to the three-dimensional case, because the three-dimensional coordinates of the particles can be calculated from the particle locations in two or more images. Furthermore, these methods are useful for measuring dispersed bubbles in multi-phase flows. In the present research, a new tracking algorithm based on Delaunay Tessellation Particle Tracking Velocimetry (*DT-PTV*) was developed. This *DT-PTV* also use the information on the location of particles in an image. In contrast with the conventional *PTV*, we track a triangle constructed by the locations of three particles after applying Delaunay Tessellation (*DT*). By introducing *DT* into *PIV*, we can devise not only an algorithm for particle tracking but also make the post-processing for *PIV* (i.e. removal of spurious vectors, interpolation) more powerful and effective.

*Received*: *24 November 1997*/*Accepted*: *7 August 1998*

X. Song, F. Yamamoto, Y. Murai
Department of Mechanical Engineering, Fukui University, 3-9-1 Bunkyo, Fukui, 910-8507, Japan

M. Iguchi
Division of Materials Science & Engineering, Graduate School of Engineering of Hokkaido University, Kita-ku, Sapporo, 060-8628, Japan

*Correspondence to*: X. Song

*DT* has had a considerable attention in the literature since 1980s. *DT* finds wide applications in the fields of mesh generation for the finite element analysis method (*FEM*) and an interpolation problem for computer graphics. The main advantage of *DT* is to connect every scattered point efficiently to form elements in either the two-dimensional or three-dimensional space. The elements are triangles in the two-dimensional case while they are triangular pyramids in the three-dimensional case. *DT* is a powerful tool for spatial analysis of scattered data, because the element-like data are easy to be handled, which can be seen from the finite element analysis.

*DT* has two additional advantages: the first is *unique tessellation*, the other is the capability of *constructing reasonable triangles whenever possible*. In general, Delaunay tessellation associated with an arbitrary set of points in the plane is unique except in some special cases. For instance, there are two choices to form the triangles from four points that are located at the vertices of a square. However, these cases are seldom found in practice. The unique tessellation means that no matter where we start to tessellate the scattered data, the tessellated results are the same. In other words, if all particles move within a limited distance, the tessellated triangular grid system will not be changed suddenly. From this viewpoint, we can devise a new algorithm based on *DT*. We can track triangles instead of particles in PIV system, because a triangle has much more information than that of a particle, e.g., the shape, the area, the length of three edges, etc. Another advantage ensures the high accuracy of interpolation. Delaunay triangles are formed under the condition that each triangle will not have too small included angles. This is also very important, for too small included angles will decrease the accuracy of interpolation. Due to these properties, Delaunay triangulation is suitable for analyzing the spatial scattered particles in an image.

The main aim of *PIV* is to extract velocity distribution from the recorded images containing many particles. We can consider these particles as a kind of scattered data. If *DT* is applied to *PIV*, the spatial relations of particles will be easier to be analyzed by generating a triangular grid system. A grid system is very useful, because some *CFD* programs which are based on triangular elements can be utilized directly to calculate other physical information, such as vorticity and pressure, and interpolation becomes more convenient. The triangle grid system can be used further to detect the spurious vectors by checking whether the continuity equation in a triangle is satisfied or not.

There are many papers describing the algorithm of *DT*. Watson (1981) published an algorithm that had the advantage of being particularly simple in a short computation time. The algorithm has a time complexity bound of $O(N^{1.5})$, where $N$ stands for the number of points. Sloan (1984) and Sloan (1987) gave an implementation of Watson's algorithm for computing two-dimensional Delaunay triangulation. An example of applying *DT* into *PIV* can also be found. Bryanston-Cross et al. (1997) calculated the vorticity map by forming *DT* from the measured velocity data.

## 2
## Delaunay tessellation

The main condition to construct the Delaunay triangle mesh is that there is none of nodes within the circumcircle of any

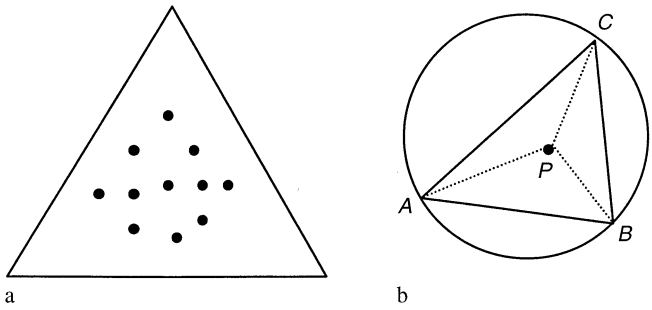


**Fig. 1. a** A *Super-Triangle* completely encompasses all of the points to be triangulated. **b** If a new point is in one of the circumcircles of tessellated triangles, this tessellated triangle is interested by connecting the point $P$ to three vertices $A$, $B$ and $C$

tessellated triangles. We can assembly Delaunay triangulation by introducing each point, one at a time, into an existing Delaunay triangulation.

According to the Watson's algorithm (Watson 1981), a *Super-Triangle* which completely encompasses all of the points to be triangulated is introduced, as shown in Fig. 1a. When a new point is introduced into the triangulation, we can always find an existing triangle which encloses this new point. (The existing triangle is called *Super-Triangle* in the first case of new point.) Three new triangles that connect the new point to each of vertices of the found triangle can be obtained, as shown in Fig. 1b. Any triangle containing the new point within the circumcircle is treated as being intersected. Adding all these triangles which will be intersected together can form a polygon. New triangles that are formed by the new point and each of pairs of vertices on the boundary of the polygon are generated. After dealing all the points to be tessellated, the final triangulation is computed by removing all the triangles that contain one or more of the vertices of the *Super-Triangle*. Figure 2 is an example of the results of *DT*, where the number of particles is 208.

## 3
## DT-PTV

As described in Sect. 1, except in special cases, *DT* is unique and most of triangles have no small induced angles. These two properties are important to apply *DT* to *PIV*.

### 3.1
### Algorithm

The algorithm of *DT-PTV* for two-dimensional *PIV* system is described as follows.

