2015 2nd International Conference on Information Science and Control Engineering

The application of SOM network to particle tracking velocimetry in a wind-blown sand flow

*Liqun Ji, Fusheng Yang**
Department of Process Equipment and Control Engineering
Xi'an Jiaotong University
Xi'an, China
*yang.fs@mail.xjtu.edu.cn

Liqun Ji
Department of Automation
Shanghai Jiaotong University
Shanghai, China

Min Guan
Department of Science and Technology Management
China Triumph International Engineering Co., Ltd.
Shanghai, China

***Abstract*—Wind-blown sand flow is the basic phenomena that have profound influences on the environment, and the experimental study on the velocity distribution of the sand particles is very important for the understanding of this phenomena. Among the various experimental techniques, particle tracking velocimetry (PTV for short) is one that attracts more and more attentions. In this paper, an algorithm based on the Self-Organizing Maps network is established for PTV in a wind-blown sand flow, and the processing results by the algorithm prove its ability to capture the characteristics of the concerning flow field.**

Keywords- wind blown sand; particle tracking velocimetry; artificial neuron network; recovery ratio; self-organizing maps

I. INTRODUCTION

Desertification is a serious environmental problem posing great threat to the health and lives of human beings. Wind blown sand flow, which is a complicated phenomena having close link to the sand transport and the formation of desertification, has attracted the attentions of many researchers since the pioneering work by Bagnold[1]. For the experimental study of the wind blown sand flow, particle tracking velocimetry (PTV for short) has been used extensively in recent years [2-5] due to its ability to capture the motion characteristics of each individual particle in the whole flow field, which provides huge amount of useful information.

In the PTV measurement, firstly a series of images should be collected for the flow within certain time interval in visualization experiments, and then the images are processed carefully. The position for the same particle during that interval will be identified through "matching", from which the particle velocity can be evaluated. Finally, the instantaneous velocity of each individual particle can be found and the particle flow field is reconstructed completely. In the abovementioned steps, matching of the same physical particles in consecutive images is of crucial importance, and the procedure to do this is termed "algorithm".

As summarized by Yang et al. [6] in a review paper, several different types of algorithms have been proposed in PTV technique. Among these algorithms, the artificial neuron network (ANN) based ones show some interesting advantages and receive many attentions. In this paper, a Self-Organizing Maps (SOM) neuron network algorithm is established and implemented using Matlab 7.0 software, and the processing results are compared with those obtained using conventional empirical algorithm to prove its capability of matching numerous particles in a complicated wind-blown sand flow.

II. ALGORITHM DESCRIPTION AND IMPLEMENTATION

A. SOM Network Fundamentals

SOM network is a typical neuron network working through unsupervised learning, which is different from the well-known Back Propagation (BP) neural network. SOM network use competitive learning rule, in which the connection weights of the winner neuron with its nearby neuron are modified, making it more conducive towards the direction of their competitive adjustment. If the modified weights of the winner neurons and neighboring neurons are close to the input values, the nearby neurons would be close to each other after several iterations. The samples with similar input mode automatically form a class, while the samples with different input mode can be discriminated, achieving the function of self-organized clustering. SOM network has good selectivity, so it was usually applied to query matching.

For the particle matching in PTV, SOM algorithm is an attractive option because of its capability of dealing with unpaired particles between two frames, moreover no a priori knowledge on the flow field is needed for the working of this algorithm, which is an important advantage in the general application to various flow conditions.

978-1-4673-6850-6/15 $31.00 © 2015 IEEE
DOI 10.1109/ICISCE.2015.114

B. Implementation of the Algorithm

Let $\boldsymbol{x}_i$ (i=1, .., N) and $\boldsymbol{y}_j$ (j=1, .., M) be the coordinate vectors of the particles in the first and the second frames, respectively. The neural network is composed of two similar sub-networks, each one corresponds to one of the two frames. The first network has N neurons (i.e. particles) located at $\boldsymbol{x}_i$ and the second one has M neurons located at $\boldsymbol{y}_j$. Since we focus on the 2-D flow field here, each neuron has two weight vectors, corresponding to the two components of the coordinate vectors $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$. The weight vectors are denoted by $\boldsymbol{v}_i$ for the first sub-network and by $\boldsymbol{w}_j$ for the second one. These weight vectors are assigned the following initial values:

