Cellular Neural Network Based PTV

Kazuo Ohmi[1], Achyut Sapkota[2]

1: Department of Information Systems Engineering, Osaka Sangyo University, Daito Shi, Osaka , Japan
ohmi@ise.osaka-sandai.ac.jp
2: Department of Information Systems Engineering , Osaka Sangyo University, Daito Shi, Osaka , Japan
sapkota@wind.ise.osaka-sandai.ac.jp

Abstract In the recent years, artificial neural networks and genetic algorithms have been attractive in the field of visualization. Among them neural networks seem particularly attractive where the use of the learning principle facilitates the development of fully automated knowledge based system of measurement. Hopfield neural network based approaches had been carried out by some of the authors for two-dimensional Particle Tracking Velocimetry (PTV) but their application is inconvenient due to inherent demerit of high computation time with the increase in number of particles. So far as three-dimensional measurement technique, where two different stages of particle matching namely: spatio-differential and time-differential pairing are carried out, is concerned, the two dimensional algorithms with slight modification can be used for the second stage while for the first stage the basic methodology is to check the proximity of a particle image and its related epipolar line in either of the two or more camera frames. In the present study, network model based on cellular neutral network, a massive parallel computing paradigm defined in discrete N-dimensional space, has been implemented for time-differential as well as spatio-differential pairing of PTV. Unlike the Hopfield neural network, connections are limited to units in local neighborhood of individual units in cellular neural network. Thus, the point of this study lies on the connection scheme of the neurons, according to which the network is reduced from fully connected to partially connected scheme. The network has been successful to yield better results to both of the cases with significantly low computation time and adequate accuracy.

1. Introduction

In the recent years, artificial neural networks and genetic algorithms have been attractive in the field of flow measurement and thus have been implemented, with great enthusiasm, in the particle pairing process of 2-D or 3-D particle tracking velocimetry (Knaak *et. al.*, 1997; Ohmi *et al.*, 2001; Ohmi & Yoshida, 2002; Ohmi, 2003; Ohmi & Sapkota, 2004; Ohmi, 2004). Among them neural networks seem particularly attractive where the use of the learning principle facilitates the development of fully automated knowledge based system of measurement.

In PTV, to explain summarily, two-dimensional and three-dimensional measurement techniques are in practice. Obviously, particle tracking operations are carried out in different manner for two-dimensional and three-dimensional particle images. In two dimensional (2-D) PTV, the particle images taken at certain time interval are processed. In other words, all the particles in an image (at $t = t_n$, say) are compared to the particles in the another image (at $t = t_{n+1}$, say) to determine physically correspondent particle pairs. In this context, Knaak *et al*. (1997) carried out the implementation of neural network techniques to particle pairing process of 2-D PTV using Hopfield neural network (Hopfield, 1982). But this algorithm has the inherent demerit of long computation time with the increase in number of particles (Ohmi and Sapkota, 2004).

Similarly, 3-D PTV is basically composed of two successive steps of particle matching. The first one is the spatial particle pairing, in which the particles viewed by two (or more than two) stereoscopic cameras with a parallax (different viewing angles) have to be correctly paired at every synchronized time step. This is quite important because the 3-D coordinates of individual particles cannot be computed without being assured of identification of two particle images. The basic conventional methodology of spatial particle pairing is to check the proximity of a particle image

and its related epipolar lines in either of the two camera frames (Nishino *et. al.*, 1989). Conventional methods, accompanied by certain concepts, yielded the better pairing results but the computation time and CPU load are largely increased with number of particles as reported by Ohmi (2003). So far as the second step i.e. time-differential particle pairing is concerned, the algorithms explained in the above case of 2-D PTV can be used with slight modification.

In the present study, the concept of cellular neural network has been implemented for time-differential as well as spatio-differential matching of the particles. In case of time-differntial matching the concept formulated by Knaak *et.al.* has been improved in the sense of connection scheme of neuron where network has been visualized as partially connected scheme rather than fully connected case of Hopfield while for the case of spatio-differntial pairing new object function has been defined.