Assume that we have obtained the locations of particles in two successive images after image processing. As shown in Figs. 3–5, two images $A$ and $B$ stand for the images at different time $t$ and $t+\Delta t$, respectively. We can generate a triangular mesh by using *DT* for each image. All tessellated triangles in the image $A$ are denoted as $\{tri_i\}$ $(i=1,\ldots,N)$. We select an arbitrary triangle $tri_i$ in $\{tri_i\}$, and then the coordinates of the center of the triangle are denoted by $x_c$ and $y_c$. The purpose of *DT-PTV* is to find the most possibly paired triangles in an interrogation area of the image $B$ in which the coordinates are $x_c$ and $y_c$ and the radius is $R$. Triangles whose centers are in

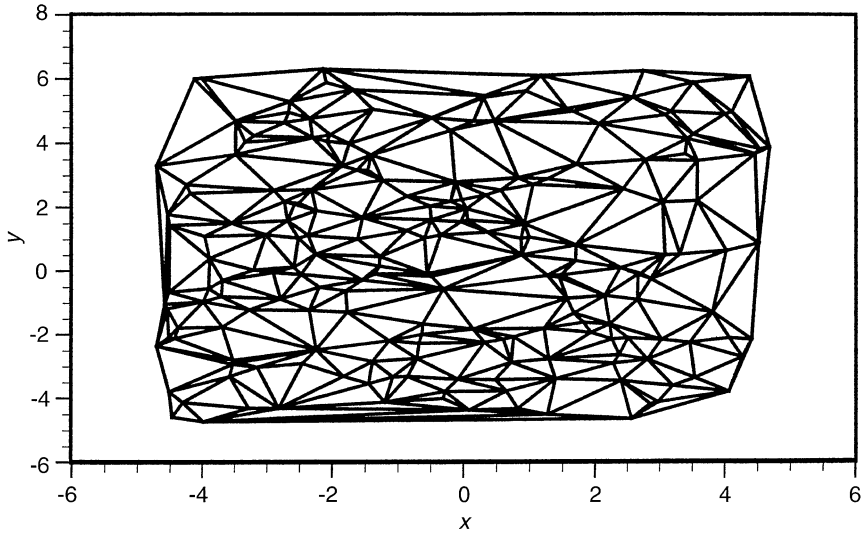


**Fig. 2.** DT of particles. Particles in image are located in the nodes. Number of particles = 208

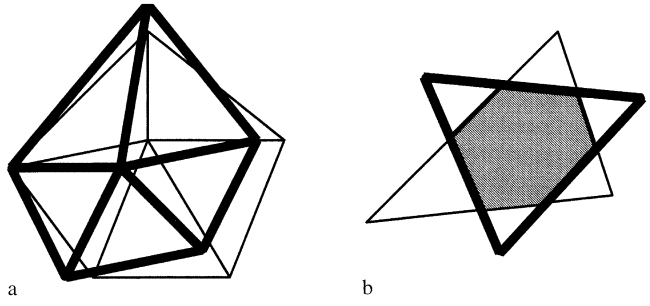


**Fig. 3a, b.** Heavy lines stand for image A, and thin line stands for image B

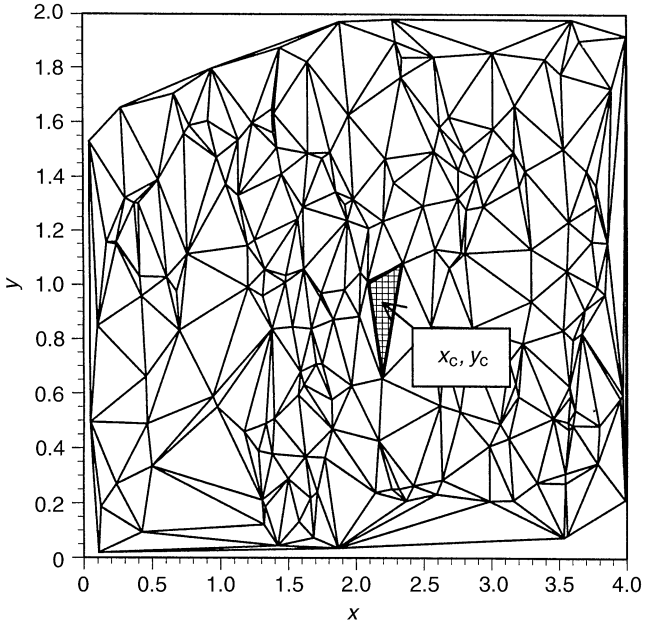


**Fig. 4.** Selection of an arbitrary triangle with center of $x_c$, $y_c$ in image $A$ $(t = t_0)$

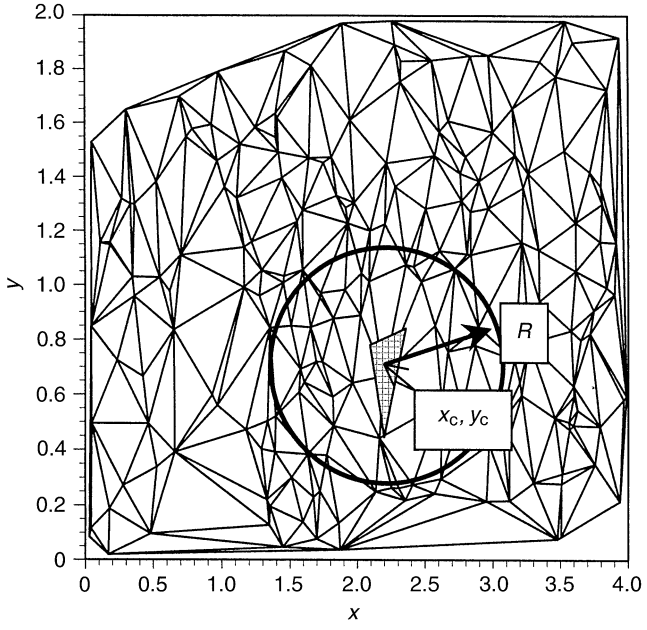


**Fig. 5.** Search triangles within the circle with the center of $x_c$, $y_c$ and radius of $R$ in image $B$ $(t = t_0 + \Delta t)$

the interrogation area are denoted as $\{tri_{ij}\}$ $(j = 1, \ldots, M)$. We check the relation between a triangle $tri_i$ and any other triangles $\{tri_{ij}\}$ by calculating the cross-correlation coefficient of two triangles. Two triangles with a maximum cross-correlation coefficient are regarded as the most possibly paired triangles.

From the definition of the cross-correlation coefficient, we have

$$Cij = \frac{\int\int f_i(x, y)\, f_j(x+p, y+q)\, \mathrm{d}x\, \mathrm{d}y}{\int\int f_i^2\, \mathrm{d}x\, \mathrm{d}y \int\int f_j^2\, \mathrm{d}x\, \mathrm{d}y} \tag{1}$$

where $f_i$ and $f_j$ are triangles in the images $A$ and $B$, $p$ and $q$ are the distances between the gravity centers of two triangles in

$x$ and $y$ directions, respectively. According to the mathematical discussion of cross-correlation by Yamamoto et al. (1996), the cross-correlation coefficient of the two triangles in the images $A$ and $B$ is expressed by

$$Cij = \frac{Area(tri_i \cap tri_{ij})}{\sqrt{Area(tri_i)\,Area(tri_{ij})}} \tag{2}$$

where *Area* stands for the area of each triangle as an interrogation region, $tri_i \cap tri_{ij}$ is the overlapping area as shown in Fig. 3. After the triangles of a pair have been found, three pairs of particles of these triangles can be obtained. The displacement vector of the paired particles $\Delta\mathbf{r}$ is calculated from the distance between their two centers, and the velocity is obtained as $\mathbf{v} = \Delta\mathbf{r}/\Delta t$.