$$v_i = x_i, i = 1,2,..., N \tag{1}$$

$$w_j = y_j, j = 1,2,..., M \tag{2}$$

The weight vectors are updated in the way that those of one sub-network should work as stimuli for the other sub-network. Concretely, the stimulus vector $\boldsymbol{v}_i$ from the first sub-network is presented to the second sub-network. Then, a winner neuron is selected from the latter sub-network as the one with the weight vector closest to $\boldsymbol{v}_i$. Let c be the index of this neuron and $\boldsymbol{w}_c$ its weight vector, then each neuron of the second sub-network is subjected to the following displacement of weight vectors:

$$\Delta(w)_j(c) = \alpha_j(v_i - w_c), j = 1,2,..., M \tag{3}$$

$$\alpha_j = \begin{cases} \alpha \ \ if\ neuron\ j \in Sc(r) \\ 0 \ \ otherwise \end{cases} \tag{4}$$

where $\boldsymbol{\alpha}_j$ is a scalar variable between 0 and 1, and Sc (r) denotes a closed circle region centered on the point $\boldsymbol{y}_c$ with a radius of r. Each time the weight vector $\boldsymbol{v}_i$ is presented to the second sub-network, the weight vectors of the latter sub-network are updated following (5):

$$w_j \leftarrow w_j + \sum_{i=1}^{N} \Delta w_j(c_i), j = 1,2,..., M \tag{5}$$

In the next step, by contrast, the stimulus vector $\boldsymbol{w}_j$ from the second sub-network is presented to the first sub-network. A winner neuron is selected as the one closest to $\boldsymbol{w}_j$. Each time the weight vector $\boldsymbol{w}_j$ is presented to the first sub-network, the weight vectors of the latter sub-network are updated as follows:

$$\Delta v_i(c) = \alpha_i(w_j - v_c), i = 1,2,..., N \tag{6}$$

$$\alpha_i = \begin{cases} \alpha \ \ if\ neuron\ i \in Sc(r) \\ 0 \ \ otherwise \end{cases} \tag{7}$$

$$v_i \leftarrow v_i + \sum_{j=1}^{M} \Delta v_i(c_j), 1 = 1,2,..., N \tag{8}$$

At each step when all the weight vectors from either sub-network are updated, the amplitude α of the weight translation is decreased according to:

$$\alpha = \begin{cases} C_1 \exp(-B_1 \cdot t / t_m) \ \ neuron\ i, j \in Sc(r) \\ 0 \ \ otherwise \end{cases} \tag{9}$$

where C_1 is the initial learning rate ranging between 0 and 1 (here 0.5), B_1 is a constant greater than 1 (here 10), t is the iteration number, t_m is the maximum number of iterations (here 100).

At the same time, the radius of the circle r, within which the neuron weights are changed, is decreased according to:

$$r = C_2 \exp(-B_2 \cdot t / t_m) \tag{10}$$

where C_2 is the initial winner neighborhood radius (specified as twice the maximum displacement of particle), B_2 is a constant greater than 1 (here 50), t is the iteration number, t_m is the maximum number of iterations.

These steps are iterated until the radius of the circle r reaches a given threshold value small enough to cover only the winner neuron. Since the correspondence between a weight vector and its matching neuron is not always reciprocally identical for the two sub-networks, a final nearest-neighbor check is conducted.

III. Validation of the Algorithm

A. Processing Results of Simulated Fluid Flow

It is common practice to test PTV matching algorithm using the simulated flow field [7] precisely described by known equations, here three typical flow fields were adopted for test: rotating flow, Couette flow and explosive flow. The corresponding velocity components are expressed as in (11)~(13), respectively:

$$\begin{cases} u_x = -r\omega \sin\theta \\ u_y = r\omega \cos\theta \end{cases} \tag{11}$$

$$\begin{cases} u_x = -2 \cdot U \cdot y / h \\ u_y = 0 \end{cases} \tag{12}$$

$$\begin{cases} u_x = rm \cos\theta \\ u_y = rm \sin\theta \end{cases} \tag{13}$$

Where r and θ are the polar coordinates, ω is the angular velocity of rotating flow, U is the moving velocity of plate in a Couette flow and h is the distance between the two plates, m is a constant characterizing the velocity magnitude of explosive flow. A certain number (here 1000, which is typical for such test) of pseudo "particles" are seeded at random yet known positions of the field to follow the flow, then the positions for them in the next frame after a certain interval can be predicted using (11)~(13), with certain error of Gaussian distribution added. These forms a pseudo image data set and can be processed by PTV matching algorithm. The reconstructed flow field is then compared with the standard one given by (11)~(13) to test the validity of the algorithm.

