SELF-ORGANIZED FEATURE MAP OF PARTICLE IMAGE FOR FLOW MEASUREMENT

Yuhai Chen and Allen T. Chwang

Department of Mechanical Engineering
The University of Hong Kong
Hong Kong

ABSTRACT

Self-organized feature map algorithm and the classical particle tracking technique have been adopted together to analyze the single-exposure double-frame particle images for flow measurement. Similar to the normal correlation technique in PIV, the whole region is divided into many small interrogation spots. Instead of applying the correlation algorithm to each of these spots to get their rigid translation, the self-organized feature map algorithm is used to compress the information such that every spot is represented by three coded equivalent particles. After tracking these three particles, a linear distributed velocity function can be obtained at every spot. The spot can contain not only translation, but also rotation, shear and expansion while there is only rigid translation in the spot assumed in the commonly used correlation method. In addition to the theoretical explanation, the suggested method has been verified by a number of digital flow fields which have randomly distributed synthetic particles.

Key words: particle image velocimetry (PIV), particle tracking, self-organized feature map (SOFM), neural network, pattern recognition

1. INTRODUCTION

Particle Image Velocimetry (PIV) is a method for whole field flow velocimetry. It makes fluid velocity measurements by measuring the distance traveled by particles in the time interval between two pulses of light. Each time the laser is pulsed, the images of the particles in the light sheet are obtained. Double pulsing the laser gives two images of every particle in the light sheet. By measuring the distance a particle has moved and dividing that by the time between pulses, the velocity for that particle is determined. This method relies on the tracking of identifiable features in the flow. Difficulty of interpretation may therefore arise in measuring highly turbulent, rotational and reversing flows. Another difficulty is the partner searching, i.e., which two images in time t_1 and t_2, respectively, are a pair of images of the same particle.

Methods such as PIV and laser speckle velocimetry (LSV) inherently do not require the tracking of individual particles. The PIV and LSV techniques rely on the images of double (or multiple) exposed particles within a thin sheet of light in the flow. Although the techniques are essentially the same, LSV uses the particle concentration so dense that individual particles are no longer distinguishable on the image plane, that is, they appear as speckles. A Fourier transformed image of the illuminated region is then obtained by passing the diffracted laser beam through a converging lens and projecting it onto a screen. If the region contains double exposed particle images, a Young's fringe pattern is generated. The spacing and angle of those interference fringes correspond directly to the displacement of the particle images. Measurement of the fringe spacing and angle is either performed directly using image processing techniques or by digitally Fourier transforming the pattern.

In recent years, auto-correlation and cross-correlation have been widely used for digitized images[1], and most of the commercial PIV systems are based on these algorithms. It has been recognized that the single-exposure double-frame PIV system using cross-correlation is superior to both double- and multiple-pulse PIV systems using auto-correlation for a range of velocity fields, for a number of reasons. The versatility in searching the size and location of successive interrogation spots permits a high level of spatial resolution by more efficient use of particles' images. As there is optimally no pair loss if interrogation spots are designed correctly, the spatial resolution can be improved or the seeding density can be reduced, to achieve acceptable detection probabilities which are comparable to auto-correlation methods. By a correlation algorithm, the average displacements of particles inside each interrogation spot can be calculated. The percentage of correct velocity obtained as a function of the number of particle pairs has been detected in every interrogation spot, i.e., the percentage of correct velocity measurements correlates strongly with the number of particle pairs in the spot. As the correlation technique can only calculate the average displacement in the interrogation spot, its size should be small enough for one vector to describe flow of that spot. As mentioned before, the spot cannot be too small in order to contain enough particle image pairs for correlation calculation. For most practical cases, low seeding density may exist which results in the imaging of individual particles. Spurious correlation peaks may occur if correlation analysis is employed in this situation.

It should be noted that the use of video related technology to improve the image acquisition and processing aspects is a trend in PIV studies[2]. For current state of video development, 1024×1024 pixels CCD and the related frame grabber are already regarded as high resolution digital system at a reasonable price level. In comparison with current video technology and the essence of correlation algorithm, the correlation technique is good for the case where the motions of particles in a small interrogation spot are parallel and rigid. However, in a real flow field, other kinds of motion may exist, such as rotation and shear which are not negligible in the spot.