## 3.2 Implementation

### 3.2.1 Calculation of *Cij*

In an earlier paper (Song et al. 1996) we calculated *Cij* from Eq. (2) by overlapping two triangles in two images directly. Some experts of *PIV* pointed out that this calculation might not be applicable to a flow field with strong translation or rotation, because the overlapped area of two triangles would become very small. We tested by numerical simulation and found that when the time interval of two images $\Delta t$ became larger, it was impossible to find two paired triangles correctly, for *Cij* of paired triangles was too small.

In the present paper the calculation of *Cij* was modified to solve the problem. Before calculating *Cij* we transformed the coordinates of two triangles to ensure a good tracking. Let the gravity centers of two triangles move to the origin, and rotate the two triangles so that the vertex of the biggest included angle of each triangle lies in the positive axis of $x$, then calculate the *Cij* of these two transformed triangles. Figure 6 illustrates the procedures. The present method emphasizes the similarity of the form of two triangles. If two triangles are paired triangles, the deformation between them must be limited, and hence the value of *Cij* must make a maximum. The rotation procedure can let us track triangles more correctly even if strong rotation exists in the flow field. For the flow with weak rotation, the rotation procedure will be of no effect.

We need to add a procedure to avoid calculating the value of *Cij* for two obviously non-paired triangles. Denote integers 1, 2 and 3 as the edges of a triangle according to the ascending sequence of edges' lengths, two paired triangles must have the same order of three integers in an counter-clockwise sequence. If there are two triangles with different order of three integers, we can skip calculating *Cij*. Such an apparently small modification of the algorithm can decrease the computation time by 15% ~ 25%.

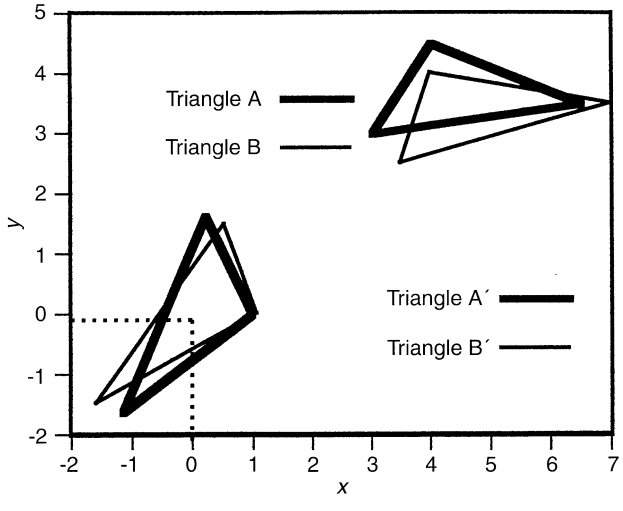


**Fig. 6.** After translation and rotation of the triangles $A$ and $B$, the cross-correlation coefficient of these two transformed triangles $A'$ and $B'$ is calculated

### 3.2.2 Calculation of the overlapping area of two triangles

As seen from Eq. (2), it is important to calculate correctly and efficiently the overlapping area formed by two triangles. This problem is called "*Clipping of Polygons*" as described by Harrington (1987). The algorithm for solving this problem can also be found in many textbooks on computer graphics. The possible number of edges of polygons that are formed by two triangles is 3, 4, 5 and 6.

### 3.2.3 Area of polygon

For a polygon with $N_P$ points of an odd number, its area is given by

$$S = \frac{1}{2}\sum_{k=2}^{N_P}(x_k - x_1)\,(y_{k+1} - y_{k-1}) \tag{3}$$

where $x_k$ and $y_k$ denote the coordinates of the vertex $P_k$. When $N_P$ is an even number, the area is as follows.

$$S = \frac{1}{2}\sum_{k=2}^{NP/2}\{(x_{2k-1} - x_1)\,(y_{2k} - y_{2k-2}) + (x_{2k} - x_2)\,(y_{2k+1} - y_{2k-1})\} \tag{4}$$

### 3.2.4 Image with noise

Noise in image will change the tessellated triangular mesh and therefore cause a wrong result of paired triangles. Fortunately, from a practical point of view, such case will not happen so frequently. According to the principle of Delaunay tessellation, there is no node within the circumcircle of any tessellated triangles, and therefore each particle due to noise should be in a tessellated triangle. A triangle containing the noise particle may not find its paired triangle. However three particles in the triangle can still find their paired particles among the other triangles because they usually belong to more than one triangle. Assume that particle $F$ in Fig. 7 is a noise point for example, $F$ will always exist in the tessellated triangle $ACE$. The triangle $ACE$ fails to find its pair, but point $A$, $E$ and $C$ still can find their pair from the triangles $ABC$ and $CED$. From this example we can conclude that the a few noises will not change the final result of paired particles.

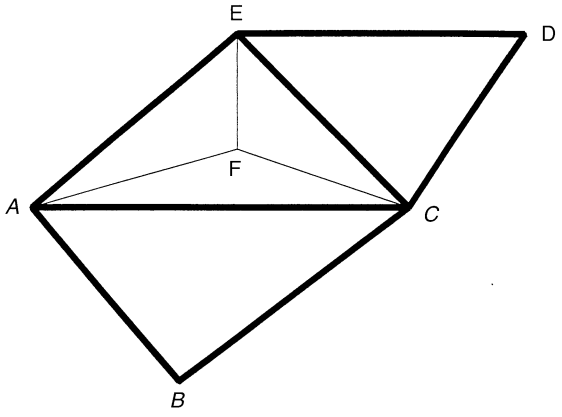


**Fig. 7.** Noise in image. *F* is a false particle

### 3.2.5
### Simple test

In order to verify the performance of the procedure of rotational transformation described in the Sect. 3.2.1, a simple numerical simulation was carried out to test the effect either in the case of solid rotational flow or in the case of translation flow. In both cases, 2000 triangles were generated randomly in the first image, while the movement of each vertex in a triangle between two images is calculated according to the local velocity where the vertex locates. The size of the image is $256 \times 256$ pixel. The unit of rotational speed is degree/interval and the unit of velocities $u$ and $v$ in the translation flow is pixel/interval. The cross-correlation coefficient between each paired triangles in two images is calculated. Both the mean values and the standard deviations of all the cross-correlation coefficients are also calculated (Tables 1 and 2).