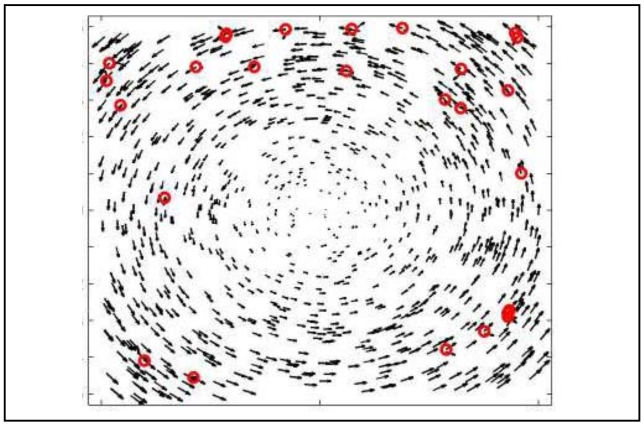

Figure 1. Reconstructed rotating flow from pseudo image by SOM algorithm

For conciseness we only present the reconstructed flow field for the rotating flow (11) by SOM algorithm in Fig.1, while the qualitative results for the Couette flow and explosive flow are largely similar to this one and skipped from detailed presentation.

We can clearly find the characteristics of typical rotating flow from the reconstructed flow field in Fig.1, although there do exist a few mismatched particles marked by red circles. The larger the particle velocity is, the more difficult the correct matching will be, which is generally the case for PTV technique.

To account for the performance of the SOM based matching algorithm quantitatively, a new index called matching ratio is defined as below, which can be taken as the combination of the conventional recovery ratio and error ratio indices:

$$Matching\ Ratio = \frac{Number\ of\ correctly\ matched\ particles}{Total\ number\ of\ particles} \times 100\% \quad (14)$$

The matching ratio of the reconstructed flow fields by SOM algorithm for (11)~(13) are evaluated by averaging the processing results of 5 pseudo data sets, and the values obtained are given in Table Ⅰ. From the table we can find that in most cases the matching ratio is higher than 97%, which implies that the minimum recovery ratio is 97% and the maximum error ratio is 3%, considering that for such pseudo image data it is unlikely to introduce more "particles" for matching than those seeded. Such performance is comparable with that of other advanced algorithms in the literature reports[7].

TABLE I. THE MATCHING RATIO OF PROCESSING RESULTS BY SOM ALGORITHM FOR DIFFERENT FLOW FIELDS

Flow fields	Matching rate
Rotating flow	97.8%
Couette flow	96.8%
Expansion flow	97.2%

B. Processing Results of Realistic Wind-blown Sand

The wind tunnel apparatus used for the measurement of wind-blown sand flow is shown in Fig. 2, and for the details the readers are referred to [5].

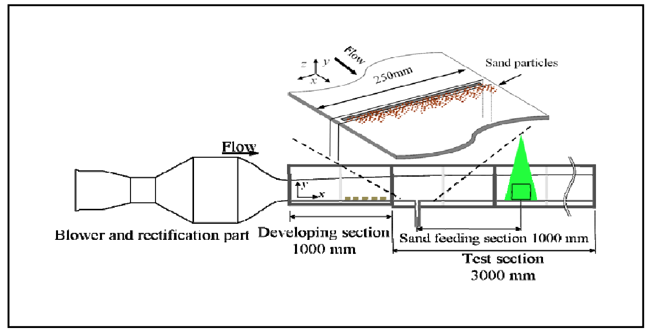

Figure 2. Experimental set-up for the measurment of wind-blown sand

In the experiment, the so-called GL particles (its density is 2500kg/m^3 and median diameter is 111μm) were fed continuously and blown by the turbulent air flow into the view region(112×112mm^2), where a thin laser sheet was shed for the collection of image data by a CCD camera. In the view region, a small rigid fence of 20mm height was installed in the middle, and the concerning turbulent flow field around it could be observed. 600 sets of image data were collected in the experiment, and each one include the positions of the sand particles in two consecutive frames with a time interval of 50μs.

Relaxation method [7] based PTV (RM-PTV) is a popular method to measure the complex flow field (e.g. local vortex flow, shear flow) in recent years. Therefore, we processed the experimental data using both SOM network algorithm and relaxation algorithm for comparison. For generality three data sets with low, moderate and high density of particles were chosen for discussion, and similar conclusion could be drawn. Here we only give the reconstructed flow field with the high particle density by the two algorithms, see Fig. 3 and 4.