2. Cellular Neural Network

Cellular neural network (CNN) proposed by Chua and Yang (1988) is a massive parallel computing paradigm defined in discrete N-dimensional spaces. Unlike Hopfield neural network, the connections are limited to units in local neighborhood of individual units in this case. In other words, any cell is connected only to its neighbor units, *i.e.* adjacent units interact directly with each other. Units not in the immediate neighborhood have indirect effect because of the propagation effects of the dynamics in the network. The basic architecture of cellular neural network is shown in Fig. 1

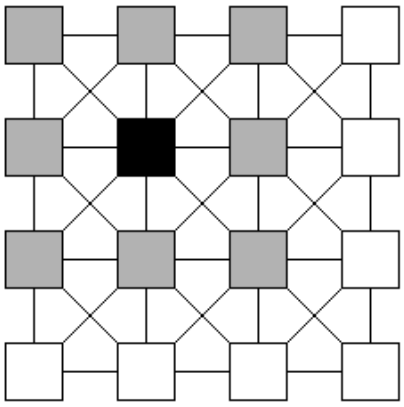


Fig. 1. Cellular neural network architecture

If the unit located in position (j, l) of a two dimensional M*N array is denoted by C_{jl}, and its neighborhood N^r_{ik} is defined by:

$$N^r{}_{ik} = \left\{C_{jl} \mid \max\left\{|j-i|,|l-k|\right\} \le r; 1 \le j \le M, 1 \le l \le N\right\} \quad (1)$$

Where, r, a positive number, is a size of neighborhood.

The basic principle now is that the Lyapunov energy function of the network system should be minimized as formulated below,

$$E = -\frac{1}{2}\sum_{(i,k)}\sum_{\substack{(j,l)\\ C_{jl}\in N^{r}{}_{ik}}} T_{ik,jl}.V_{ik}.V_{jl} - \sum_{(i,k)} I_{ik}V_{ik} \tag{2}$$

Where V_{ik} is the neuron state at the crossing of row i and column k (so is V_{jl} in like manner), $T_{ik,jl}$ is the connection weight of the two neuron units ik and jl, and I_{ik} the energy threshold of unit ik. N and M defined in (1) represents the number of particles in the first and second frames.

The neuron units undergo series of state updates in order to minimize the function, and the initial state and the updated state of neuron units are not computed in a simple threshold scheme (0 or 1) but according to the following normalized function $V_{ik}^{(new)}$

$$V_{ik}{}^{(\mathrm{new})} = \frac{1}{2}\cdot\left(1+\tanh\left(\frac{u_{ik}}{0.5}\right)\right) \tag{3}$$

Where,

$$u_{ik} = \sum_{\substack{j\\ C_{jl}\in N^{r}{}_{ik}}} T_{ik,\,jl}\cdot V_{jl} + I_{ik} \tag{4}$$

The mapping of the particles for particle pairing algorithm is as shown in the Fig. 2.

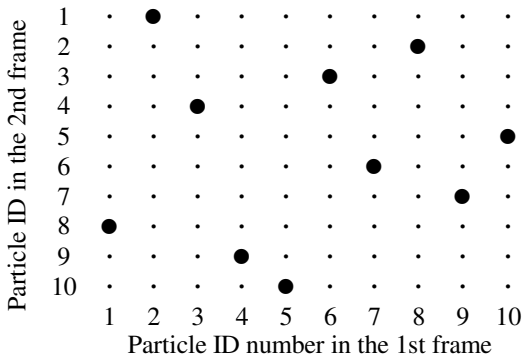


Fig. 2. Neuron units' arrangement for PTV particle pairing problem

The physical state of neuron units is expressed as "active" or "inactive" as shown in Figure. The active neurons (designated by large dots) stand for correct pairing of particle ID numbers and the inactive ones (designated by small dots) for incorrect pairing. As a general rule of this type of particle pairing problem, the particles in the first and second frames have to meet with a unique one-to-one correspondence. So, every row or column of Fig. 2 ought to have at least but no more than one active neuron.