Table 1 shows that the cross-correlation coefficients will become smaller as the rotational speed increases. But with the procedure of rotational transformation, we can get the larger coefficient than those without rotational transformation, that means the judgement of the similarity by rotational transformation is more correct than the circulation without rotational transformation. On the other hand, in the case of translation flow, Table 2 shows the rotational transformation will not affect the result, because both the average values and the standard deviations of the cross-correlation coefficients keep the same when we add the rotational transformation.

## 4
## Results and discussion

### 4.1
### Generation of images

An analytical solution of the Navier–Stokes equation was used for testing the performance of the new cross-correlation method. The following expressions for the two velocity components $u$ and $v$ satisfy both the continuity equation and the Navier-Stokes equations.

$$u = \cos(x)\cos(y)\exp(-2t/Re)$$
$$v = \sin(x)\sin(y)\exp(-2t/Re) \tag{5}$$

where $x \in [-3/2\pi,\ 3/2\pi]$, $y \in [-2\pi,\ 2\pi]$.

**Table 1.** Numerical simulations for verifying the performance of rotational transformation in the case of rotational flow field. Angle of rotation stands for the rotational angle between two images, *CCC* stands for cross-correlation coefficients between two triangles, *STD* stands for the standard deviation, *RT* means the procedure of rotational transformation. The unit of angle is degree

| Angle of rotation | Average of *CCC* with *RT* | *STD* of *CCC* with *RT* | Average of *CCC* without *RT* | *STD* of *CCC* without *RT* |
|---|---|---|---|---|
| 0 | 1.000 | 0.00E+00 | 1.000 | 0.00E+00 |
| 10 | 0.985 | 1.64E−05 | 0.680 | 2.09E−01 |
| 20 | 0.944 | 1.49E−05 | 0.554 | 2.23E−01 |
| 30 | 0.886 | 1.54E−05 | 0.490 | 2.09E−01 |
| 40 | 0.820 | 4.40E−06 | 0.435 | 2.04E−01 |
| 50 | 0.753 | 2.00E−05 | 0.400 | 2.02E−01 |
| 60 | 0.691 | 9.58E−06 | 0.377 | 1.88E−01 |
| 70 | 0.633 | 4.51E−05 | 0.346 | 1.80E−01 |
| 80 | 0.582 | 6.09E−06 | 0.339 | 1.70E−01 |
| 90 | 0.537 | 4.96E−06 | 0.324 | 1.62E−01 |

**Table 2.** Numerical simulations for verifying the performance of the procedure of rotational transformation in the case of rotational flow field. $u$ and $v$ are the translational velocities in $x$ and $y$ directions, respectively. The means of *CCC*, *STD* and *RT* are described in Table 1. The unit of $u$ and $v$ is pixel/interval

| $u$ | $v$ | Average of *CCC* with *RT* | *STD* of *CCC* with *RT* | Average of *CCC* without *RT* | *STD* of *CCC* without *RT* |
|---|---|---|---|---|---|
| 2 | 2 | 1 | 1.23E−03 | 1 | 3.22E−05 |
| 4 | 4 | 1 | 9.07E−05 | 1 | 6.00E−05 |
| 6 | 6 | 1 | 8.81E−05 | 1 | 5.68E−05 |
| 8 | 8 | 1 | 6.71E−04 | 1 | 1.07E−04 |
| 10 | 10 | 1 | 2.45E−04 | 1 | 7.74E−05 |
| 12 | 12 | 1 | 2.58E−04 | 1 | 1.01E−04 |
| 14 | 14 | 1 | 1.62E−03 | 1 | 3.20E−04 |
| 16 | 16 | 1 | 3.89E−04 | 1 | 2.39E−04 |
| 18 | 18 | 1 | 4.84E−03 | 1 | 4.80E−03 |
| 20 | 20 | 1 | 5.82E−04 | 1 | 3.46E−04 |

The resulting flow pattern is shown in Fig. 8 at $Re = 1000$. In the present study particles with the coordinates $(x, y)$ were located randomly. The coordinates $(x, y)$ were calculated with adding a $\pm 1\%$ error randomly to simulate a realistic image. Furthermore $\pm 3\%$ of the particles were removed or added randomly in the second image because several particles often disappear when photographing a real image. The velocity components $u$ and $v$ were calculated according to Eq. (5) at time $t$, while $x_1$ and $y_1$ were calculated at the consecutive $t + \Delta t$ by

$$x_1 = x + u \cdot \Delta t$$
$$y_1 = y + v \cdot \Delta t \tag{6}$$

Particles are convolved with the kernel matrix having a Gaussian intensity distribution which has the following form (Willert and Gharib 1991):

$$I(X) = I_0 \exp\left(-\frac{|\mathbf{X} - \mathbf{X}_c|}{2\sigma^2}\right) \tag{7}$$