In this study, a method based on the self-organized feature map[4] (SOFM) and particle tracking[3] is introduced. The SOFM is a concept in artificial intelligence and has been successfully used in pattern recognition. Particle tracking, on the other hand, is a classical technique in experimental fluid mechanics. The displacement distribution in each interrogation spot is assumed as a linear function rather than constant which is adopted in correlation techniques. Therefore, not only translation, but also rotation, shear and expansion can be included in each spot. After information compression of every interrogation spot by the Kohonen algorithm[4], three equivalent particles in each spot which have been coded by certain rules can be obtained. In other words, the original particles in the spot have been replaced by three of their equivalents if the particle number is larger than or equal to 3. Even further, these three particles have been coded by certain rule for pair searching. By tracking the movement of these three equivalent particles in two sequential images, a highly accurate displacement distribution in each interrogation spot can be obtained. The calculation seems lighter than the normal correlation algorithm and is also suitable for future real-time velocity survey by parallel processor. In the following chapters, an outline of this method is given. It works well for both numerically simulated images and real gray scale images. Limitations for the current stage of this technique have also been mentioned, but they are more on programming skill than essence of the method itself.

2. MESSAGE COMPRESSION BY SOFM ALGORITHM

The SOFM is one of the most fascinating topic in the neural network field[4,5,6]. It is motivated by a distinct feature of human brain. The brain is organized in many places in such a way that different sensory inputs are represented by topologically ordered computational maps. Consequently, the neurons transform input signals into a plane-coded probability distribution that represents the computed values of parameters by sites of maximum relative activity within the map. Such networks can be used to detect regularities and correlations in input and adopt their feature responses to that input accordingly. Once the algorithm has converged, the feature map computed by this algorithm displays important statistical characteristics of the input. The basic aim of using SOFM algorithm to PIV analysis is to store a large set of input vectors $\mathbf{x}_i \in \mathbf{X}$, ($i=1, 2, \ldots, N$, $N\geq3$), by finding a smaller set of prototypes $\mathbf{w}_j \in \mathbf{W}$, ($j=1, 2, 3$), so as to provide a good approximation to the original input space $\mathbf{X}$. The theoretical basis of the idea is rooted in the vector quantization theory, the motivation for which is the dimension reduction or data compression. In the following paragraphs, the idea of how to derive a new algorithm for PIV analysis is explained. Figure 1a represents a region of flow field which we want to further analyze.

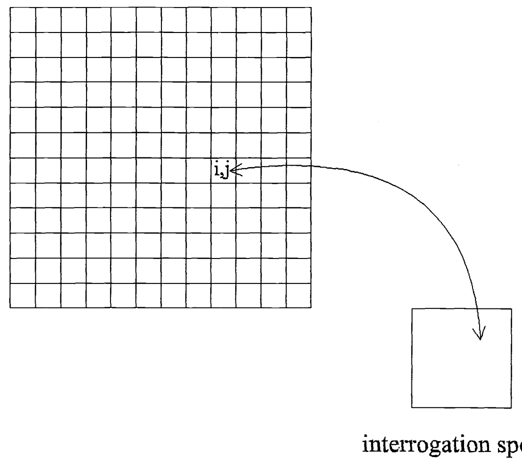

Figure 1a. The flow field under consideration has been divided into many interrogation spots, while spot (i, j) is chosen for calculation

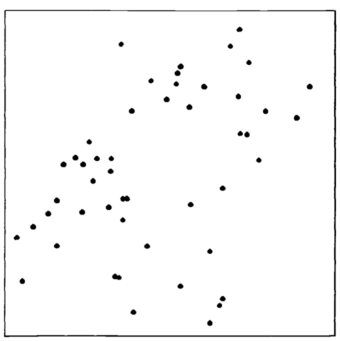

Figure 1b. Enlarged interrogation spot (i, j) with particle images

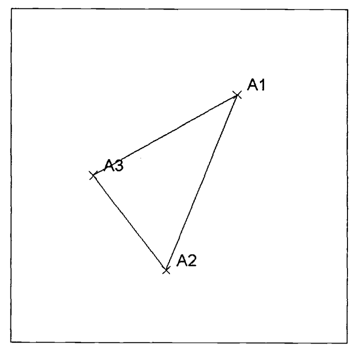

Figure 1c. Equivalent particles A1, A2, A3 calculated by SOFM

Figure 1b describes an enlarged interrogation spot arbitrarily collected from the image of flow field at time $t = t_1$. Figure 1c represents equivalent particles calculated by the SOFM algorithm. A1, A2 and A3 are called synaptic weighted vectors in neural network. Equivalent particles at time $t = t_2$ are also obtained using the same algorithm. The velocity distribution in this spot can be determined by tracking the behavior of these equivalent particles. Now let us go a little further into the theory of SOFM in order to show the procedures of how to find these equivalent particles and also to explain the theory behind the algorithm. Figure 2 shows a two-dimensional lattice of neurons with feed-forward connections and lateral feedback connections.

