Experiments in Fluids 25 (1998) 461–473 © Springer-Verlag 1998

A genetic algorithm particle pairing technique for 3D velocity field extraction in holographic particle image velocimetry

J. Sheng, H. Meng

Abstract This research explores a novel technique, using Genetic Algorithm Particle Pairing (GAPP) to extract three-dimensional (3D) velocity fields of complex flows. It is motivated by Holographic Particle Image Velocimetry (HPIV), in which intrinsic speckle noise hinders the achievement of high particle density required for conventional correlation methods in extracting 3D velocity fields, especially in regions with large velocity gradients. The GA particle pairing method maps particles recorded at the first exposure to those at the second exposure in a 3D space, providing one velocity vector for each particle pair instead of seeking statistical averaging. Hence, particle pairing can work with sparse seeding and complex 3D velocity fields. When dealing with a large number of particles from two instants, however, the accuracy of pairing results and processing speed become major concerns. Using GA's capability to search a large solution space parallelly, our algorithm can efficiently find the best mapping scenarios among a large number of possible particle pairing schemes. During GA iterations, different pairing schemes or solutions are evaluated based on fluid dynamics. Two types of evaluation functions are proposed, tested, and embedded into the GA procedures. Hence, our Genetic Algorithm Particle Pairing (GAPP) technique is characterized by robustness in velocity calculation, high spatial resolution, good parallelism in handling large data sets, and high processing speed on parallel architectures. It has been successfully tested on a simple HPIV measurement of a real trapped vortex flow as well as a series of numerical experiments. In this paper, we introduce the principle of GAPP, analyze its performance under different parameters, and evaluate its processing speed on different computer architectures.

Received: *7 September 1997*/*Accepted*: *3 February 1998*

J. Sheng, H. Meng
Program for Complex Fluid Flow
Mechanical and Nuclear Engineering Department
Kansas State University, Manhattan, KS 66506, USA

Correspondence to: H. Meng

This work is funded by NSF through a CAREER Award CTS96-25307 and an equipment grant CTS-9700373, program directors M. C. Roco and Roger Arndt. Partial support from K*STAR is appreciated. The authors wish to thank members of the Laser Flow Diagnostics Laboratory for collaboration on related projects and Robert Brodkey for helpful suggestions for the manuscript.

1 Introduction

Holographic Particle Image Velocimetry (HPIV) is a novel three-dimensional (3D) flow diagnostics tool that takes advantage of the inherent high-resolution 3D recording capability of holography. It instantaneously records a 3D volume of particles seeded in a complex flow on holograms and reconstructs particle field images, from which the 3D velocity fields are extracted via appropriate algorithms. Encouraging HPIV progresses are reported by several groups including Meng and Hussain (1995a, b), Barnhart et al. (1994), Zhang et al. (1996), and Gray and Created (1993).

Previous investigations of HPIV have mostly focused on optimizing hologram recording and reconstruction schemes, while less research has been conducted in 3D velocity extraction algorithm development. Generally, it is assumed that conventional 2D correlation methods to extract velocity data can be extended to 3D and used for HPIV (Gray and Greated 1993). It is well-known that the success of correlation methods requires high-density particle images. "Bad vectors" are often generated in flow regions with low seeding densities or large velocity gradients (Meinhart et al. 1995). These situations are more delicate in holographic PIV, since increasing seeding density would increase speckle noise due to coherent superposition of light waves (Meng et al. 1993). For 3D volumetric recording of any means, the high-seeding densities required for successful correlation are difficult to achieve at an acceptable signal-to-noise ratio. In multi-camera recording configurations as Particle Tracking Velocimetry (PTV) (Kasagi and Nishino 1991, Guezennec et al. 1994), even a rather low seeding density ($\sim 10^3$ particles for the entire flow volume) causes grave difficulties such as image overlapping. Holographic particle image velocimetry tolerates the highest particle densities owing to the intrinsic 3D recording nature of holography. Nevertheless, typically at a density of $10 \sim 10^2$ particles/mm^3 HPIV encounters speckle noise (Meng et al. 1993). Although speckle noise can be controlled to various extents by devising innovative holographic schemes (Meng and Hussain 1995a, b; Simons et al. 1993; Zimin et al. 1993; Barnhart et al. 1994), the cost of increased system complexity can be rather high, while the achieved seeding densities are still less than allowed in a light sheet for 2D PIV and thus in most cases insufficient for successful correlation of the 3D volume of particles. In contrast to correlation, another type of algorithms, particle tracking, finds velocities by tracking individual particles through identifying the smoothest particle trajectories in a series of exposures (Kasagi and Nishino 1991; Guezennec et al. 1994) or

tracking groups of particles by imposing certain restrictions (Yamamoto et al. 1993; Okamoto et al. 1995a, b; Wernet et al. 1993, 1995; Baek and Lee 1996; Cowen and Monismith 1997). Due to high computational requirements, these tracking methods do not provide sufficient spatial resolution and processing speed to be used in holographic PIV, where the data quantity is tremendous (typically 1 Gbytes per hologram or $10^6 \sim 10^8$ velocity vectors after processing).

Clearly, new data processing approaches need to be explored for velocity extraction in HPIV. Our past experience shows that accuracy of velocity field extraction is greatly enhanced if intelligence is introduced to the particle image processing whenever statistical methods fail. With intelligence involved, there is no need to overly seed the fluid just to provide the average velocity in a finite region (as with correlation methods), nor any need for a number of successive exposures to provide the smoothest trajectories (as with tracking methods). Meng and Hussain (1995a, b) demonstrated an HPIV measurement of high-spatial-resolution 3D velocity and vorticity fields. Their data were processed from an in-line hologram via IROV (In-line Recording Off-axis Viewing) technique using correlation mixed with human-assisted particle pairing. From their experience, correlation methods fail to give the correct results in regions with large velocity gradients (such as inside a vortex core), even though the particle pairs are obvious to the operator's eyes. This shows that although sufficient information about the flow velocity field is recorded by a hologram, methods relying on statistical averaging are often insufficient for extracting wanted data. We hence propose that "smart" or "intelligent" algorithms need to be developed in lieu of purely statistical methods.

We develop a new algorithm, identified as Genetic Algorithm Particle Pairing (GAPP), which performs particle pairing using Artificial Intelligence (AI). In an abstract sense, particle pairing is a process of mapping/transformation from the space of the first-exposure particles to that of the second. These two spaces consist of particle centroid locations extracted from holographically reconstructed images. Since there exist many mapping possibilities, out of which only one corresponds to the correct velocity field, different mapping schemes need to be numerated and "filtered" for the best one to emerge. Without the aid of statistics, this filtering must rely on good judgment. Like in the common practice of post-processing to clean up "bad vectors", intelligence can be incorporated by making sense out of the vectors through fluid dynamics itself. In our algorithm, this judgment is made during the extraction of velocity vectors from particle images, not after. This "intelligence" is incorporated in our algorithm via the evaluation function in Genetic Algorithm (GA). In other words, GAPP uses fluid dynamics as its filtering criteria to pair a group of particles instead of statistical averaging, thus, it is more suitable for large velocity gradients and low particle density regions in a 3D volume.

Usually, the search for the best pairing scenario is a lengthy process in the presence of a large number of particles. However, GA offers a superior capability of searching a large solution space and inherent parallelism to pair a large number of particles simultaneously and intelligently (Goldberg 1989; Michalewicz 1992). In this way, GAPP promises to be practical.

We were informed that Tukiji et al. (1994) also proposed a GA approach for particle tracking in 2D PIV. However, they did not address the important aspects such as handling a large number of particle pairing scenarios, defining proper evaluation functions with physical meanings, designing suitable genetic operators (tailored for particle pairing), and studying algorithm performance. All of these issues are critical to the success of a GA application to velocity field extraction. Furthermore, their work did not address 3D velocity field extraction. Our GAPP algorithm aims at 3D velocity extraction by pairing a large number of particles in a 3D volume, and our evaluation functions are defined based on the dynamics of fluid flow. Supported by a set of specially designed genetic operators, our algorithm shows robustness and high processing speed. This enables our algorithm to pair particles in a 3D space accurately and efficiently. We show with this work that GAPP is not merely a concept but a practical experimental alternative to conventional correlation methods, especially for 3D velocimetry.