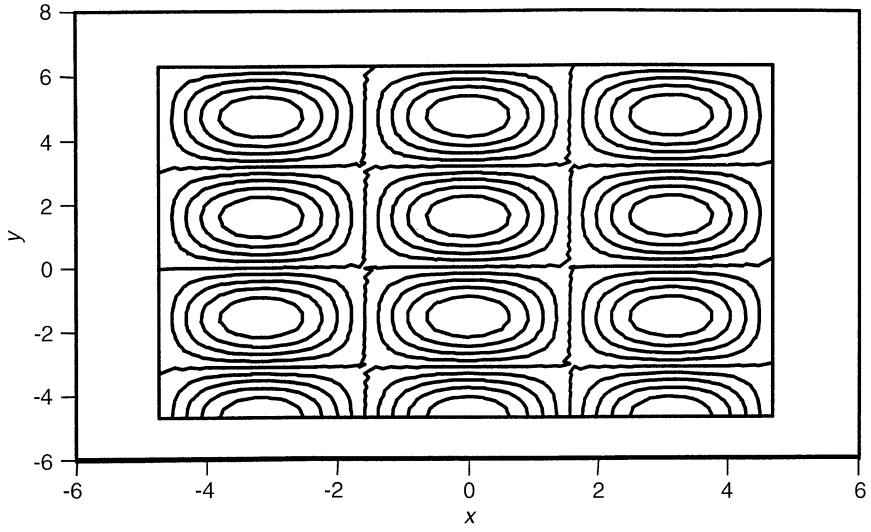


**Fig. 8.** Flow pattern for a simulation at $Re = 1000$ and $t = 0$

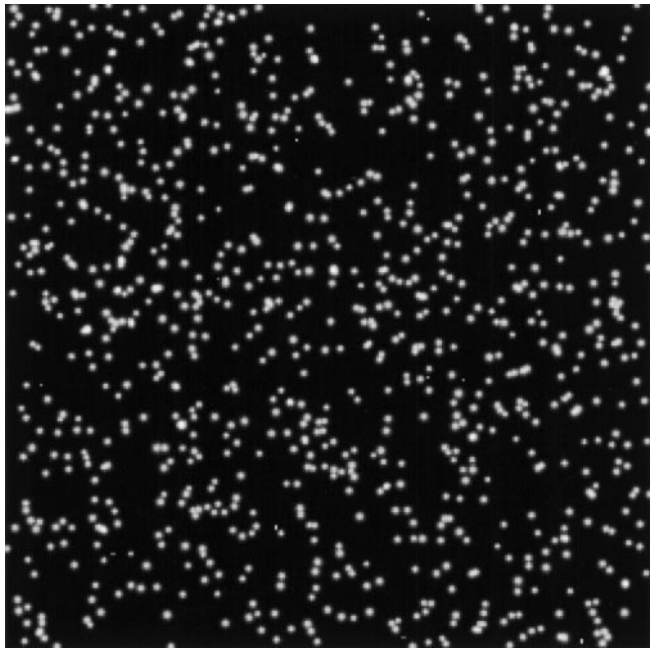

**Fig. 9.** Generated image at time $t$, $I_0 = 240$, $\sigma = 68.3\%$, $256 \times 256$ pixels, the number of particle = 977

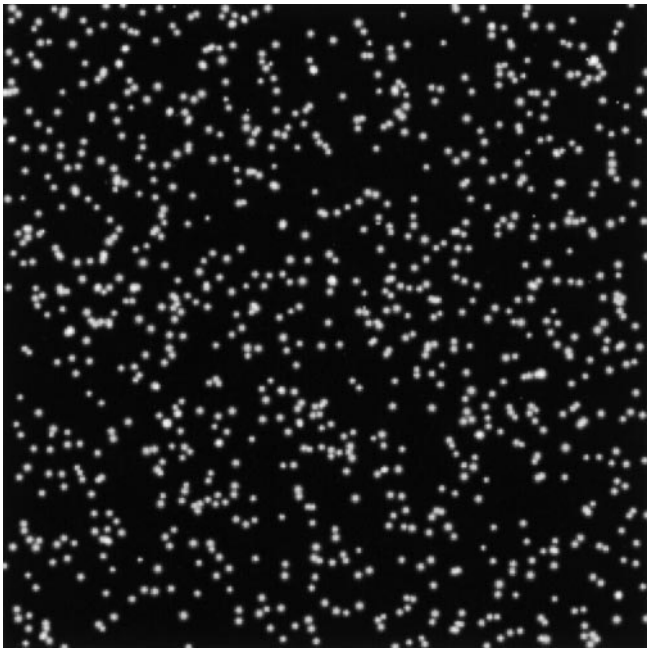

**Fig. 10.** Generated image at time $t + \Delta t$, $I_0 = 240$, $\sigma = 68.3\%$, $256 \times 256$ pixels, the number of particle = 986

where $\mathbf{X}_c$ denotes the gravity center of a particle and $\sigma$ stand for the standard deviation of a Gaussian intensity distribution. The generated images are shown in Figs. 9 and 10.

### 4.2
### Results of the numerical simulation

We can generate a triangular grid system for any distribution of particle images after the procedure of *DT* as described in Chap. 2.

The generated images were processed according to the algorithm of *DT-PTV*. The velocity of each node was obtained. Figs. 11–14 show the results of the numerical simulation. Table 3 lists the computational conditions. The third and fifth rows are the conditions of Figs. 11–14, respectively. Whether all vectors in these figures are spurious or not has been checked. From these figures and Table 3 we can see clearly that when $\Delta t$ is small, there is almost no difference between *DT-PTV* and *BICC*. However, when $\Delta t$ becomes bigger, we can find more correct vectors by *DT-PTV* than by *BICC*. Consider the areas of the vortex centers parts shown in Figs. 13 and 14. The *DT-PTV* is superior to *BICC* because *DT-PTV* finds more vectors in these parts. Figure 13 shows that it is impossible to calculate the vectors by *BICC*, because it causes many spurious vectors.

Table 3 also lists the computation times. The time cost in computation by *DT-PTV* was shorter than that by *BICC*,