Each particle in the interrogation spot is regarded as an input vector $\mathbf{x_p}$,

$$x_p = (x_{p1}, x_{p2}) , \qquad (p = 1,2,3,\ldots,N) . \tag{1}$$

The synaptic weighted vector of neuron j is denoted by

$$w_j = (w_{j1}, w_{j2}) , \quad (j = 1,2,3) . \tag{2}$$

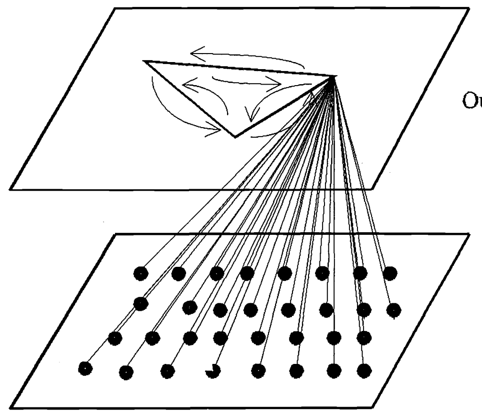

Figure 2. Kohonen's model: two-dimensional lattice of neurons

To find the best match of the input vector $\mathbf{x}_p$ with the synaptic weighted vectors $\mathbf{w}_j$, we simply compare the Euclidean distance between input vectors $\mathbf{x}_p$ and the synaptic weighted vectors $\mathbf{w}_j$. That is the so-called best-matching criterion. Specifically, if index i(**x**) is used to identify the neuron that best matches the input vector $\mathbf{x}_p$, then i(**x**) can be determined by applying the condition

$$i(x) = min\|x_p - w_j\| , \qquad (j = 1,2,3; \quad p = 1,2,3,\ldots,N) . \tag{3}$$

The particular neuron i that satisfies this condition is called the best-matching or winning neuron of the input vector $\mathbf{x}_p$. The response of network towards input $\mathbf{x}_p$ is the synaptic weighted vector $\mathbf{w}_j$. The modified Hebbian hypothesis[6] which includes a nonlinear forgetting term $-g(\mathbf{y}_j)\mathbf{w}_j$ is proved as a guide to make the change, where $\mathbf{w}_j$ is the synaptic weighted vector, $\mathbf{y}_j$ is the network output at neuron j and $g(\mathbf{y}_j)$ is

$$g(y_j) = 0 \quad when \ y_j = 0 , \qquad (j = 1,2,3) . \tag{4}$$

The Hebbian formula can be described as

$$\frac{dw_j}{dt} = \eta y_j x_i - g(y_j)w_j , \qquad (j = 1,2,3; \quad i \ is \ the \ winning \ neuron) , \tag{5}$$

where t denotes the continuous time and η is the learning rate parameter of the algorithm. The network output $\mathbf{y}_j$ is given by

$$\begin{cases} y_j = 1 , \quad (j = i, i \ is \ the \ winning \ neuron) , & (6a) \\ y_j = 0 , \quad (j \neq i) . & (6b) \end{cases}$$

In a corresponding way, the function $g(\mathbf{y}_j)$ is given as

$$\begin{cases} g(y_j) = \alpha , \quad (j = i, \ i \ is \ the \ winning \ neuron) , & (7a) \\ g(y_j) = 0 , \quad (j \neq i) , & (7b) \end{cases}$$

where α is some positive constant. Without loss of generality, we may use the same scaling factor for the input vector $\mathbf{x}_i$ and the weighted $\mathbf{w}_j$. In other words, by equations (4) to (7), we have

$$\begin{cases} \dfrac{dw_j}{dt} = \eta(x_i - w_j)\ , & (j = i,\ i \text{ is the winning neuron})\ , \quad (8a) \\ \dfrac{dw_j}{dt} = 0\ , & (j \neq i)\ . \quad (8b) \end{cases}$$