2.1. Application for time-differential particle matching of PTV

The important thing in the implementation of this type of network in PTV is the definition of a physically appropriate object function in the form of constraint conditions. Knaak *et al.* (1997) introduced four different physical constraint conditions for the object function in the case of two dimensional particle images, which are re-formulated as below in accordance with the proposed new scheme.

$$\phi_0 = C_0 \cdot \sum_{(i,k)} f_R(i,k) \cdot V_{ik} \tag{5}$$

$$\phi_1 = C_1 \cdot \sum_{(i,k)} f_X(i,k) \cdot V_{ik} + C_2 \cdot \sum_{(i,k)} f_Y(i,k) \cdot V_{ik} \tag{6}$$

$$\phi_2 = C_3 \cdot \sum_{(i,k)} \sum_{\substack{(j,l) \\ C_{jl} \in N^r{}_{ik}}} \left| g(i,j) - g(k,l) \right| \cdot V_{ik} \cdot V_{jl} \tag{7}$$

$$\phi_3 = C_4 \cdot \sum_{(i,k)} \left(\sum_{\substack{j \\ j \neq i \\ C_{jl} \in N^r{}_{ik}}} V_{ik} \cdot V_{jk} + \sum_{\substack{l \\ l \neq k \\ C_{jl} \in N^r{}_{ik}}} V_{ik} \cdot V_{il} \right) \tag{8}$$

The first condition (5) requires the minimization of the total sum of the distances between all the paired particles, where f_R (i,k) stands for the Euclid distance between the particles i and k in different frames. The second condition (6) asks for the minimization of the sum of local fluctuations of the particle displacement, where f_X(i,k) stands for the x-component of the displacement vector from particle i to particle k and f_Y(i,k) the y-component. The third condition (7) requires the minimization of the sum of local fluctuations of the distances to every neighbor particle, where g(i,j) stands for the Euclid distance between the two neighboring particles i and j in the same frame. The fourth condition (8) requires the uniqueness of solution *i.e.* one particle in one frame will have only one unique partner in the other frame. The five coefficients C_0, C_1, C_2, C_3, C_4 are weight constants by which the weight balance of the four different constraint conditions can be controlled.

Using this new combination of constraint conditions, the final object function is given by

$$\phi = \phi_0 + \phi_1 + \phi_2 + \phi_3 \tag{9}$$

Comparing the second-order term of (9) with that of the Lyapunov function in (2), the connection weight $T_{ik,jl}$ is mathematically determined as below

$$\begin{aligned} T_{ik,\,jl} = & -\frac{C_3}{2} \cdot (1-\delta_{ij}) \cdot (1-\delta_{kl}) \cdot \left| g(i,j) - g(k,l) \right| \\ & -\frac{C_4}{2} \cdot \delta_{ij} \cdot (1-\delta_{kl}) - \frac{C_4}{2} \cdot \delta_{kl} \cdot (1-\delta_{ij}) \end{aligned} \tag{10}$$

And, by comparing the first-order terms, the threshold I_{ik} is determined.

$$I_{ik} = -f_R(i,k) \cdot C_0 - f_X(i,k) \cdot C_1 - f_Y(i,k) \cdot C_2 \tag{11}$$