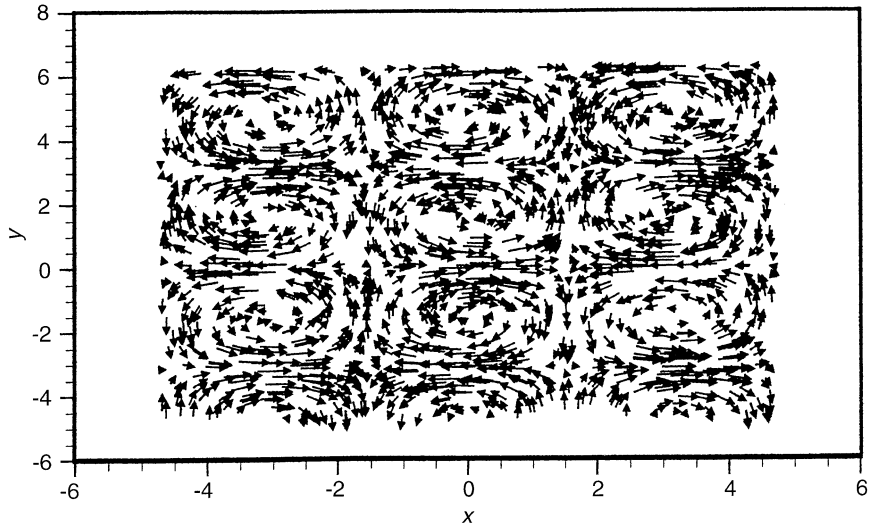


**Fig. 11.** Vector field by BICC method for $\Delta t = 0.1$ and the number of particles is 1505

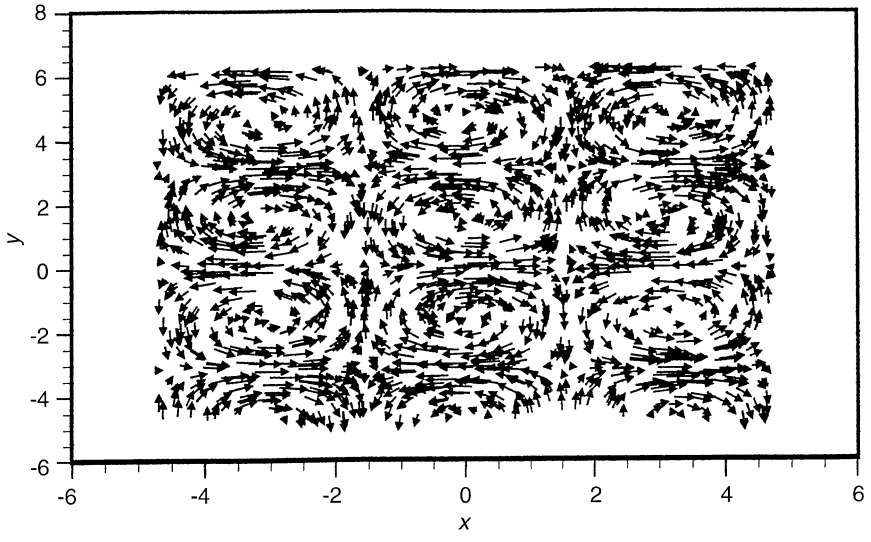


**Fig. 12.** Vector field by DT-PTV method for $\Delta t = 0.1$ and the number of particles is 1505

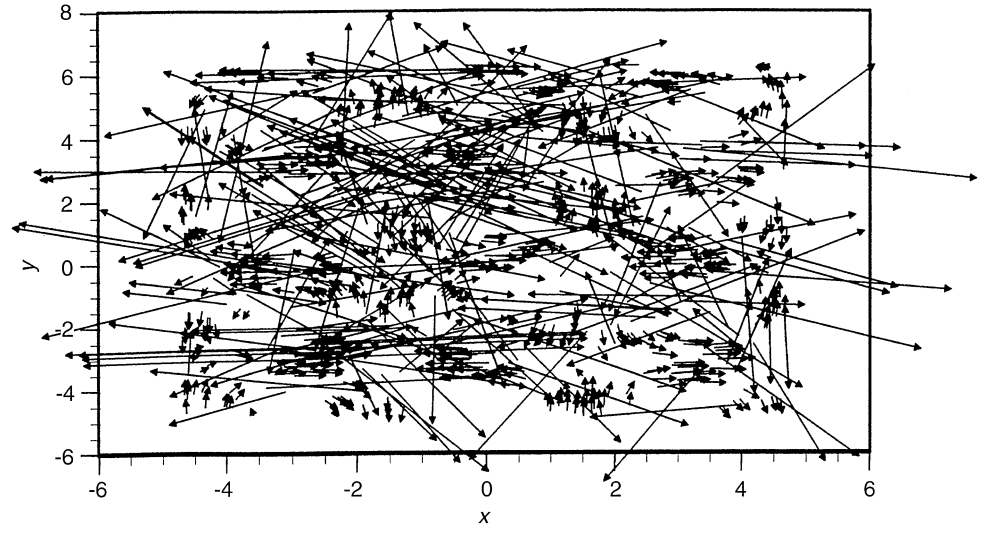


**Fig. 13.** Vector field by BICC method for $\Delta t = 0.3$ and the number of particles is 1500

especially when the number of particles and the radius of interrogation are large. The times of calculating particle centroids from the original images are not included because they are very short. The time for generating a triangular mesh is shorter than 10/100 s when the particle number is about 1000. This can be seen from the algorithms of *DT-PTV* and *BICC*. For *BICC*, it needs a longer time for searching one point and to calculate the distance between two points, and this

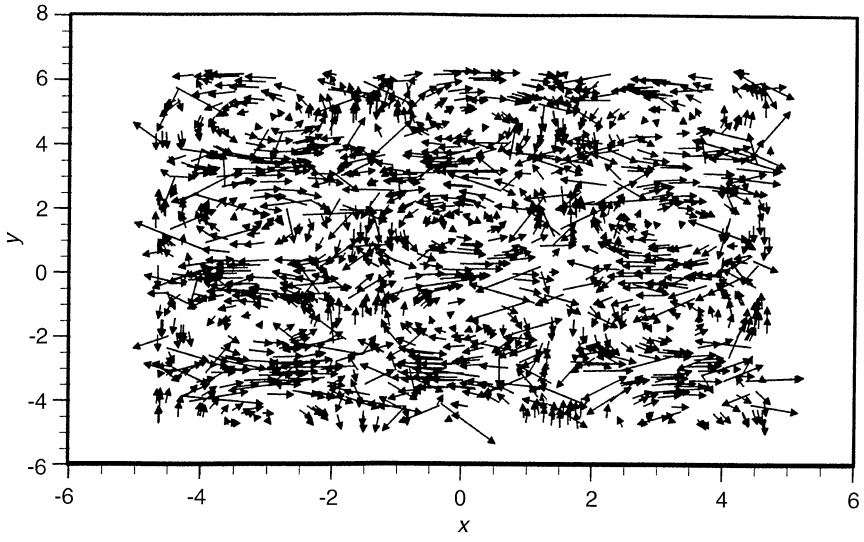


**Fig. 14.** Vector field by DT-PTV method for $\Delta t = 0.3$ and the number of particles is 1505

**Table 3.** Computational results

| Num. | $\Delta t$ | $R$ | *CPU* TIME | | Vectors num. | |
|---|---|---|---|---|---|---|
| | | | BICC | DT-PTV | BICC | DT-PTV |
| 632 | 0.01 | 0.1 | 131 | 6 | 632 | 632 |
| 631 | 0.05 | 0.1 | 130 | 6 | 630 | 631 |
| 1505 | 0.1 | 0.12 | 2480 | 24 | 1500 | 1480 |
| 1498 | 0.2 | 0.22 | 7437 | 26 | 1245 | 1383 |
| 1500 | 0.3 | 0.33 | 14721 | 31 | 910 | 1295 |

The first column is the number of particles. $\Delta t$ means the time interval between two images, whose unit is second. $R$ stands for the radius of interrogation, its unit is centimeter. Computation time is *CPU* times which are measured to the 1/100th s. The computer is CONVEX 3400. The last column lists the number of all vectors found. Whether they are spurious vectors or not has not been checked. The time for generating a triangular mesh is shorter than 10/100 s

procedure will last until the end of computation. However, for *DT-PTV*, after the triangular mesh is generated, it is not necessary to calculate the distance between two points since the geometrical relation between the two points is already known.

### 4.3 Applicability of *DT-PTV*

The applicability of *PIV* system is related to particle number density and moving velocity of particles. Because the flow structure varies a lot in practical situations, and seeding methods and illumination methods are also quite different, it is not easy to evaluate the applicability of each *PIV* system. In the present paper, we introduce a non-dimensional parameter called non-dimensional particle number density to evaluate the applicability. Similar evaluation parameter can also be found in the paper of Baek (1996).

We define newly a non-dimensional particle number density as

$$\rho = r_{\max}\sqrt{\frac{\pi N}{S}} \tag{8}$$

where $N$ is the particle number in an image, $S$ is the area of the image in pixel$^2$ and $r_{\max}$ is the radius of the interrogation area. In fact $\rho$ is the ratio of the maximum distance of the moving particle to the average distance between particles in the image.