By using the discrete-time formalism, $\mathbf{w}_j(n)$ represents the synaptic weighted vector of neuron j at discrete time n. The updated value of $\mathbf{w}_j(n+1)$ at time n+1 is computed by

$$\begin{cases} w_j(n+1) = w_j(n) + \eta(n)[x_i - w_j(n)]\ , & (j = i,\ i \text{ is the winning neuron})\ , \quad (9a) \\ w_j(n+1) = w_j(n)\ , & (j \neq i)\ . \quad (9b) \end{cases}$$

The effect of update equations (9a, b) is to move the synaptic weighted vector $\mathbf{w}_i$ of the winning neuron i toward the winning input vector $\mathbf{x}_i$. Upon repeated presentation of the training data, the synaptic weighted vectors tend to follow the distribution of the input vectors due to the neighborhood updating. In this way, the SOFM algorithm can store a large set of input vectors $\mathbf{x}_p$, (p=1, 2,..., N, $N \geq 3$), by finding a small set of $\mathbf{w}_j$, (j=1, 2, 3), so as to provide a good approximation to the original input space $\mathbf{x}_p$, (p=1, 2, 3,...,N, $N\geq3$). For our PIV analysis, the final synaptic weighted vector $\mathbf{w}_j$ is just a representative of the original large number of randomly distributed particle images in each interrogation spot. These $\mathbf{w}_j$, (j=1, 2, 3), are called equivalent particles in order to distinguish them from the real input particles $\mathbf{x}_p$, (p=1, 2, 3, ..., N, $N\geq3$). The displacements of these coded $\mathbf{w}_j$ are relatively easy to handle by the particle tracking technique.

During the PIV analysis, we first divide the whole flow region into many interrogation spots, then particle images $\mathbf{x}_p$, (p=1, 2, 3, ... , N, $N\geq3$), in each spot are processed by the SOFM algorithm to find their equivalent particles $\mathbf{w}_j$, (j=1, 2, 3). It simply means that the message of many input particles is compressed into that of only 3 equivalent particles. After getting the velocities of $\mathbf{w}_j$, the velocity distribution in each interrogation spot can be determined by linear interpolation.

Before we show the capability of this method for the PIV analysis, we would like to address to some questions about the SOFM algorithm which may be encountered during its application in the PIV image processing.

(1) The choice of learning rate parameter

The learning rate parameter $\eta(n)$ should be time-varying. During the initial phase of the algorithm, it should begin with a value close to 1, thereafter $\eta(n)$ should decrease gradually, but staying above 0.1. For good statistical accuracy, $\eta(n)$ should be maintained during the convergence phase at a small value on the order of 0.01 or less. It is chosen as

$$\begin{cases} \eta(n) = \exp(\dfrac{-n}{t_m})\ , & (n < 2.3t_m)\ , \quad (10a) \\ \eta(n) = 0.001\ , & (n \geq 2.3t_m)\ , \quad (10b) \end{cases}$$

t_m is a given time constant to the network.

(2) Help "unfortunate" neurons

In the SOFM algorithm, the weighted vector $\mathbf{w}_j$ is updated only for winning neurons, i.e., with index i which is selected by equation (3). Some neuron weighted vectors may start out far from some input vectors and, accordingly, they will never win the competition no matter how long the procedure continues. The result is that those vectors will have nothing to do with the weighted vectors and have no contribution to the message compression procedure. To stop this from happening, biases are used to give neurons, which are only winning the competition rarely, an advantage over neurons which are winning often. We can use equation (11) instead of equation (3).

$$i(x) = min(\|x_p - w_j\| - b) \ , \quad (p = 1,2,3,\ldots,N; \ j = 1,2,3) \ , \tag{11}$$

b is the bias and it is a positive value.

We can choose an adaptive form of b which changes its own value according to the competition situation. The form is suggested as[7]

$$\begin{cases} z(n+1) = Cz(n) + (1-C)y_j(n) \ , & (12a) \\ b = exp(1 - log\, z) \ , & (12b) \end{cases}$$

where C is a positive value slightly less than 1, $\mathbf{y}_j$ is the output from the network, $\mathbf{y}_j = 1$ when j wins the competition, otherwise $\mathbf{y}_j = 0$. From equations (12a, b), it can be shown that if a neuron never wins a competition, its bias will eventually become large enough so that it will be able to win. Once the neuron's weights have moved into a group of input vectors and the neuron is winning consistently, its bias will decrease to zero. Thus the problem is wisely solved.