2
Principle of genetic algorithm particle pairing

2.1
Introduction to genetic algorithm

Genetic Algorithm is a searching process based on the laws of natural selection and genetics (Goldberg 1994). Usually, a simple GA consists of three operations: *Selection*, *Genetic Operations* (which include initialization, crossover, and mutation), and *Replacement*. The basic GA cycle is shown in Fig. 1.

The population upon which the GA evolution takes place comprises a group of chromosomes that represent possible solutions of the problem and from which parents for a new generation are selected. By evaluating fitness values for all chromosomes, a particular group of chromosomes is selected from the population as parents to generate offspring (new solutions of the problem). The better its fitness, the greater chance a chromosome stands to be selected as a parent for re-production. Fitness values for offspring are evaluated in the same fashion as for their parents, and then, chromosomes in the current population are replaced by offspring based on certain replacement strategies.

Such a GA cycle is repeated until the termination criterion is reached. If all goes well throughout this process of simulated evolution, the best chromosomes in the final population

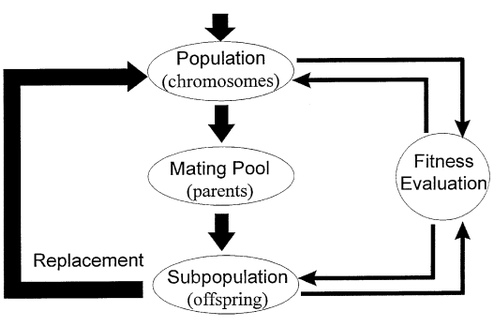

Fig. 1. A basic genetic algorithm cycle

comprise a highly evolved solution group to the problem. Although GA cannot always guarantee the absolutely correct answers, it is a practical, efficient, and highly reliable means to solving conventionally Non-Programmable (NP) problems, such as those having a huge or infinite searching space like the particle-pairing problem studied in this paper.

2.2 Particle pairing

Mathematically, the particles in two exposures form two phase spaces, P^1 and P^2:

$$P^1 = (p_1^1, p_2^1, \ldots, p_n^1) \in R^n, \quad P^2 = (p_1^2, p_2^2, \ldots, p_m^1) \in R^m \tag{1}$$

where $p_i^1 = (x, y, z) \in R^3$, $p_i^2 = (x, y, z) \in R^3$. The elements of these two phase spaces are the 3D position vectors of particle centroids. These two phase spaces, in general, do not necessarily have the same dimensions. For example, some *identified* particles are not *real* particles but speckle noise. The pairing scheme is viewed as a mapping (T) from an n-dimensional phase space P^1 to an m-dimensional space P^2:

$$P^1 \overset{T}{\Leftrightarrow} P^2 \tag{2}$$

This mapping is further represented by an $m \times n$ matrix (**T**), which transforms an n-dimensional vector consisting of particles in P^1 to an m-dimensional vector consisting of particles in P^2:

$$\begin{bmatrix} p_1^2 \\ p_2^2 \\ \vdots \\ p_m^2 \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & \ddots & & t_{2,n} \\ \vdots & & \ddots & \vdots \\ t_{m,1} & & & t_{m,n} \end{bmatrix} \times \begin{bmatrix} p_1^1 \\ p_2^1 \\ \vdots \\ p_n^2 \end{bmatrix} = \mathbf{T} \times \begin{bmatrix} p_1^1 \\ p_2^1 \\ \vdots \\ p_n^1 \end{bmatrix} \tag{3}$$

Here **T** is a linear line transformation matrix whose elements take only value 0 or 1. We further compress the matrix into a one-dimensional vector **C**:

$$\mathbf{C} = [a_1, \ldots, a_m], \tag{4}$$

where $a_i \in \{1, 2, \ldots, n\}$ and $a_i \neq a_k$ for $i \neq k$. Each element of **C**, that is, a_i, represents one pairing scheme: The index i represents the label of a particle in the first exposure, and the value of a_i represents the label of a particle in the second exposure. When $a_i = j$, particle p_i^1 is mapped into (paired with) p_j^2. Figure 2 illustrates the basic particle-pairing scheme.

2.3 Encoding scheme

Now that particle pairing is treated as a phase-space mapping problem, we apply GA to find the best mapping **C** for each group of particles being interrogated. For this purpose, each mapping scheme is first encoded into a chromosome, which is usually a string of "bits" of information. We find that the mapping vector **C**, as given in Eq. (4), can be used directly as a chromosome; in this case the chromosome is an order-based character string. This encoding scheme works well when the particle number in the first exposure, n, is no greater than that in the second exposure, m. However, n can be greater than m (or vice versa) in the presence of noise. To include this case, we modify the encoding scheme by assigning the first phase

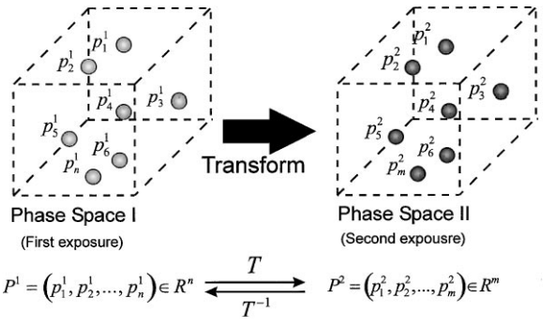

Fig. 2. Particle pairing viewed as a phase-space mapping

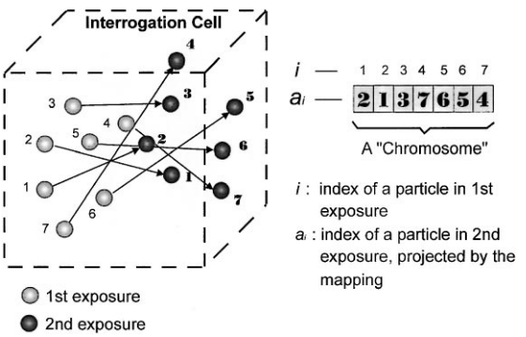

Fig. 3. Encoding scheme (chromosome) for GAPP. In this example, the chromosome is "2 1 3 7 6 5 4" indicating one possible mapping scheme

space P^1 to be the one with the smaller dimension, which corresponds to either the first or second exposure. To distinguish these two cases, we add a new "gene" (a_0), called switch, to the chromosome to indicate whether P^1 corresponds to the first exposure ($a_0 = 1$) or the second ($a_0 = 0$). Figure 3 illustrates the chromosome in our GA application as an order-based string in which a pairing scheme is encoded. As an example, "0 2 1 3 7 6 5 4" represents a mapping of particle group (1, 2, 3, 4, 5, 6, 7) from phase space P^1 into particle group (2, 1, 3, 7, 6, 5, 4) from phase space P^2, with the first digit $a_0 = 0$ indicating that P^1 corresponds to the second exposure.

2.4 Evaluation functions

Out of thousands of mapping schemes (chromosomes), only one gives the correct velocity field that meets the constraints of fluid dynamics. Hence, it is critical to define an objective function to evaluate chromosomes fitness, so that the fittest survives in GA evolution. We develop two evaluation functions, both with real physical meanings, and apply them to a flow measurement by HPIV as well as to numerically simulated flows.

2.4.1 Evaluation function based on incompressible continuity equation

When the best chromosome is established, i.e. the particles in two exposures are correctly paired, the velocity vector field

should satisfy the continuity equation of a fluid flow assuming particles follow the flow faithfully. In an incompressible flow, the continuity equation is $\nabla \cdot \mathbf{V} = 0$. Turning the statement around, if a mapping scheme produces a velocity field with divergence-free characteristics, this scheme represents the best particle pairing scheme. In this way, the continuity equation itself can be used as an evaluation function for chromosomes. Due to unavoidable experimental errors, however, the divergence of velocity field processed from particle displacements at a single point usually are not exactly zero. Hence, we only try to find a solution scheme that gives the minimum sum of velocity divergence over all would-be velocity vectors in a finite 3D region (interrogation cell). The criterion for chromosome evaluation is then represented by

$$\sum_{1}^{N} (\nabla \cdot \mathbf{V}^i)^2 = \sum_{1}^{N} (\partial_j V_j^i)^2 = \text{Min} \tag{5}$$

where N is the number of particle pairs in this interrogation cell and i denotes the ith particle pair (displacement vector).