We evaluated the performance of *DT-PTV* and found that the higher applicability of *DT-PTV* can process the image when the particle number density is larger than 4. In the two examples mentioned above, $\rho$ is 3.6, 4.2 and 0.8, respectively. It can be said that *DT-PTV* works much better when $\rho$ is about 4. In contrast, *BICC* and 4-frame *PTV* cannot work properly when $\rho$ is larger than 2.

## 5 Elimination of spurious vectors

A velocity field obtained by *PIV* usually includes spurious vectors. These vectors are difficult to be eliminated completely, because they distribute randomly. Many methods have been devised to remove spurious vectors. One of the most popular methods is a *manual* method (Sun et al. 1996). A vector is regarded as being incorrect if it is not within a certain tolerance of both magnitude and direction in comparison with its neighbors. However, making use of this method, we meet difficulties when the flow pattern has a significant deformation, e.g. a flow with strong vortex. In this case we cannot set a limit for the tolerance of the vectors' directions because the directions may vary from 0 to $2\pi$. Furthermore some spurious vectors may have the same order of size compared to their neighbors, and we cannot remove them only by the *manual* method.

Some authors, e.g. Hartmann (1996), Astola (1990) proposed the so-called *Median Filter Method*. Westerweel (1994) used a *Statistical Model* to remove spurious vectors. The present authors tested all these methods. We found that they could remove some of the spurious vectors and that the *Statistical Model* performs much better. The *Median Filter Method* is a smoothing procedure. Some correct vectors will be smoothed by the components of spurious vectors. In fact, the *Median Filter Method* is widely used in the field of reducing picture noises in color television. The *Statistical Model* considers

that the occurrence of spurious vectors satisfies a kind of probability function. As we know, one of the most important properties is that in the flow field, the continuity equation should hold everywhere in many cases. Here we developed a method which can delete such kind of spurious vectors when their neighbors will not satisfy the continuity equation.

Since the vector for each particle in the field must satisfy the continuity equation (Wada 1996), we can detect the spurious vectors by checking the continuity equation in a triangle. If the continuity equation is not satisfied in a triangle, at least one vector in a node will be a spurious vector. We need two steps to find out a spurious vector. The first step is to find out which triangle contains a spurious vector, the next step is to determine node at which the spurious vector is in the triangle. A further step to check the neighboring triangles is needed. Usually it is sufficient if two triangles are checked.

It is convenient to calculate the net flow flux in the triangle or to calculate the integral form of the continuity equation in a triangle. These methods are identical. As shown in Fig. 15, $(x_i, y_i)$ and $(u_i, v_i)$, $i=1, 2, 3$, are the coordinates of the vertices of a triangle and velocities of them. $Q_1$, $Q_2$ and $Q_3$ are the flux at three edges, respectively. Assuming that the density is constant, we have,

$$\int_S \vec{V}\cdot\vec{u}\,\mathrm{d}s = \oint_C \vec{u}\cdot\vec{n}\,\mathrm{d}l = Q_1+Q_2+Q_3 \tag{9}$$

Here $S$ is the area of the triangle and $C$ is the circumference of the triangle. Separating $Q_1$, $Q_2$ and $Q_3$ into two groups according to the sign of the flux, where one is positive, denoted as $Q^+$, and the other is negative $Q^-$. We define

$$E=\frac{Q_1+Q_2+Q_3}{\max(|Q^+|,|Q^-|)} \tag{10}$$

We can use $E$ to determine the spurious vector. From Eq. (10), we can see that $E$ varies from 0 to 1. Usually the magnitude of $E$ of a normal triangle is very small. However, if a triangle contains one spurious vector, it will become considerably bigger than usual. It is sufficient if the threshold of $E=0.5$ is set to determine whether a triangle contains a spurious vector or not. After the triangle is judged to have a spurious vector, we can determine the spurious vector by checking the other triangles containing this node, since one node usually belongs to several triangles.

Figure 16 illustrates how to remove the spurious vector by checking four nodes and four triangles. Table 4 shows the

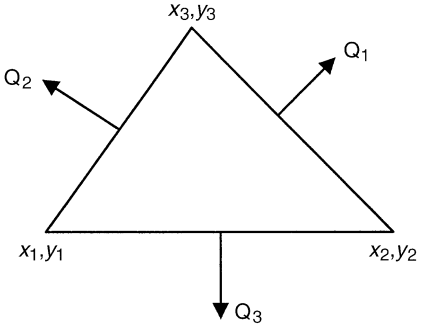


**Fig. 15.** Flow flux at three edges. The integral form of continuity equation in a triangle is as $Q_1+Q_2+Q_3=0$

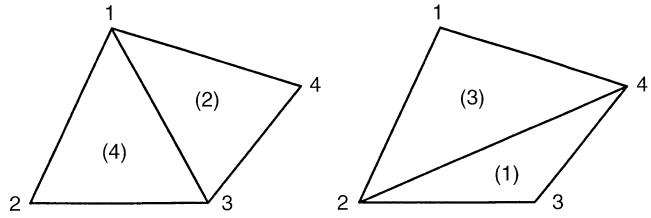


**Fig. 16.** Removal of the spurious vectors by checking flow flux at four triangles



**Table 4.** Triangles and their vertices

| Triangle | Node number | Triangle | Node number |
|---|---|---|---|
| (1) | 2, 3, 4 | (3) | 1, 2, 4 |
| (2) | 1, 3, 4 | (4) | 1, 2, 3 |

triangles and their vertices. Assuming that a vector on one node, for example node 3, is a spurious vector, $E_{(1)}$, $E_{(2)}$ and $E_{(4)}$ are larger than 0.5, $E_{(3)}$ is less than 0.5.

We can detect spurious vectors readily and effectively by the method described above. Sometimes it is not sufficient if only two triangles are checked to determine a spurious vector. For instance, if there are two spurious vectors in one triangle, we need to check more than two triangles for determining these spurious vectors. However these cases happen seldom. Figure 17 illustrates the result of deleting spurious vectors for the original distribution of vectors shown in Fig. 14.

## 6 Conclusion

A new algorithm of *DT-PTV* has been proposed in this study. It is similar to the Particle-Tracking-Velocimetry (*PTV*) algorithm because we track each triangle instead of each particle. It can also be regarded as an extended method of *BICC* since the patterns are constructed by the spatial distributions of particles. There are many types of algorithms for *PIV* based on cross-correlation methods. In these methods usually the cross-correlation coefficients of two particle distribution patterns are calculated. Particle identification is established with a pattern matching technique. However, *DT-PTV* focuses on the degree of similarity of spatial distributions of particles. The information of gray scale values of the particle is not used but the coordinates of particles. The present method can save computing time and memory capacity significantly. The calculation time is considerably shorter compared with other *PIV* algorithms. *DT* is a powerful tool for post-processing of *PIV*. Interpolation for velocity distribution leads to higher accuracy because Delaunay triangles with small included angles are reasonable triangles. The spurious vectors can be detected correctly by checking the flow rate in a triangle based on the equation of continuity.