(3) Convergence law

There is still no law that can ensure the computation is converging. The accuracy of $\mathbf{w}_j$ depends on the parameters of the algorithm, namely, η, C and t_m. Let's define L(n) by

$$L(n) = \sum_{p=1}^{N} \sum_{j=1}^{3} \|w_j(n) - x_p\| \ , \tag{13}$$

then we may introduce another criterion to describe the convergence of calculation

$$\frac{L(n+1) - L(n)}{L(n)} \leq \varepsilon \ , \tag{14}$$

where ε is an acceptable tolerance.

Either equation (14) is satisfied or the assumed maximum number of input presentation is reached, the iteration will end. These values of $\mathbf{w}_j$, (j=1, 2, 3), represent the equivalent particles of the input vectors $\mathbf{x}_p$, (p = 1, 2, 3, ..., N). In this way, we use three vectors $\mathbf{w}_j$ to represent essential characteristics of N input vectors, $(N \geq 3)$, in every interrogation spot.

3. NEW TRACKING ALGORITHM FOR FLOW ANALYSIS

By the SOFM algorithm described above, the inputs $\mathbf{x}_p$, (p = 1, 2, 3,..., N, $N \geq 3$), which are the original captured images of N particles, have been compressed as three of their equivalent particles $\mathbf{w}_j$, (j = 1, 2, 3), in each spot. After tracking the behavior of these three particles at time t_1 and t_2, the movement of this interrogation spot can be obtained. As there are only three equivalent particles in each spot, it is easy to identify their partners. The equivalent particles can be connected to a unique triangle as shown in Figure 1c. The number of each vertex is defined by given rules, such as the vertex of the smallest angle is defined as particle number 1, the medium angle corresponds to particle number 2 while the largest angle corresponds to particle number 3. We have to ensure that the time period between t_1 and t_2 is short enough to keep this shape relationship unchanged, and this assumption can easily be achieved in normal PIV experiments.

In the present method, the velocity in each interrogation spot is linearly distributed, rather than rigid motion assumed in the correlation method. In each spot, not only translation, but rotation, shear and expansion can be obtained. The linear interpolation is expressed by

$$\begin{cases} u_{j1} = u_{01} + a w_{j1} + b w_{j2}\ , \quad (j = 1,2,3)\ , & (15a) \\ u_{j2} = u_{02} + c w_{j1} + d w_{j2}\ , & (15b) \end{cases}$$

where $u_j = (u_{j1}, u_{j2})$ is the velocity of equivalent particle w_j, and it is simply calculated by

$$u_j = \frac{w_j(t_2) - w_j(t_1)}{t_2 - t_1}\ , \quad (j = 1,2,3)\ , \tag{16}$$

The unknowns in equation (15), i.e., u_{01}, u_{02}, a, b, c and d, can be calculated by

$$\begin{bmatrix} 1 & 0 & w_{11} & w_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 & w_{11} & w_{12} \\ 1 & 0 & w_{21} & w_{22} & 0 & 0 \\ 0 & 1 & 0 & 0 & w_{21} & w_{22} \\ 1 & 0 & w_{31} & w_{32} & 0 & 0 \\ 0 & 1 & 0 & 0 & w_{31} & w_{32} \end{bmatrix} \begin{Bmatrix} u_{01} \\ u_{02} \\ a \\ b \\ c \\ d \end{Bmatrix} = \begin{Bmatrix} u_{11} \\ u_{12} \\ u_{21} \\ u_{22} \\ u_{31} \\ u_{32} \end{Bmatrix}. \tag{17}$$

During experiments, two single-exposure images are taken. A consistent threshold is simply used to process them into binary images. The value at each pixel whose light index beyond the threshold will be assigned as white (255th gray scale) while others will be assigned as black (0th gray scale). The white pixels are regarded as images of particles, i.e., the inputs to the SOFM algorithm, while the black pixels are, on the other hand, as background. The number of inputs (light pixels) can be changed if the threshold value is altered. In this way, the calculation work can be reduced, while the accuracy is maintained by a correct choice of the threshold.