In (11), δ_{ij} stands for the Kronekker delta and is equal to 1 if $i=j$, or 0 if otherwise. By substituting the values of $T_{ik,jl}$ and I_{ik} to (5) from (10) and (11), the unit output u_{ik} is calculated as

follows:

$$u_{ik} = -\frac{C_3}{2}\left(\sum_{\substack{(j,l) \\ C_{jl}\in N^r{}_{ik}}} \left| g(i,j) - g(k,l) \right| \cdot V_{jl} - \sum_{\substack{l \\ C_{jl}\in N^r{}_{ik}}} g(k,l)\cdot V_{il} - \sum_{\substack{j \\ C_{jl}\in N^r{}_{ik}}} g(i,j)\cdot V_{jk} + V_{ik} \right)$$
$$-\frac{C_4}{2}\left(\sum_{\substack{l \\ C_{jl}\in N^r{}_{ik}}}^{M} V_{il} + \sum_{\substack{j \\ C_{jl}\in N^r{}_{ik}}}^{N} V_{jk} - 2\cdot V_{ik} \right) - C_0 \cdot f_R(i,k) - C_1 \cdot f_X(i,k) - C_2 \cdot f_Y(i,k) \qquad (12)$$

From thus obtained unit output u_{ik}, the updated unit value $V_{ik}^{(new)}$ in (3) is calculated. This series of computation is iterated until the object function ϕ tends to a minimum and the unit values come to their respective optima, which are basically active or inactive. More precisely, since the unit values in (3) are not given in binary digits, the maximal unit value for each row (or column) of Fig. 2 is regarded as "active" and all the others "inactive". The terminal condition of the iterative calculation is that all the "active" neurons remain unvaried over ten successive iterations.

The values of the weight constants C_0 to C_4 are determined on a try-and-error basis. The initial values of V_{ik} and V_{jl} are given random decimal numbers ranging between 0.0 and 1.0. In order to facilitate the determination of the weight constants, the distance parameters in the object functions ϕ_0 to ϕ_3 are also normalized in the range between 0.0 and 1.0.

For the time-differential matching of 3-D PTV, the relation (6) is changed to accommodate the third component (*i.e.* z component) as follows.

$$\phi_1 = C_1 \cdot \sum_{(i,k)} f_X(i,k)\cdot V_{ik} + C_2 \cdot \sum_{(i,k)} f_Y(i,k)\cdot V_{ik}$$
$$+ C_5 \cdot \sum_{(i,k)} f_z(i,k)\cdot V_{ik} \qquad (13)$$

Where the additional component *i.e.* f_z(i,k) is the z-component of the displacement vector from particle i to particle k. C_5 is an additional weight constant. All the other constraint conditions from ϕ_0 to ϕ_3 remain unchanged. The process similar to above is followed afterwards.

2.2. Implementation for spatio-differential particle matching of PTV

Spatio-differential particle matching is the first and quite important stage of 3-D PTV because the 3-D coordinates of individual particles cannot be computed without being assured of identification of two particle images. In this stage particles viewed by two (or more than two) stereoscopic cameras with a parallax (different viewing angles) have to be correctly paired at every synchronized time step. If the parallactic angle between the two cameras is small (say 10 degrees or less), some of the current temporal particle pairing algorithms can be applied to the spatial particle pairing. But, in this case, the resultant measures of particle coordinates are not so much resolved in depth direction as in the two planar directions. For more depth resolution, the parallactic angle has to be larger. The basic methodology of the large-parallax stereoscopic particle pairing is to check the proximity of a particle image and its related epipolar lines in either of the two camera frames.

The mathematical form of this epipolar line can be formulated from the following mapping equations representing a perspective transform:

$$\begin{aligned} &c_{11}x + c_{12}y + c_{13}z + c_{14} - c_{31}xX_1 - c_{32}yX_1 - c_{33}zX_1 = X_1 \\ &c_{21}x + c_{22}y + c_{23}z + c_{24} - c_{31}xY_1 - c_{32}yY_1 - c_{33}zY_1 = Y_1 \\ &d_{11}x + d_{12}y + d_{13}z + d_{14} - d_{31}xX_2 - d_{32}yX_2 - d_{33}zX_2 = X_2 \\ &d_{21}x + d_{22}y + d_{23}z + d_{24} - d_{31}xY_2 - d_{32}yY_2 - d_{33}zY_2 = Y_2 \end{aligned} \tag{14}$$