Divergence calculation: Many direct numerical approximation methods exist for evaluating velocity divergence. The most straightforward seems to be calculating the velocity gradient tensor, $\partial_i V_j$, from the discrete (would-be) velocity field directly and then obtaining the divergence for each particle pair by evaluating the trace of the tensor matrix. The velocity gradient tensor is commonly evaluated using finite difference approximation, which requires homogeneous mesh grids for the velocity field. However, through particle pairing the recorded 3D particle image field in HPIV at best provides a sparse and non-uniformly dispersed velocity field before any post-processing procedure. Hence, direct finite difference approximations are not applicable in this problem. On the other hand, we find that by using the following equation we can get a fairly good approximation for the directional gradient (or line gradient) of the velocity vector from a sparse and non-uniformly dispersed velocity field:

$$\frac{\partial \mathbf{V}}{\partial \mathbf{n}} = \lim_{\Delta S \to 0} \frac{\Delta \mathbf{V}}{\Delta S} \cdot \mathbf{n}_0 \tag{6}$$

where $\mathbf{n}_0$ is the unit vector along the direction of $\mathbf{n}$; and $\Delta \mathbf{V}$ is the difference in the velocity vector over a short segment ΔS. This means that the directional gradient is able to be resolved by finite difference approximation methods. Therefore, we propose to calculate velocity gradients from the resolved directional gradients based on the definition:

$$\frac{\partial V_j}{\partial \mathbf{n}} = (n_0)_i \partial_i V_j, \quad i, j = 1, 2, 3 \tag{7}$$

In Eq. (7), left-hand side term $\partial V_j / \partial \mathbf{n}$ and unit vector $(n_0)_i$ can be obtained from the scattered velocity field directly; only velocity gradient tensor $\partial_i V_i$ is unknown. Hence, by solving the three equations, we are able to compute all nine elements of the velocity gradient tensor at a point and back out the divergence, $\partial_i V_i$. To provide some redundancy in order to reduce experimental error, we make use of q ($q \geqslant 3$) directional gradients at a point and solve the equation set:

$$\underbrace{\begin{bmatrix} n_1^{(1)} & n_2^{(1)} & n_3^{(1)} \\ n_1^{(2)} & n_2^{(2)} & n_3^{(2)} \\ \vdots & \vdots & \vdots \\ n_1^{(q)} & n_2^{(q)} & n_3^{(q)} \end{bmatrix}}_{\text{A}} \times \underbrace{\begin{bmatrix} \partial_1 V_1 & \partial_1 V_2 & \partial_1 V_3 \\ \partial_2 V_1 & \partial_2 V_2 & \partial_2 V_3 \\ \partial_3 V_1 & \partial_3 V_2 & \partial_3 V_3 \end{bmatrix}_{(x,y,z)}}_{\text{T}}$$

$$= \underbrace{\begin{bmatrix} \frac{\partial V_1}{\partial n^{(1)}} & \frac{\partial V_2}{\partial n^{(1)}} & \frac{\partial V_3}{\partial n^{(1)}} \\ \frac{\partial V_1}{\partial n^{(2)}} & \frac{\partial V_2}{\partial n^{(2)}} & \frac{\partial V_3}{\partial n^{(2)}} \\ \vdots & \vdots & \vdots \\ \frac{\partial V_1}{\partial n^{(q)}} & \frac{\partial V_2}{\partial n^{(q)}} & \frac{\partial V_3}{\partial n^{(q)}} \end{bmatrix}}_{\text{Q}} \tag{8}$$

where $n^{(k)}$ denotes the kth unit directional vector at the point (x, y, z). In this way, we obtain a fairly good estimate of the velocity divergence for each pairing scheme (chromosome, or "would-be" velocity field). We can then evaluate this pairing scheme based on the minimum-divergence criterion.

It can be argued that discrepancies between particle velocities and fluid velocities are small enough so that they do not affect the effectiveness of the minimum-divergence criterion for filtering out erroneous mappings. Otherwise, the PIV measurement itself should be questioned. By the same token, the approximation of velocities by displacements over a finite time interval Δt is considered sufficiently accurate. These considerations stress the general requirements for successful PIV measurement, e.g. the use of neutrally buoyant particles and short double-exposure separations.

2.4.2 Evaluation function based on particle-group morphology conservation

Unlike the first evaluation function, the evaluation function based on particle-group morphology conservation is applicable to both incompressible and compressible flow. It is based on the assumption that over a small time interval Δt, the morphological change of a particle group in a continuum medium is minimum. The morphology of a particle group is defined by connecting adjacent particles to form a triangle mesh *in a certain sequence*. The morphology of the mesh is precisely conserved only when $\Delta t \to 0$. However, as long as the double-exposure interval Δt is kept small, a maximum similarity (or minimum morphological change) exists between the patterns of the two particle groups. In other words, a "wrong" sequence in one particle group (exposure) displays a larger morphological deviation from the other particle group (exposure). Hence, the goal of this pairing scheme evaluation function is to find the most similar particle patterns between two exposures.

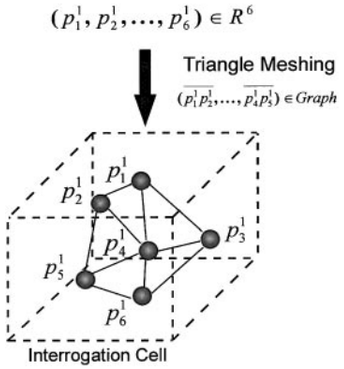

Fig. 4. Morphology of a 6-particle group in one interrogation cell, represented by vectors connecting the particles in a certain sequence

The morphology of a particle group in either P^1 or P^2 is expressed by a triangle mesh (a 3-D graph) whose vertices are the individual particle centroids, as illustrated in Fig. 4. Using the notification introduced in Eq. (1) to represent the vertices, a mesh is represented by morphology vectors $\mathbf{M}^1$ or $\mathbf{M}^2$ given in Eq. (9).

$$\begin{aligned}\mathbf{M}^1(\mathbf{C}) &= (\overline{p_1^1 p_2^1}, \ldots, \overline{p_i^1 p_j^1}, \ldots, \overline{p_{n-1}^2 p_n^1}) \\ \mathbf{M}^2(\mathbf{C}) &= (\overline{p_1^2 p_2^2}, \ldots, \overline{p_i^2 p_j^2}, \ldots, \overline{p_{n-1}^2 p_n^2})\end{aligned} \tag{9}$$

A pairing scheme $\mathbf{C}$ establishes the corresponding relationship between two meshes from different exposures. Then, the morphological similarity of two meshes is defined as the correlation of their morphology vectors in corresponding sequence:

$$\text{Similarity}(\mathbf{M}^1, \mathbf{M}^2; \mathbf{C}) = \sum_{i=1}^{\text{Dim}(\mathbf{M}^1, \mathbf{M}^2)} \sum_{i=1, i \leqslant j}^{\text{Dim}(\mathbf{M}^1, \mathbf{M}^2)} \frac{\| \overline{p_i^2 p_j^2} - \overline{p_i^1 p_j^1} \|^2}{\sqrt{\| \overline{p_i^1 p_j^1} \|^2 \cdot \| \overline{p_i^2 p_j^2} \|^2}} \tag{10}$$

where $\text{Dim}(\mathbf{M}^1, \mathbf{M}^2)$ is the dimension of the morphology vectors. The similarity representing the fitness value of a chromosome $\mathbf{C}$ is to be maximized to find the best chromosome by GA.