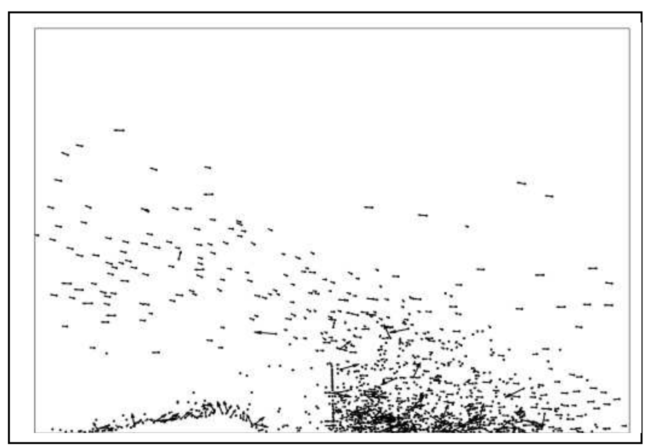

Figure 3. The reconstructed flow field by relaxation algorithm

Figure 4. The reconstructed flow field by SOM algorithm

We can see in Fig.3 and 4 that the two flow fields look quite similar, from which the so-called saltation behavior of sand particles can be recognized. Most particles are blocked by the fence, while a few ones flying with relatively high velocity are able to cross the fence to the downstream on the left side. The expected function of decreasing sand transport rate was realized by the fence. The above comparison shows that the SOM network algorithm is able to reconstruct such a complicated flow field as the relaxation algorithm does.

In order to further compare the performance of relaxation algorithm and SOM algorithm quantitatively, we introduce the definition of vector ratio expressed as below:

$$Vector\ Ratio(\mu) = \frac{Number\ of\ vectors\ by\ matching}{Total\ number\ of\ particles} \times 100\% \tag{15}$$

We choose 10 sets of data randomly for processing by using the two algorithms respectively, and the vector ratio for the results are calculated and shown in Table II (μ_1 is the vector ratio by SOM network algorithms, μ_2 is the vector ratio by relaxation algorithms).

The results indicate that the vector ratio of processing results by SOM algorithm is always higher than that by relaxation algorithm, this may be due to the removal of spurious vectors by relaxation algorithm. However, such comparison still proves the ability of SOM algorithm in reproducing the practical flow field.

TABLE II. COMPARISON OF VECTOR RATIOS FOR THE TWO ALGORITHMS

Data set No.	μ_1	μ_2
10	95.4%	91.6%
53	94.0%	89.3%
101	92.0%	91.3%
176	94.7%	90.8%
245	97.9%	94.7%
300	96.8%	92.0%
366	96.5%	94.4%
423	96.8%	93.9%
499	93.6%	90.5%
567	95.2%	94.7%

IV. CONCLUSION

In conclusion, the SOM network algorithm proposed in this paper is an effective algorithm for particle matching in PTV. Compared with conventional relaxation algorithm, the new algorithm is able to generate similar or even superior processing results in a complex flow like the wind blown sand over a rigid fence.

ACKNOWLEDGMENT

F.S.Yang thanks Prof. Kawaguchi, Dr. Tsukahara, Mr. Aoshima, Mr. Otakaguchi and Mr. Sugiyama in Tokyo University of Science for their assistance in gathering and processing the experimental data of wind-blown sand.

REFERENCES

[1] R.A.Bagnold, The physics of blown sand and desert dunes, New York: Dover publications Inc., 1954.

[2] M.Creyssels, P.Dupont, A.Ould El Moctar, A.Valance, I.Cantat, J.T.Jenkins, J.M.Pasini, K.R.Rasmussen, "Saltating particles in a turbulent boundary layer: experiment and theory", Journal of Fluid Mechanics, vol. 625, pp. 47-74, 2009.

[3] T.D.Ho, A.Valance, P.Dupont, A.Ould El Moctar, "Scaling laws in aeolian sand transport", Physical Review Letters, vol.106, pp. 094501, 2011.

[4] W.Zhang, J.H.Kang, S.J.Lee, "Tracking of saltating sand trajectories over a flat surface embedded in an atmospheric boundary layer", Geomorphology, vol.86, pp.320-331, 2007.

[5] F.S.Yang, D.Aoshima, K.Otakeguchi, T.Tsukahara, Y.Kawaguchi, "Experimental study on the local turbulence modulation in a horizontal particle laden flow with rigid fence", Proceedings of 7th Int. Symp. On Multiphase Flow, Heat Mass Transfer and Energy Conversion, Oct. 26-30, Xi'an, China, 2012.

[6] F.S.Yang, Z.X.Zhang, S.M.Wang, Y.Yawaguchi, "Particle tracking velocimetry and its application to the measurement of multiphase flow-A review", Fluid Machinery(in Chinese), vol.42(2), pp.37-42, 2014.

[7] S.J.Baek, S.J.Lee, "A new two-frame particle tracking algorithm using matching probablity", Experiments in Fluids, Vol.22, pp.23-32, 1996.