where x, y and z are the physical-space 3-D coordinates of a particle centroid, X_1 and Y_1 are the 2-D particle coordinates on the left-camera projection screen, and X_2 and Y_2 on the right-camera projection screen. The two sets of matrix coefficients c_{yy} and d_{yy} are the camera parameters for left and right cameras, which are determined by means of calibration using a given number of calibrated target points viewed by the same two cameras. In these equations, if either set of (X_1, Y_1) or (X_2, Y_2) is given, the other set of X and Y come into a linear relation, providing an arithmetic equation of the relevant epipolar line. The normal distance between given particle image and a given epipolar line is calculated from a basic geometric formula.

Having explained so, a new physically appropriate object function has been defined, for spatial particle tracking, in the form of constraint conditions which are formulated as follows,

$$\phi_{0s} = C_{0s} \cdot \sum_{(i,k)} (f_n(i,k) + f_n(k,i)) V_{ik} \tag{15}$$

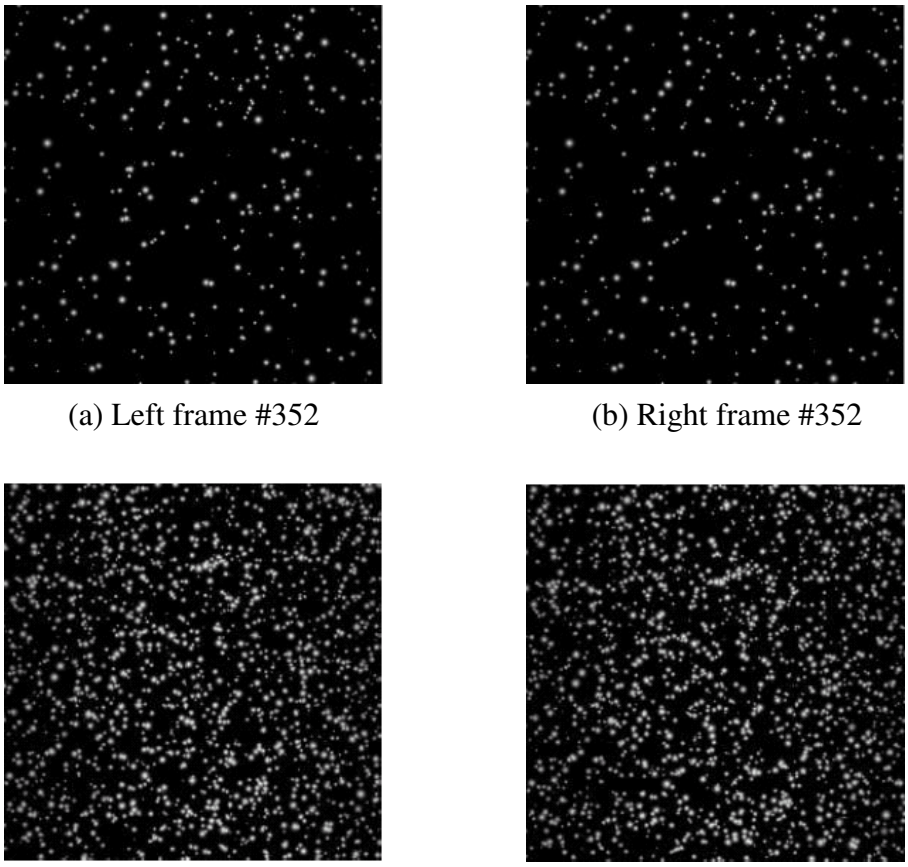
$$\phi_{1s} = C_{1s} \cdot \sum_{(i,k)} \left(\sum_{\substack{j \\ j \neq i}} V_{ik} \cdot V_{jk} + \sum_{\substack{l \\ l \neq k}} V_{ik} \cdot V_{il} \right) \tag{16}$$

The first condition (15) requires the minimization of the sum of normal distance of epipolar line of given particle from its correct pair, where $f_n(i, k)$ stands for the normal distance of epipolar line in right frame (of particle i in left frame) from particle k in the right frame. The second condition (16) asks for particles in the left and right frames to meet unique one to one correspondence. The two coefficients C_{0s} and C_{1s} are weight constants by which the weight balance of the two different constraint conditions can be controlled. The process goes similar to the computation procedure in time-differential matching explained above.