2.5 Parent selection

Having defined chromosome and evaluation function, we now describe the GA evolution procedure including Parent Selection, Genetic Operations, and Replacement.

Parent selection emulates the mechanism of survival of the fittest in nature. We expect a fitter pairing scheme (which meets objective better) to stand a greater chance of being selected as a parent and produce a larger number of offspring. Many parent-selection mechanisms have been developed in the GA field including ranking, tournament, and proportionate schemes (Davis 1991). In this work, we implement a commonly used proportionate scheme, Roulette Wheel Selection (Goldberg 1989). The selection procedure to choose one parent follows three major steps:

1. The fitness value for each chromosome is evaluated, and the total fitness of the whole generation (summation of all chromosomes' fitness) is calculated.
2. The chromosomes in one generation are sorted according to descending fitness. The relative fitness (growth rate) for chromosome $\mathbf{C}_j$ is calculated from

$$G(\mathbf{C}_j) = \frac{\sum_{i, f(\mathbf{C}_i) < f(\mathbf{C}_j)}^{M} f(\mathbf{C}_i)}{\sum_i^M f(\mathbf{C}_i)}, \quad j = 1, \ldots, M \tag{11}$$

   where M is the number of chromosomes in one generation; $f(\mathbf{C}_i)$ denotes the fitness value for chromosome $\mathbf{C}_i$; $G(\mathbf{C}_j)$ is the growth rate for chromosome $\mathbf{C}_j$, varying from 0 to 1. Chromosomes that are sorted by growth rate form a selection wheel. Each slot of the whell has a width $P(C_j) = G(C_j) - G(C_{j-1})$, which represents the probability of chromosome $\mathbf{C}_j$ to be selected as parent, like a dice.
3. A random number p_s between 0 to 1 is generated to be the parent selection possibility. The chromosome whose growth rate value is immediately above p_s is selected from the wheel as one parent.

The third step repeats until enough parents are selected to produce the next generation. Clearly, the higher fitness and higher growth rate a chromosome has, the better chance it stands of being selected as parents for reproduction. In fact, fitter chromosomes often get selected multiple times.

2.6 Genetic operations

Proper design of genetic operations is critical to a successful Genetic Algorithm. It is especially necessary in the current work, where standard genetic operations are inadequate due to some special features of the problem. First, our searching space is a dynamic, multidimensional, non-numerical discrete space while conventional genetic operations are designed for searching continuous numerical spaces. Second, our chromosome is a character string instead of a binary string normally encountered in GA operations. Therefore, a new set of genetic operators has been designed, as described below.

2.6.1 Initialization

Initialization creates the first generation of chromosomes for evolution. The distribution in the first generation in the solution space can significantly affect the algorithm convergence speed. Our experience shows that a homogeneous distribution of chromosomes in the first generation provides the fastest convergence as well as the least probability for the searching procedure to be caught by local optima. Therefore, it is necessary to generate a homogeneously distributed initial generation in an effective way. Since our searching space of particle pairing schemes is a multidimensional discontinuous discrete space, a fast and homogeneous initialization operator leads to a fast convergence searching. Conventional initialization methods, designed mainly for continuous numerical spaces, do not work properly for this problem. Some initialization methods have been developed for order-based character-string chromosome representations, such as the greedy algorithm and the nearest-neighbor algorithm, both designed for the Traveling Salesman Problem (TSP) (Johnson 1990).

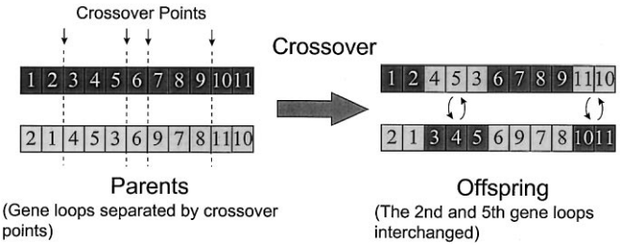

Fig. 5. Example of multi-point crossover operation with gene loop detection for the particle pairing algorithm

However, these methods are often used under certain specific conditions and cannot be generalized for other applications. Also, the speed of these algorithms are rather low when the searching space is large. Our initialization operator is designed as a wavelet-like transformation. In this method we initialize the whole population by a set of linear transformation. With this operation a homogenous population is produced at a high speed. Further discussion on this subject is beyond the scope of this paper and, therefore, will be reported in the future.

2.6.2 Crossover

Crossover combines sub-parts of two parent chromosomes to produce offspring that contain some parts of both parents' genes. Unfortunately, the simple multiple-point crossover operation generates many invalid offspring for particle pairing. This is because chromosomes representing particle-pairing schemes must meet some restrictions:

1. All genes in a chromosome must be unique. For example, "1 2 3 4 1 3" is an invalid chromosome since particle with index "3" may not appear twice in the same group.
2. All values from 1 to N must appear in a chromosome once and only once.

After further investigation, we find that in analogy to the real world, every chromosome can be decomposed into one or several gene loops which have the smallest meaningful information segments. Further decomposition of these loops results in incomplete information. Therefore, in our algorithm every gene loop operates as a *single* gene during the crossover stage. The algorithm first detects gene loops and then performs crossover operation on them. Figure 5 shows an example of a multiple crossover operation with gene loop detection in our algorithm whereby "3 4 5" in one parent and "4 7 3" in the other cannot be further decomposed. If they were decomposed, then crossover between "3" and "4" across the two strings would produce "1 2 3 4 4 5 6 7 8 9 10 11" and "2 1 3 5 3 6 9 7 8 11 10." Both of these chromosomes contain repeating genes and thus are invalid.

2.6.3 Mutation

Mutation is an operator that introduces variations within one chromosome. The operation only occurs occasionally, with a small probability p_m. Because of the aforementioned constraints, the mutation operator here is defined as the interchange of any two genes in one chromosome. In order to prevent the searching procedure from being caught by local optima as well as from vibrating and delaying the convergence speed, we vary the mutation probability from generation to generation. A larger p_m introduces more variations into new generations, and thus, enables the searching procedure to "surf" the whole solution space. On the other hand, a smaller p_m guarantees the fitter chromosomes to be preserved in the offspring. This dynamically varying mutation parameter scheme is found to serve aforementioned two purposes well in our experimental and numerical tests. Acceleration in processing speed is clearly observed.

2.7 Replacement strategies

After generating offspring, an old generation (a group of chromosomes) is replaced by a new generation. For ease of implementation, we keep the population constant. To make sure that the better pairing schemes produce more offspring, some of the best chromosomes are copied from the old generation into the new one. This replacement strategy increases the probability of conservation of better genes (correct particle pairs) in the chromosomes, thereby accelerating the algorithm convergence speed. The entire GAPP evolution procedure is illustrated in Fig. 6.

3 Test of GAPP in an HPIV experiment

In our preliminary HPIV test system, a trapped vortex flow was generated by horizontally moving a vertical plate in a water tank while allowing a large gap between the plate and the bottom. The flow was recorded by using the IROV holography method introduced by Meng and Hussain (1995a, b). Water flow in a 208 mm × 89 mm × 105 mm tank was seeded with 10.2 μm polystyrene spherical particles at a seeding density of about 5 particles/mm^3. Simple enough to give a clearly defined vortex structure, the flow represents one of the basic but rather difficult structures for velocity extraction algorithms. The difficulty comes from the large velocity gradients and low seeding density inside the core.