As an alternative, the original image can be used directly without the binary pre-processing. If I_p is defined as the value of gray scale of pixel number p. The competition procedure of equation (3) can be re-written in weighted form as

$$i(x) = min\left\| \frac{255}{I_p + 1}(x_p - w_j) \right\|, \quad (p = 1,2,3,\ldots,N;\ j = 1,2,3;\ 0 \le I_p \le 255)\ . \tag{18}$$

The physical meaning behind equation (18) is that a light pixel is easier to be selected than its dark counterpart since the value of I_p is large for a light pixel. For dark pixels (with small value of I_p), on the other hand, their distances cannot get closer because the existing factor, $\frac{255}{I_p + 1}$, enlarges the distances. Dark pixels will never win the competition, so they will be automatically filtered out in this way.

4. VERIFICATION BY NUMERICAL SIMULATION

In order to verify this method, some standard two-dimensional flows which contain randomly distributed synthetic particles have been constructed. Simple cases, such as constant translation and rigid rotation will be discussed first to show the basic steps of our method without mixing with the complication of the flow itself. Figure 1a represents a region of flow field to be further analyzed. Figure 1b shows the randomly distributed synthetic particle images at interrogation spot (i, j). Figure 1c is obtained by using the SOFM algorithm. A1, A2 and A3 are the so-called equivalent particles.

The flow undertakes a small amount of translation within time period $t_2 - t_1$. Figure 3a shows synthetic particles at t_2 and the equivalent particles B1, B2, and B3, together with their counterparts at t_1. With the method mentioned in section 2, (A1, B1), (A2, B2), (A3, B3) are coded as partners. Flow characteristics in this small interrogation spot (i, j) can be obtained by tracking these three pairs. The velocities of particles in this spot can be interpolated by equation (15). Figure 3b shows velocity vectors of every particle in the spot.

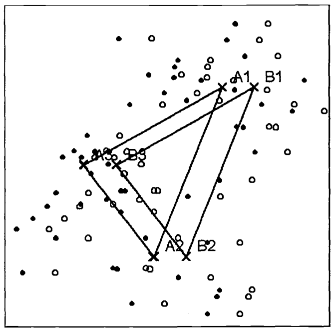

Figure 3a. Particle images at t_1 (solid) and at t_2 (hollow)

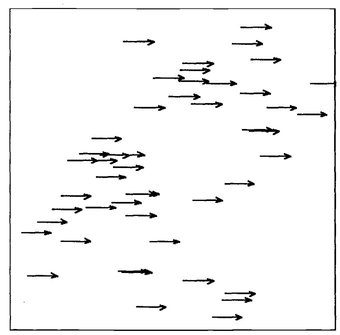

Figure 3b. Velocity vectors

The second example is designed to test its capability to deal with rigid rotation of any interrogation spot (i, j). It is assumed that there is a 15° counter-clockwise rotation around left lower corner of spot (i, j). Figure 4a shows its important feature map obtained by the SOFM algorithm, while Figure 4b shows particle velocity vectors of the same spot by particle tracking and then interpolating.

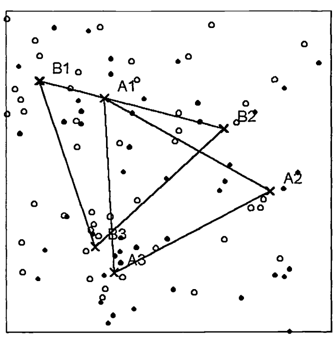

Figure 4a. 15° counter-clockwise rotation at spot (i, j)

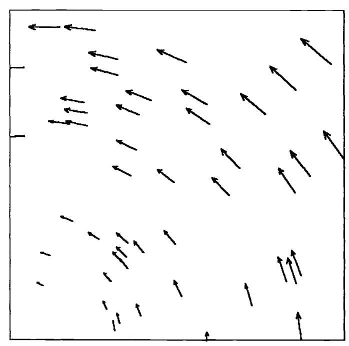

Figure 4b. Velocity vectors of the same spot

Lost particle pairs are the major source of noise in the SOFM. Interrogation spots are defined in a static coordinate, namely, the boundary of each spot does not move with the flow. Due to this reason, the particles close to the spot boundary at t_1 may leave their own spot at t_2. In the same way, new particles may come into this spot. Unexpected three-dimensional movement of the flow is another reason for particles to "appear" and "disappear". By wisely choosing the location and size of interrogation spots, the mismatch due to particles' boundary crossing can be effectively avoided but the mismatch caused by the third directional motion will still exist. For the cross-correlation technique, the number of correct vectors reduces dramatically by lost pairs, resulting in a large number of erroneous vectors. We are also eager to know the mismatching tolerance of the present method. The explicit theoretical conclusion for this question is not yet available, but we can get some useful conclusion depending on the statistical nature of the method. If the number of presentation to the SOFM algorithm is high enough, the divergence of the present method is linearly proportional to the noise rate. This result is further proved by processing a flow with numerically simulated particles. The noise rate can be controlled in this case.