The performance of *DT-PTV* mainly depends on particle distribution in an image. When the non-dimensional particle number density is less than one, there is no difference between *BICC* and *DT-PTV*. *DT-PTV* does not work significantly better than the *BICC* or other *PIV* techniques on less complex flow

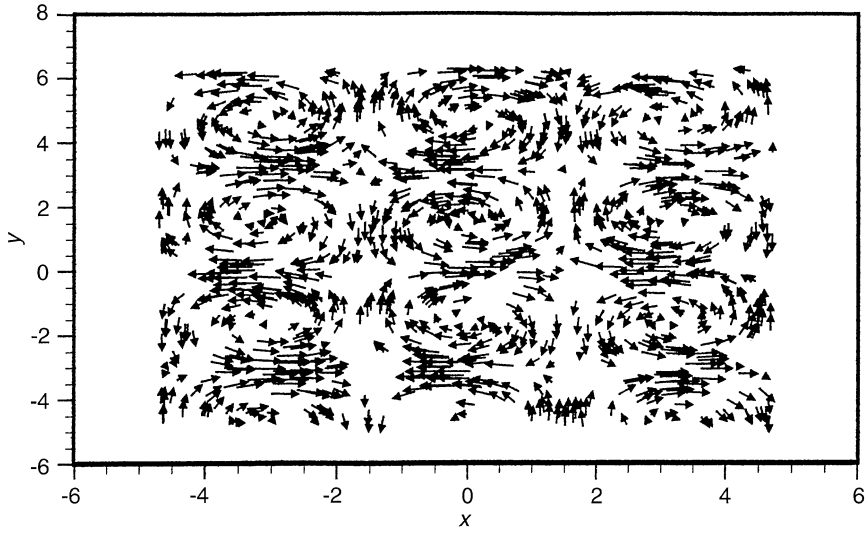


**Fig. 17.** Vector field after removing the spurious vectors in the case of DT-PTV method in Fig. 14 for $\Delta t = 0.3$ and the number of particles is 1505. 1295 vectors are obtained. After removing the spurious vectors, the number of vectors is 1077

fields. But in the case of higher particle number density (usually larger than 2), *BICC* cannot work well. *DT-PTV* is suitable for tracking particles with higher density rather than that with lower density.

The algorithm of *DT* used in the present study is applicable to a multi-dimensional space. In three-dimensions the element becomes a triangular pyramid. The present study can be extended to the three-dimensional case.

## References

**Astola J; Haavisto P; Neuvo Y** (1990) Vector median filters. Proc IEEE 78: 678–689

**Baek SJ; Lee SJ** (1996) A new two-frame particle tracking algorithm using match probability. Exp Fluids 22: 23–32

**Bryanston-Cross P; Udrea DD; Guenette G; Epstein A; d'Hoop EM** (1997) Whole-field visualization and velocity measurement of an instantaneous transonic turbine flow. Int Congress on Instrumentation in Aerospace Simulation Facilities, 20 September–2 October, pp. 278–286, California

**Gui LC; Merzkirch W** (1996) A method of tracking ensembles of particle images. Exp Fluids 21: 465–468

**Harrington S** (1987) Computer graphics – a programming approach, pp. 190–195, New York: McGraw-Hill Int

**Hartmann J; Kohler J; Stolz W; Flogel H** (1996) Evaluation of unsteady flow fields using cross-correlation in image sequences. Exp Fluids 20: 210–217

**Huang HT; Fiedler HE; Wang JJ** (1993a) Limitation and improvement of PIV. Part I: limitation of conventional techniques due to deformation of particle image patterns. Exp Fluids 15: 263–273

**Huang HT; Fiedler HE; Wang JJ** (1993b) Limitation and improvement of PIV. Part II: particle image distortion, a novel technique. Exp Fluids 15: 263–273

**Kasagi N; Nishino K** (1991) Probing turbulence with three-dimensional particle tracking velocimetry. Exp Thermal Fluid Sci 4: 601–612

**Malik M; Dracos T; Papantoniou D** (1993) Particle tracking velocimetry in three-dimensional flows. Part II: particle tracking. Exp Fluids 15: 279–294

**Nishino N; Kasagi N; Hirata M** (1989) Three-dimensional particle tracking velocimetry based on automated digital image processing. J Fluids Eng 111: 384–391

**Sloan SW; Houlsby GT** (1984) An implementation of Watson's algorithm computing two dimenional Delaunay triangulations. Adv Eng Software 6: 192–197

**Sloan SW** (1987) A fast algorithm for constructing Delaunay triangulations in the plane. Adv in Eng Software 9: 34–55

**Song X; Yamamoto F; Iguchi M; Murai Y** (1996) A new cross-correlation algorithm for PIV based on delaunay tessellation. J Visual Soc Japan 16: 19–22

**Sun JH; Yates D; Winterbone DE** (1996) Measurement of the flow field in a diesel engine combustion chamber after combustion by cross-correlation of high-speed photographs. Exp Fluids 20: 335–345

**Uemura T; Yamamoto F; Koukawa M** (1990) High speed algorithm for particle tracking velocimetry using binary image. J Visual Soc Japan 10: 58–64

**Wada A** (1996) Ph.D. dissertation, Fukui University, Fukui, Japan: 97–98

**Watson DF** (1981) Computing the n dimensional Delaunay Tessellation with application to Voronoi polytopes. Comp J 24: 167–172

**Wernet MP** (1993) Fuzzy logic particle tracking velocimetry. NASA Technical Memorandum 106194

**Wernet MP** (1995) Fuzzy inference enhanced information recovery from digital PIV using cross-correlation combined with particle tracking. NASA Tech Memor 106896

**Westerweel J** (1994) Efficient detection of spurious vectors in particle image velocimetry data. Exp Fluids 16: 236–247

**Willert CE; Gharib M** (1991) Digital particle image velocimetry. Exp Fluids 10: 181–193

**Yamamoto F; Uemura T; Tian H; Ohmi K** (1993) Three-dimensional PTV based on binary cross-correlation method. JSME Int J 36: 279–284

**Yamamoto F; Wada A; Iguchi M; Ishikawa M** (1996) Discussion of the cross-correlation methods for PIV: J Flow Visual Image Process 3: 65–78