A double-pulsed Nd : YAG laser beam (532 nm) was collimated and launched perpendicularly to the tank and the hologram plate placed 5 cm behind the water tank wall. This laser was a regular PIV-400 model from Spectra-Physics and did not have injection seeding or other mechanisms to boost its coherence length. This limitation was one of the main reasons for our choice of the IROV holographic scheme.[1] The time interval between the two pulses was 3 ms. In hologram reconstruction, an Argon-Ion laser (514 nm) was used. A 512 × 480 CCD video camera with a 4 × microscope objective, which gave a 2.5 mm × 2 mm viewing area, was used to interrogate the reconstructed flow at an angle of 10° from the principle optical axis. The whole flow field, 3 cm × 3 cm × 3 cm, was divided into 15 × 15 × 20 Interrogate Cells (IC). Each IC had the dimensions of 2 mm × 2 mm × 1.5 mm. To acquire 3D

[1] A more sophisticated holographic PIV system using off-axis holography is currently being implemented at our laboratory using an injection-seeded PIV-400 laser from Spectra-Physics

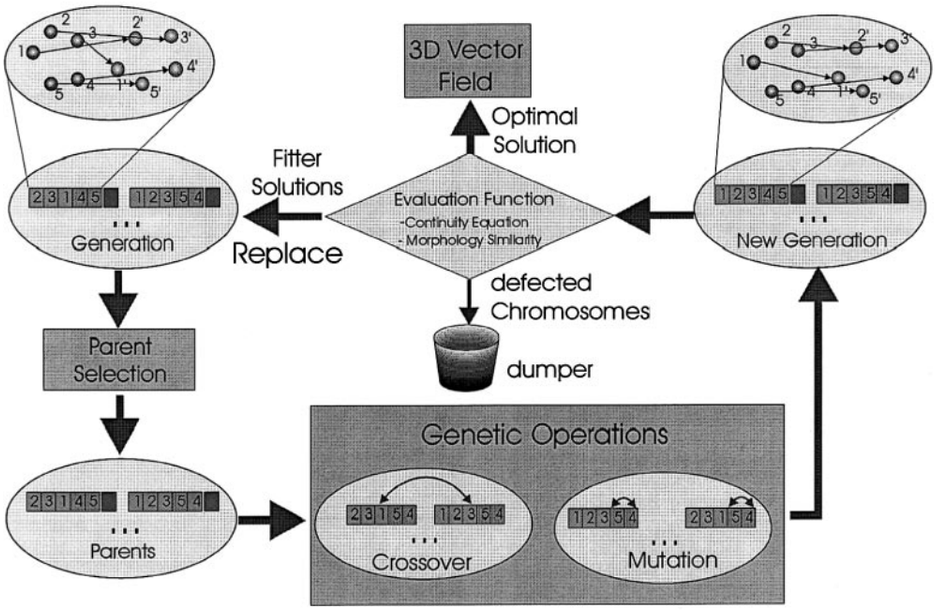

Fig. 6. GAPP evolution procedure

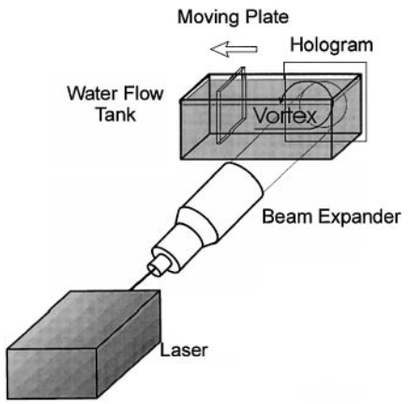

Fig. 7. Holographic PIV recording of a trapped vortex flow using a double-pulsed Nd : YAG laser (532 nm). The flow was seeded by 10.2 μm polystyrene particles at a density of 5 particles/mm^3. The hologram was interrogated via IROV technique

particle images, these ICs were scanned by traversing the camera using a high-precision 3D positioning system. Figure 7 shows the flow facility and in-line recording scheme of a vortical flow. To apply the particle pairing technique, particles of the two exposures need to be separated. There are many ways in HPIV and PIV to separate the two images, including double reference beams (Barnhart et al. 1994), angular recording (Meng 1994), and global shifting (Meng & Hussain 1991). Since these techniques introduce some optical complications not essential for testing the GAPP algorithm, we used manually- assisted particle recognition just to identify particles and separate the two exposures. After particle centroids were located and separated, they were encoded into position vectors. These position vectors were then input into two "particle pools" that correspond to the two exposures.

The implemented GAPP process (based on master–slave architecture) can be described in a nutshell as follows. It works with two continuously refreshing particle pools and runs in pipeline with particle acquisition, which eventually goes through the whole 3D volume. When a pool is filled with a certain number of particles (whose displacements are roughly contained in the spatial domain of the particle group), the master program converts all the particles into a phase space. Meanwhile the pools are emptied for new particle inputs. The master program starts to initialize the first generation of pairing schemes and triggers the genetic evolution cycles. In one cycle, the master selects two parent chromosomes at a time, and passes them to either one of the two slave programs, which run separately on two physical processors, for crossover, mutation and fitness evaluation. The slave program returns two offspring with fitness values back to the master program and the generation is updated (evolved). This particle pairing cycle continues until a pre-set criterion (based on the evaluation function being employed) is met and velocity extraction for this particle group is completed. When all 3D velocity vectors are obtained, this non-uniform velocity field was interpolated onto a uniform grid and its vorticity field was calculated.

Figure 8 shows the 3D velocity field extracted by the GAPP algorithm from the holographically recorded vortical flow. Figure 9 shows one plane of the velocity field superimposed by its vorticity contour. From images, we see a clear vortex sitting at the left quarter of the plots. These results show that GAPP performs reasonably well in flow regions with large velocity gradients and low seeding density such as in the vortex core.

Both evaluation functions described in Sect. 2.4 were tested in this vortical flow to be effective and produced the same correct velocity field. But the evaluation function based on incompressible continuity equation provided a faster processing speed than that based on morphology conservation.

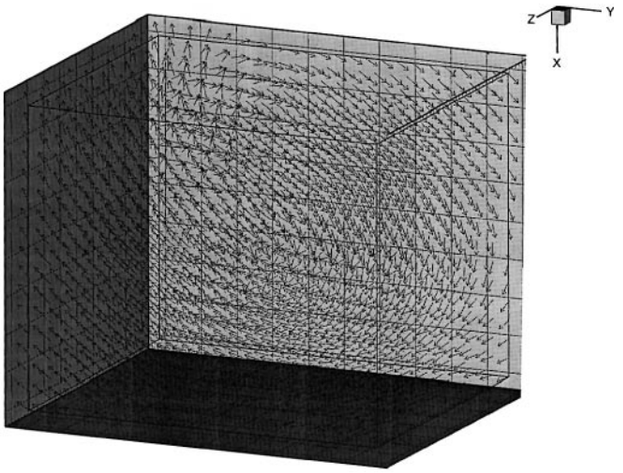

Fig. 8. 3D velocity field for a trapped vortex flow processed from a hologram using GAPP

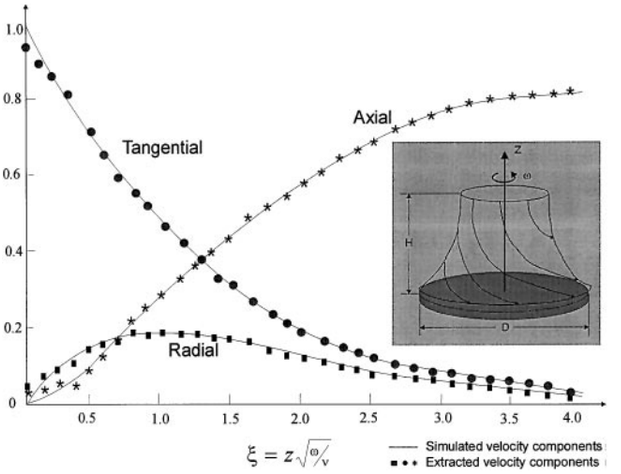

Fig. 10. Simulated flow field near a disk rotating in a viscous fluid compared with that extracted by GAPP

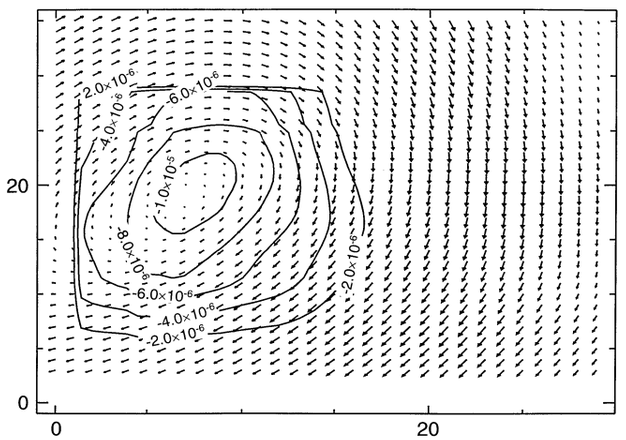

Fig. 9. A plane of trapped vortex velocity field (2D projection) interpolated onto a uniform grid, from which vorticity field is computed. Vorticity contours are shown

4
Algorithm accuracy and performance evaluation

To complete the transition from the concept to a practical tool, the GAPP algorithm needs to pass systematic evaluations of its performance including correctness, noise suppression, processing speed, parallelism, and scalability. Among these aspects, correctness and processing speed are the most critical. In this section, we conduct systematic numerical experiments to test the performance of GAPP in these two aspects.