3. Results

The particle matching algorithm is tested by using the 3-D PIV Standard Images (Okamoto *et. al.,* 1997) available from the Visualization Society of Japan. These images are composed of various sets of synthetic time-series particle images generated from DNS (Direct Numerical Simulation) results of a 3-D impinging jet in a square cavity and 6 sets of them are stereoscopic images for with different portions of the flow filed and/or different viewing angles of the cameras. One of the advantages of these standard images is that the data sets come with text files of the original particle coordinates, so that one can compare the analysis results of particle matching with the correct data. In this work, the matching results are presented for image data sets #351 and #352. These two images (#351 and #352) sets represent the same portion of the flow field with same viewing angles of the cameras. The only difference is the number of particles distributed in the view field. The camera viewing angle is inclined to left or right by 30 degrees with respect to the vertical axis,

which is perpendicular to the major observation plane of the flow field. Fig. 3 shows sample frames of above mentioned images series



(a) Left frame #352 (b) Right frame #352

(c) Left frame #351 (d) Right frame #351

Fig 3. Sample frames from the PIV Standard Image

3.1 Spatio-differential matching results

Table 1 and 2 show the spatio-differential particle paring results for #352 and #351 image series. As shown in the table, particle pairing performance is greatly improved with the new algorithm in comparison to conventional method of epipolar line proximity analysis. Even when tested with loss-of-pair particle images also, the results were satisfactory, though conceptually, the technique proposed here deals the perfectly paired data sets. The total computation time for 1509 perfect pairs of #351 image series was only 85 seconds with Celeron 2.5 GHz CPU which takes around half an hour for conventional proximity analysis, this is indeed a great improvement in computation time.

Table 1. Test results for #352 image series

	Frame #	Number of particles	Pairing results	
			Conventional method	With CNN
Perfectly paired particle images	000	277	89.16%	98.19%
	001	277	88.80%	98.19%
Loss-of–pair Particle images	000	327	74.36%	92.41%
	001	326	74.00%	89.53%

Table 2. Test results for #351 image series

	Frame #	Number of particles	Pairing results	
			Conventional method	With CNN
Perfectly paired particle images	000	1509	69.44%	93.63%
	001	1509	71.10%	94.30%
Loss-of–pair particle images	000	1818	67.52%	73.69%
	001	1802	68.12%	73.42%

3.2 Time-differential particle matching results

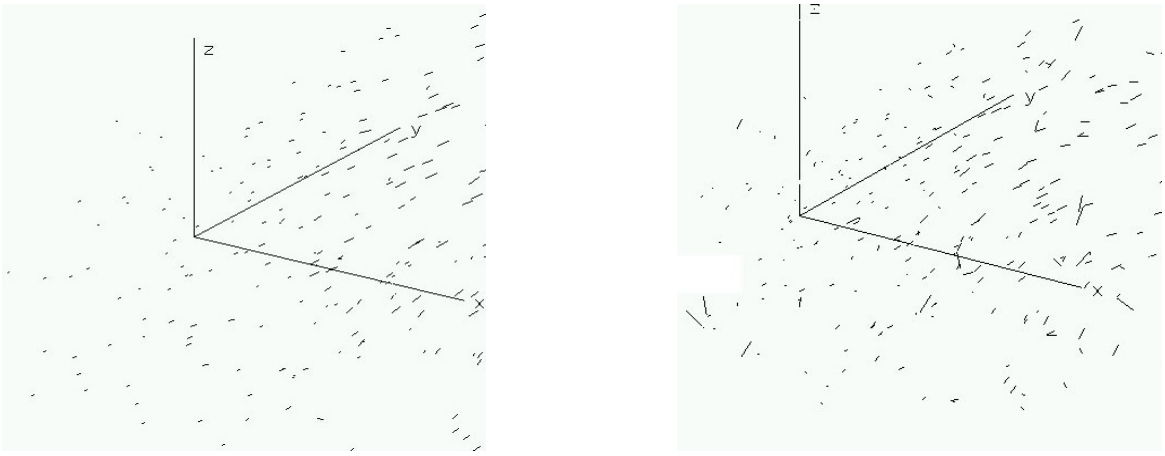
Table 3 and 4 shows the time-differential particle paring results after the computation of 3-D coordinates through the successful matching of the stereoscopic images as shown in Tables 1 and 2. The results have been greatly improved even in this case which is more evident in the case of #351 image series where the accuracy has been improved by more than three times than that of conventional method. But, the algorithm has slightly lost the accuracy with the increase in number of particles especially in the case of loss-of-pair particle images. Sample vectors maps are shown in the Fig.4.