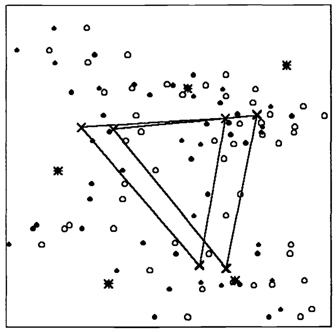

Figure 5a. Same motion as shown in Figure 3, but with added noise represented by '*'

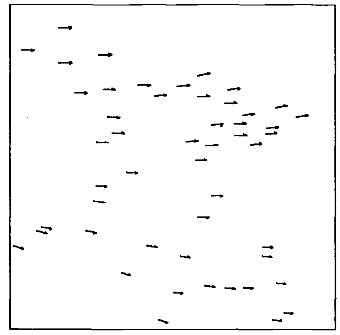

Figure 5b. Velocity vectors

Figure 5 shows the same interrogation spot as in Figure 3, but there is 10% noise. '*' represents particles without partners. The results show that the displacement divergence is around 7% for U_x.

Before this present method is used for any real flow image, a slightly complicated flow which has a known flow pattern is analyzed in order to further prove the validation of this algorithm. It is the flow between two cylinders. The inner cylinder is rotating while the outer cylinder is fixed. We analyze a part of the flow field, a square region, as shown in Figure 6a. We divide this region into 49 interrogation spots and use the SOFM algorithm for each spot. By following the same steps illustrated before, its feature map is first found by the SOFM in each spot, the velocity vectors of all particles in the region are also obtained, as shown in Figure 6c.

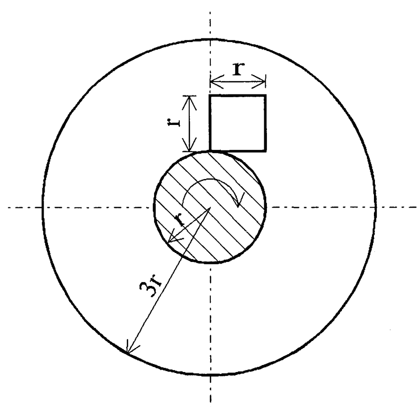

Figure 6a. Flow between two cylinders

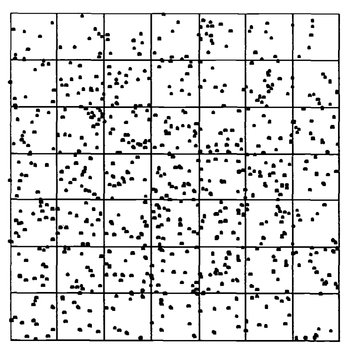

Figure 6b. Numerical simulated particles

The velocity distribution in the whole region can be interpolated as shown in Figure 6d. In the current computer program, we have not designed any wise spot to avoid particles' boundary crossing, namely, lost particle pairs. In order to manipulate the validation of input data, noise rate has been checked in each spot before the SOFM algorithm being used. The calculation is given up in certain spot if its noise rate is higher than a selected threshold. It may be noticed that there are some empty spots in Figure 6c. One of the reason is that there are less than three particles in this spot so that three equivalent particles cannot be constructed. The other is that the noise rate is higher than the threshold. The final velocities in this kind of spots are obtained by interpolating the values of their neighbors.