4.1
Numerical experiment setup

All numerical experiments were conducted in a simulated 3D viscous flow near a rotating disk ($D=4$ cm, $H=4$ cm) with a known exact solution (Schlichting 1979), as illustrated in Fig. 10. The flow was seeded by "virtual" particles at a density of 100/mm^3 to a total number of 10^5 particles. Displacement of each particle was calculated by integrating the local velocity over a small time interval (5 ms). The displacements gave rise to the particle positions in the second exposure.

To simulate the effect of image noise, two percent of total particles in the second exposure was deleted. Their counterparts in the first exposure, then, were considered as noise particles. Consider three models of real-world situations: (i) no noise particles are present in either exposure; (ii) only one exposure contains noise particles; and (iii) both exposures contain noise particles. In the current work, we focused on the first two cases, which were simulated and tested in the experiments. The third case is rather difficult. We have developed a very sophisticated procedure for the algorithm to identify and eliminate noise particles from particle groups in the third case, but this procedure slows down the processing speed. On the other hand, our preliminary study shows that a good evaluation function (such as our minimum total divergence of extracted velocity field) can help in this situation to find and preserve most correct particle pairs during the GA evolution, even in the presence of noise particles in both particle groups. Hence, without evoking noise identification and elimination procedure, most particle pairs found by GAPP will be correct except for a few bad vectors when noise is present in both particle groups. To be sure, noise suppression is critical to the success of GAPP as well as other velocity extraction algorithms for HPIV, and therefore it deserves to be studied separately.

To test its accuracy or robustness of the velocity extraction results, we applied GAPP to the simulated flow since its exact velocity field was known. This experiment was conducted on a dual Pentium Pro (180 MHz) Processor PC using a sequential code. Then we evaluated the algorithm's speed performance by conducting numerical experiments using the same sequential implementation of GAPP. Finally, to evaluate the algorithm's feasibility and speed in parallel processing, we introduced parallel implementation of GAPP on three different computer architectures based on their corresponding parallel programming paradigms. Details of the setup for parallel GAPP experiments are given in Sect. 4.3.

4.2
Robustness of GAPP velocity extraction

The robustness of GAPP velocity extraction was tested by comparing the 3D velocity field extracted from the "seeded" virtual particles with the simulated flow field itself. Figure 10 shows the three velocity components plotted against the similarity variable ξ, defined as $z\sqrt{\varpi/v}$, where z is distance from the surface, ω is the disk rotational velocity and v is the kinematic viscosity. It is evident that the two results match with each other very well except in the near disk region. The discrepancy in the near-disk region was due to the approximation by particle displacements. The particle pair detection rate was 100% for this simulated flow with 2% noise particles.

4.3
Sequential particle-pairing speed

The speed of the GAPP algorithm is influenced by the number of particles per Interrogation Cell, N_c, and the population size, S. The particle number per IC usually indicates how many particles need to be paired at a time and how many elements a chromosome contains. Therefore, it determines the total time for calculating the fitness of each pairing scheme. Since more particles need more time for evaluating each chromosome, an increasing N_c would lower the algorithm speed. We performed a numerical experiment to evaluate the effect of N_c on the processing speed using a sequential particle-pairing code running on a dual Pentium Pro Processor (180 MHz) PC. For each N_c value that we investigated, we conducted 1000 independent pairing experiments and found their average pairing time, T, for obtaining the correct velocity field solution. The population sizes for each experiment was fixed at 100 chromosomes. Figure 11 shows the results plotted as pairing time (T) vs. particle number per IC (N_c). Oscillation of the curve is mainly due to initial population size effects: a different pairing size (N_c) requires a different initial population size, S, to reach its maximum speed. Generally, our result shows that the algorithm provides rather reasonable speeds even when a large number of particles need to be paired simultaneously.

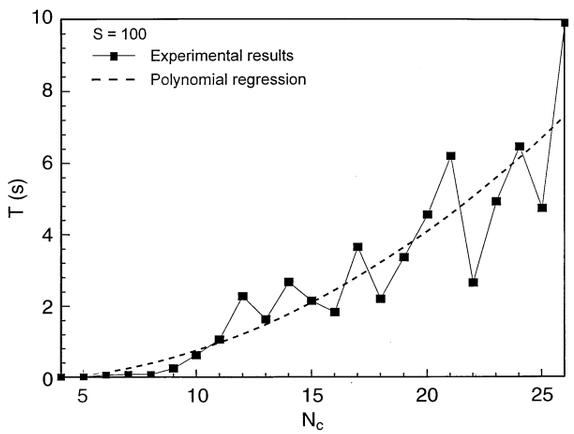

Fig. 11. Average pairing time, T, vs. Particle number per interrogate cell, N_c. Population size is fixed at 100. Results are based on a sequential code

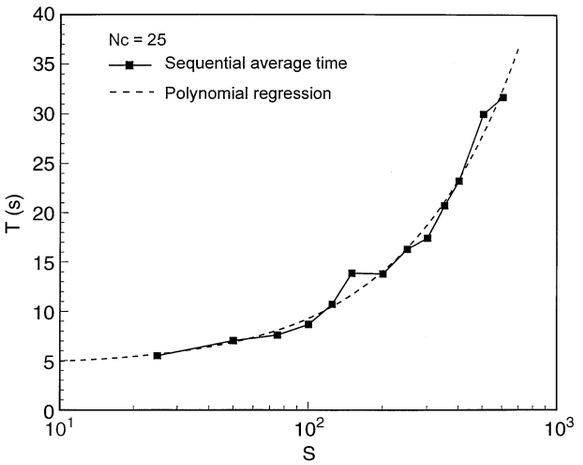

Fig. 12. Average pairing time, T, vs. generation size, S. The exponential growth of T indicates the algorithm performance is degraded when S increases. Results are based on a sequential code

A larger population size takes a longer time to produce one generation; in other words, it tends to converge more slowly. A smaller size evolves faster, but it cannot maintain many diversified chromosomes so that the generation may be caught by a local maximum. On the other hand, a larger population size generates a stable solution set and suppresses severe vibrations. We designed an experiment to investigate the effects of the population size. Figure 12 shows average pairing time (T) vs. population size (S). Notice that the curve has a rather "flat" region for initial population size from 100 to 1000 chromosomes. This slowly changing region shows that the algorithm provides both fast convergence speeds and stable solutions in this region. Further experiments showed that when the population size was in this region, total particle pairing time for 10^6 particles – a typical order for the task of interrogating one hologram – was 10 h for sequential implementation of the algorithm.

Overall, our sequential GAPP code experimental results show that within a large range of pairing particle numbers and initial population sizes, GAPP provides very reasonable processing speeds. However, because GAPP mainly involves intensive logic operations instead of floating-point computations (like in correlation method), its efficiency decreases when particle seeding density increases. In the following section we discuss the feasibility of pairing particle parallelly to boost the processing speed further.