Table 3. Test results for #352 image series

	Frame #	Number of particles	Pairing results	
			Conventional method	With CNN
Perfectly paired particle images	000/ 001	277/ 277	83.03%	97.83%
Loss-of-pair particles images	000/ 001	327/ 326	68.23%	79.42%

Table 4. Test results for #351 image series

	Frame #	Number of particles	Pairing results	
			Conventional method	With CNN
Perfectly paired particle images	000/ 001	1509/ 1509	25.84%	88.53%
Loss-of –pair particles images	000/ 001	1818/ 1802	22.00%	53.54%



(a) Perfectly paired image pairs (b) Loss-of-pair particle images

Fig. 4. Vectors maps obtained from successful matching of the reconstructed 3-D coordinates (Image series #352)

Similary, when tested with the Standard Image 3-D coordinates (the coordinate data which were used to make the standard images) the erroneous pairing is actually zero after optimization of the weight constants C_0 to C_5, with significantly low computation time as shown in Table 5. It shows that the algorithms' performance is comparatively excellent in the case of time-differential matching than in the case of spatio-differential matching. The vector map is shown in Fig 5.

Table 5. Time differential particle pairing using Standard Image 3-D coordinates

No of particles	C0	C1	C2	C3	C4	C5	CPU Time	Iterations
50	100	1	1	100	1	1	0'	14
100	100	1	1	100	1	1	3'	21
150	200	1	1	200	1	1	7'	22
300	1000	100	100	1000	1	100	13'	18
500	3000	300	300	3000	1	300	26'	20
1000	10000	1000	1000	10000	1	1000	1"22'	24

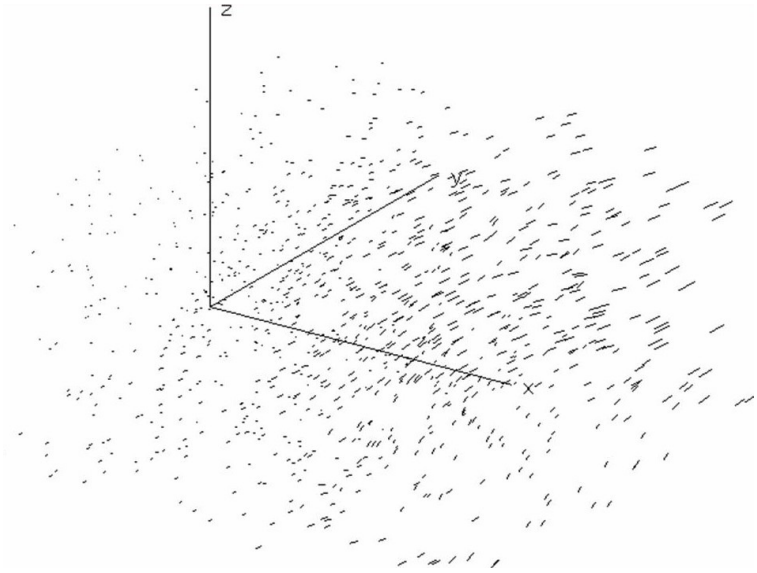


Fig.5 Vectors maps obtained by matching the standard 3-D coordinates
(1000 Particles of #351 Image series)