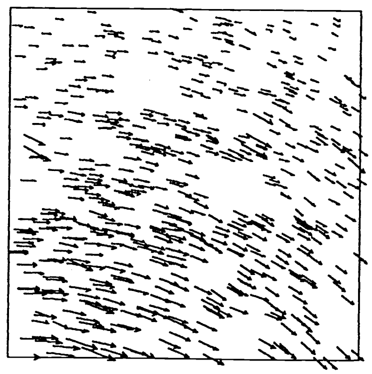

Figure 6c. Velocity vector of each particle

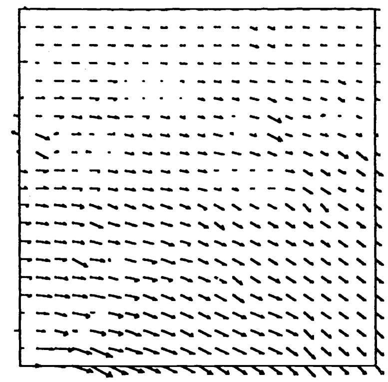

Figure 6d. Velocity distribution interpolated from Figure 6c

5. EXAMPLE

Flow passing a cavity is used as an example here. In the competition procedure in the SOFM algorithm, we can deal with either a binary image by the normal competition procedure described by equation (3), or an original gray scale image by equation (15). Figure 7a is the binary picture while velocity vectors obtained by the SOFM algorithm are shown in Figure 7b.

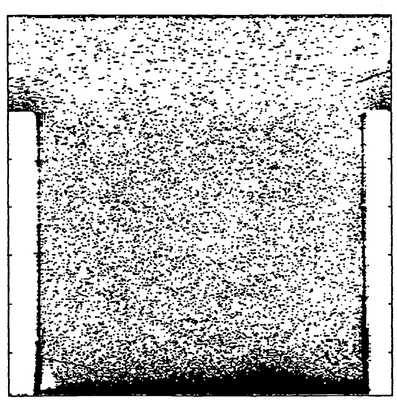

Figure 7a. Binary picture

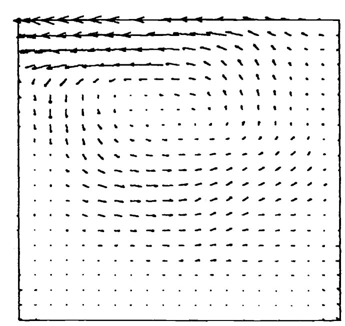

Figure 7b. Flow velocity vectors in the cavity

6. CONCLUSIONS

An intelligent tracking algorithm based on the self-organized feature map (SOFM) for particle image velocimetry has been proposed. In addition to its clear theoretical explanation, the method has been verified by real and numerically simulated images. The essence of this method is to apply the SOFM algorithm to compress many particles in each interrogation spot into three coded equivalent particles. The classical particle tracking technique is used to find the velocities of those equivalent particles, then velocity distribution in each interrogation spot can be obtained by interpolation. The following conclusions can be made for this method:

(1) The velocity in each interrogation spot is assumed to be linear instead of constant. In the correlation method, only rigid translation is assumed for each spot. Hence, much smaller interrogation spots must be used to achieve a reasonable accuracy in the region where the flow is turbulent, rotational or reversing. The calculation becomes tedious.

(2) There is no special requirement on particle number in each spot as long as its particle number is more than 3.

(3) The SOFM is a parallel algorithm in nature. It may be used for real-time, on-line PIV analysis by parallel processors.

(4) Data divergence on input noise rate is proved to be moderate by numerical simulation.

(5) Its usage is not limited to PIV. It can also be used to determine the movement of any moving bodies. Its accuracy can be adjusted by changing the time interval of two images.

7. REFERENCES

1. R. J. Adrian, "Double Exposure, Multiple-Field Particle Image Velocimetry for Turbulent Probability Density", *Optics and Lasers in Engineering*, Vol. 9, pp. 211-228, 1988.

2. C. E. Willert and M. Gharib, "Digital Particle Image Velocimetry", *Experiments in Fluids*, Vol. 10, pp. 181-193, 1991.

3. A. A. Adamczyk, "2-dimensional Particle Tracking Velocimetry (PTV): Technique and Image Processing Algorithm", *Experiments in Fluids*, Vol. 6, pp. 373-380, 1988.

4. T. Kohonen, *Self-Organization and Associative Memory*, 2nd Edition, Berlin, Springer-Verlog, 1987.

5. G. A. Corpenter, "Neural Network Models for Pattern Recognition and Associative Memory", Chapter 1, *Neural Networks*, Pergamon Press, Inc., pp. 243-257, 1989

6. S. Haykin, "Learning Process", Chapter 2, *Neural Networks*, Macmillan College Publishing Company, Inc., pp. 49-52, 1994.

7. H. Demuth and M. Beale, *Neural Network TOOLBOX for Use with MATLAB*, The Math Works, Inc., 1994.