4.4
Parallel GAPP performance

As we have noticed, the solution space of GAPP, which is the permutation of particle pairs in two phase spaces, p^1 and p^2, increases its size drastically with the respect to the particle number. Hence, processing can be slowed down significantly when a large number of particles need to be paired. The main cause for processing speed slowing down is the increased demand for logical operations. To expedite the processing procedure of GAPP in high seeding density or large measurement volume, parallel processing is the natural remedy. Hence,

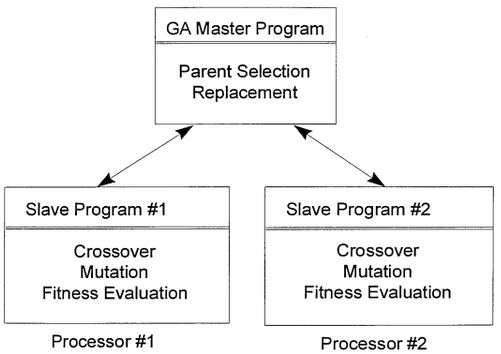

Fig. 13. Scheme for parallel GAPP implemented in a mater–slave architecture

it is important to extend GAPP algorithm from sequential to parallel implementation.

Genetic Algorithm provides strong inherent parallelism (Clipperfield et al. 1994; Petty et al. 1987), which has been explored in many different computer architectures in the form of Global GA, Migration GA (MGA), and Diffusive GA (DGA). With MGA and DGA, multiple "tribes" evolve on different computer processors with occasional migration between tribes, and the overall performance depends highly on the use of certain computer architectures, e.g. the hyper-node supercomputer. With Global GA, there is only one tribe, whose evolution is accomplished by coordinating multiple processors in a master-slave architecture (Tanse 1990). Since Global GA offers the maximum flexibility for a heterogeneous computer architecture, it is an attractive framework for a continuously-evolving technique like Holographic PIV in an era of fast-advancing computer technology. Therefore, Global GA is adopted in our parallel GAPP algorithm. In our parallel GAPP, a master process selects two parents from an old generation at a time. These parents are dispatched instantaneously to two physical processors, each of which runs a slave program to generate offspring through crossover and mutation and to evaluate the fitness value of new chromosomes. Offspring with fitness values are then sent back to the master program for replacement operation. Figure 13 shows the computer architecture for a master-slave Global GA Particle Pairing.

In this section, we investigate the feasibility of parallel master-slave GAPP algorithm in all existing parallel processing paradigms and examine their parallelization characteristics, speedup and scalability.

Parallel implementation setup: We implemented three parallel paradigms in the model of Global GA to mimic three existing parallel processing paradigms, as described below.

(a) Multiple Instruction Single Data (MISD), or Shared Memory Paradigm (SMP), was implemented on a dual Pentium Pro Processor PC. Two slave processes ran on two different physical processors and operated on data stored in the shared memory.

(b) Multiple Instruction Multiple Data (MIMD), or Message Passing Paradigm (MPP), was implemented on a workstation cluster of 11 SunSPARC10 workstations. This paradigm is also known as Distributed Parallel Computation Paradigm. Eleven slave programs were executed separately on different workstations. A master program sat on one workstation with the largest memory where all the particle information was stored. The slave and master programs exchanged data through a PVM-like (Parallel Virtual Machine) communication package that interconnects with remote workstations, evoke and execute programs on remote UNIX Workstations and Windows NT PCs.

(c) Hybrid Paradigm is a combination of SMP and MMP. Each machine was connected with others through Message Passing. Each machine could have multiple physical processors to run slave processes. The processes running in a multiprocessor machine used SMP to coordinate among themselves. One master program dynamically detected the workload of each machine including communication overhead and computation time. This master program determined the data packet size for the next patching to allow the algorithm to optimize its global performance. This active dynamic control makes the Hybrid Paradigm more flexible. We configured nine single-processor SunSPARC10 workstations and one dual-processor Pentium Pro PC as computation nodes in MPP. The dual processors in the PC were further configured to work in SMP.

Speedup: Speedup is evaluated as the parallel processing speed versus the sequential processing speed. It is directly related to workload and processor number (Lester 1993). The workload can be represented by parallel-to-sequential ratio, $F_{p/s}$, which is defined as the number of operations that can be parallelized divided by the remaining number of operations. It can be shown that

$$F_{p/s}=\frac{2N_c^2}{S+1} \tag{12}$$

The main parallel operations include crossover, mutation, and fitness evaluation. The larger the number of particles to be paired in one IC (N_C), the longer a chromosome is, and hence the more parallel workload. Initialization and parent selection, on the other hand, are the main sequential operations. Hence population size S affects $F_{p/s}$ adversely. Figure 14 shows the log plot of theoretical maximum speedup of the algorithm vs. parallel-to-sequential ratio $F_{p/s}$. From the plot speedup increases with $F_{p/s}$ and number of processors N_p. From Eq. (12), we see that a larger particle number per IC and a smaller chromosome number per generation result in a higher parallel-to-sequential ratio, and $F_{p/s}$, and thus, a higher speedup. This implies that the GAPP algorithm has an intrinsically better parallelism in handling larger data sets than smaller data sets. This is a major distinction between our algorithm and conventional Particle Tracking methods.

Scalability: Scalability is another important feature to be considered for a parallel algorithm. Two experiments are designed to test scalability for MPP and Hybrid architectures. The programs run on eleven SunSPARC10 workstation clusters and a hybrid architecture with nine SunSPARC10 workstations and one dual processor PC, separately. Figure 15 shows the experimental results of speedup vs. processor number for MMP. It shows an exponential growth of speedup with respect to processor number. The results for Hybrid Paradigm demonstrate the similar relationship between speedup and

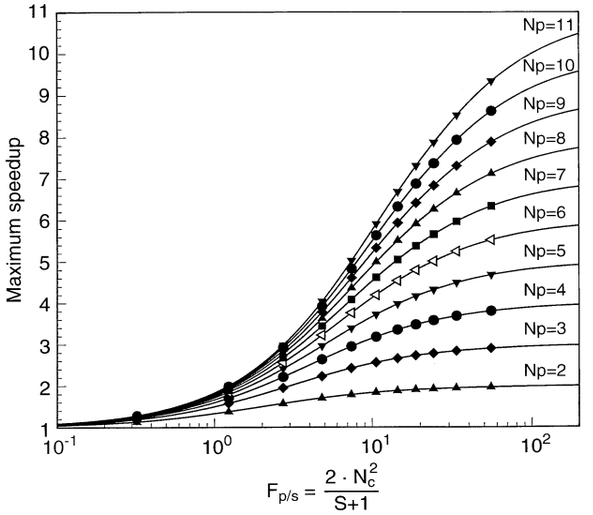

Fig. 14. Theoretical maximum speedup vs. parallel-to-sequential ratio $F_{p/s}$, evaluated for different numbers of processors

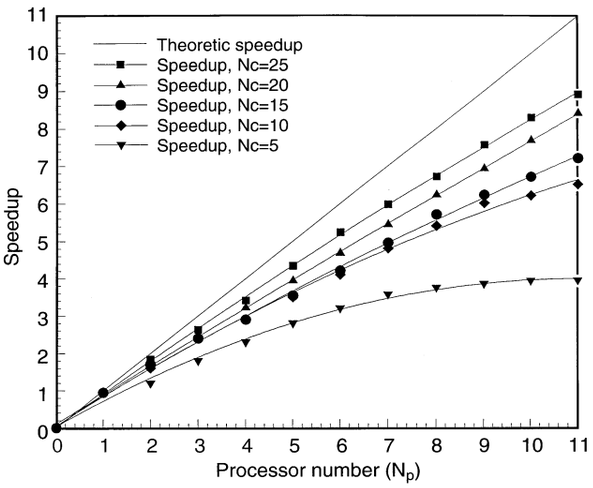

Fig. 15. Speedup vs. processor number for MPP running on a cluster of 11 networked single-processor SPARC10 workstations

processor number. However, Hybrid Paradigm shows slightly better scalability than MMP, especially with large $F_{p/s}$ ratios. Because of the use of a dynamic workload detector, Hybrid Paradigm approaches the theoretical linear speedup more closely.