4. Conclusions

The concept of Cellular neural network has been successfully implemented for particle pairing algorithm of Particle tracking velocimetry with significantly low computation time and adequate accuracy. The significant improvement in computation time has cited a comfortable way for the use of such networks, which was previously seemed tedious due to long computation time. The limitation for the time being lies on dependency of the performance of algorithm on the value of weight constants.

The definition of new object function for spatial particle pairing went well with high degree of accuracy in the perfectly paired data sets but accuracy seems to be degraded with the inclusion of loss-of-pair particle images. This is especially effective for higher number of particle images, while in case of lower number of particle images the degree of accuracy is still more than satisfactory. It seems that special treatment is required for loss-of-pair particle images with higher number of particles per frame. In either of the cases, perfectly paired images and loss-of-pair images, the performance of the algorithm is excellent both in terms of accuracy and computation time in comparison to conventional method of epipolar line proximity analysis. Centering on this fact, the algorithm is the better alternative for conventional methods of spatial particle pairing.

Moreover, since the algorithm has been tested with synthetic time-series particle images generated from DNS (Direct Numerical Simulation), there are still different avenues, to go through, for the algorithm in the series of performance test where practically captured image data are used.

As a matter of fact, particle tracking by itself is typically not capable of successfully tracking particles at the very high seed particle densities, and its alternative in flow visualization i.e. correlation based PIV are not efficient solution for low seed particle density regimes. A hybrid system combining both the techniques is the best solution for the development of high-resolution PIV algorithm. The algorithm proposed for PTV here can be the major component of such system.

References

[1] Chua L.O, Yang L, 1988, "Cellular Neural Networks: Theory and Applications", *IEEE Trans. on Circuits and Systems, (CAS)*, Vol.35, pp.1257-1290.

[2] Hopfield J.J., 1982, "Neural networks and physical systems with emergent collective computational abilities," Proc. *Nat'l Acad. Sci. USA*, Vol.97, pp.2554-2558.

[3] Knaak M., Rothlübbers C., Orglmeister R., 1997, "A Hopfield neural network for flow field computation based on Particle Image Velocimetry/ Particle Tracking Velocimetry image sequences", *IEEE International Conference on Neural Networks,* PP. 48-52

[4] Nishino K., Kasagi N., Hirata M., 1989, "Three-dimensional particle tracking velocimetry based on automated digital image processing," *Trans. ASME, J. Fluids Eng*., Vol.111, pp.384-391.

[5] Ohmi K., Yoshida N., Huynh Thai Hoang, 2001, "Genetic algorithm PIV", *Proc. 4th International Symposium on Particle Image Velocimetry*, P1051.

[6] Ohmi K., Yoshida N., 2002, "3-D Particle Tracking Velocimetry Using a Genetic Algorithm", *Proc. 10th International Symposium on Flow Visualization*, F0323.

[7] Ohmi K., 2003, "3-D Particle Tracking Velocimetry with an improved genetic algorithm", *Proc. 7th International Symposium on Fluid Control, Measurement and Visualization*, #227.

[8] Ohmi K., Sapkota A., 2004, "Particle Tracking Velocimetry using a Hopfield Neural Network", *Proc. 15th International Symposium on Transport Phenomena*, #20.

[9] Ohmi K., Sapkota A., 2004 "Improvement in Hopfield Neural Network PTV," *Proc. International conference on Advanced Optical Diagnostics in Fluids, Solids and Combustion, Tokyo, Japan,*#V0010

[10] Okamoto K., Nishio S., Kobayashi, T., SagaT., 1997, "Standard images for particle imaging velocimetry", *Proc. PIV-Fukui,* pp.229-236.