Table 1 highlights the speedup for three major parallel processing paradigms and their corresponding architectures tested in this study.

4.5 Comparison with other algorithms

We would like to compare GAPP with existing PIV velocity extraction algorithms, roughly categorized into particle tracking and correlation, in the context of 3D velocity field extraction. Particle tracking algorithms are optimization

Table 1. Speedup for three major parallel processing paradigms and architectures

Architecture	Parallel processing paradigm	Speedup
Dual Processor Pentium Pro 180 MHz (2 CPUs)	Shared Memory	1.5–1.9
11 clustered SunSPARC10 via 10Base T connection (11 CPUs)	Message Passing	9
9 SunSPARC10 and 1 dual-processor Pentium Pro 180 MHz (11 CPUs)	Hybrid Paradigm	10

procedures which find the smoothest particle trajectories among several consecutive frames or search for the most probable displacements between two exposures. Most of them are developed for low particle density situations. When seeding density increases, the task for searching the optimization space becomes too huge to be tackled by normal optimization methods. Such methods, while appropriate for low-density, multi-camera PTV (Kasagi and Nishino 1991; Guezennec et al. 1994) that tracks only several hundred particles for the entire flow, are inappropriate for HPIV, which handles millions of particles. Also, optimization criteria for these methods often lack robust physical meanings so that the detected velocity vectors often have biases. On the other hand, correlation-based methods make use of statistical averaging and high floating-point performance of computers to extract velocity rapidly. However, they are mainly designed for two-dimensional or stereoscopic 2D flow measurements. A three-dimensional volume correlation for 3D velocity extraction (Gray and Created 1993) has the drawbacks of large memory consumption and low processing speed. Furthermore, the high-seeding density required for correlation algorithms is often difficult to achieve with a simple holographic scheme. Our algorithm combines the strength of both tracking and correlation and uses holography as its basic recording scheme to transfer highly compact, instantaneous, three-dimensional flow field information. With the aid of the robust Genetic Algorithm, 3D velocity fields are extracted reliably with high spatial resolutions at practical speeds.

To have more specific measures, we compare the performance of GAPP and 3D correlation for a typical 3D flow. With the hologram-recorded trapped vortex, however, we found that the 3D correlation method produced excessive "bad" vectors due to low seeding density. Hence, to evaluate processing speed of the two algorithms, we used the simulated flow near a viscous rotating disk. In a flow region of 6 cm × 6 cm × 6 cm "seeded" with 10 μm particles at a density of 5 particles/mm^3, we applied both GAPP and 3D cross-correlation to extract the velocity field. The entire flow field was divided into 60 × 60 × 60 ICs. The 3D cross-correlation algorithm was implemented by 64 × 64 × 64 FFT. For the purpose of comparison, both algorithms were tested on a single Pentium Pro Processor (180 MHz) of a dual-processor PC running WindowsNT 4.0. Table 2 highlights the performance comparison of these two methods. Figure 16 shows the velocity components extracted

Table 2. Performance comparison of 3D correlation method and Particle Pairing using Genetic Algorithm for the flow near a rotating disk with the dimension of 6 cm × 6 cm × 6 cm

Features	3D correlation	GAPP
Total vectors	2.16×10^5	1.08×10^6
Spatial resolution	1 vector(s)/mm^3	5 vector(s)/mm^3
Total calculation time	2450 h[(1)]	10 h
Calculation time per IC	4 s	100 ms
Memory consumption per IC	8 Gbytes[(2)]	<1 kbytes

Note: [(1),(2)] Total calculation time and memory consumption can be greatly reduced if correlation is performed on particle centroids instead of on particle images. This new correlation approach is a part of our on-going research. The main advantage of GAPP is its intelligent processing in situations that correlation cannot handle, such as sparse seeding in complex flow fields

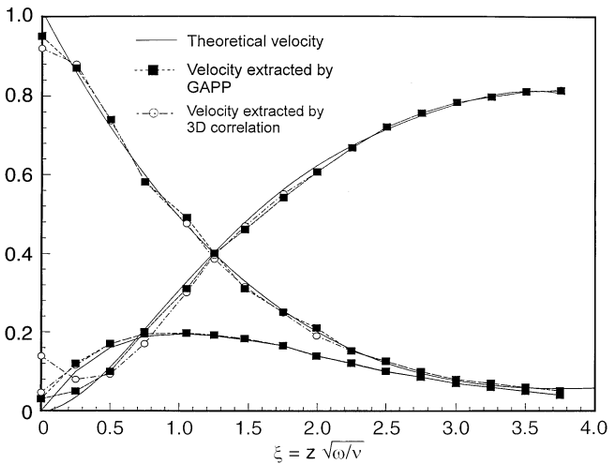

Fig. 16. Velocity fields extracted by 3D correlation and particle pairing methods compared with simulation result

by GAPP and 3D correlation methods as compared to the simulated rotating disk flow, as functions of the similarity variable ξ. Results show that, with correlation method more "bad" vectors were generated near the disk surface (especially visible on the radial component), making up to 4% of the total vectors. It must be pointed out that this flow does not have as large velocity gradients as most turbulent and vortical flows in real applications and hence the reduction of bad vector by GAPP is not obvious. However, it suffices for evaluation and comparison of the algorithm speed.

5 Conclusions

We have proposed a novel velocity extraction technique, GAPP, designed for 3D volumetric Holographic PIV data processing. Conventional PIV data processing methods based on correlation face the problem of inaccurate velocity representation in low seeding density regions and large velocity gradients. A 3D PIV technique such as a relatively simple holographic PIV scheme (e.g. IROV) presents a much more severe challenge to correlation methods than planar PIV, since the particle density allowed in 3D imaging volume is fundamentally limited. A promising solution for velocity field extraction in this case is particle pairing. We have shown that pairing particle images between exposures separated by a small Δt not only is viable but also offers higher accuracy and higher spatial resolution in 3D velocity extraction than correlation.

Without the aid of statistics, particle pairing must rely on good judgment. We have proposed two evaluation functions, based on continuity equation of incompressible flow and morphology conservation of particle groups, to filter and search for the best pairing schemes. Both evaluation functions are shown to be reliable and effective in our GAPP method. However, such a search is usually a lengthy process. With its inherent efficiency in searching a large solution space, Genetic Algorithm (GA) therefore becomes the implementation engine for particle pairing. To further increase the speed, a set of new genetic operators are specially designed including fast initialization, dynamic mutation, and order-based crossover. All these developments make GAPP not only accurate but fast as well. The GAPP algorithm is tested in HPIV measurement and a series of numerical experiments. Results show that compared to 3D correlation, GAPP gives more accurate velocity measurement in low seeding density regions and large velocity gradients in 3D space as well as higher processing speed.

Since Holographic PIV handles a massive amount of data, processing speed of GAPP is critical. Tests of sequential implementation of GAPP yield encouraging performance. Furthermore, its inherent parallelism and scalability make GAPP a promising practical alternative to correlation techniques for HPIV data processing.

References

Beck SJ; Lee SJ (1996) New two-frame particle tracking algorithm using match probability. Exp Fluids 22: 23–32

Barnhart DH; Adrian RJ; Meinhart CD; Papen GC (1994) Phase-conjugate holographic system for high-resolution particle image velocimetry. Appl Opt 33: 7159–7170

Chipperfield AJ; Fleming PJ (1994) Parallel genetic algorithms: a survey. ACSE Research Report No. 518, University of Sheffield

Cowen EA; Monismith SG (1997) A hybrid digital particle tracking velocimetry technique. Exp Fluids 22: 199

Davis L (1991) Handbook of genetic algorithm. Van Nostrand Reinhold, New York

Gray C; Greated CA (1993) Processing system for the analysis of particle displacement hologram, Proc SPIE Conf Optical Diagnostics and Fluid and Thermal Flow, V 2005, San Diego, 1993

Goldberg DE (1989) Genetic algorithm in search, optimization, and machine learning. Addison–Wesley, Reading, MA

Goldberg DE (1994) Genetic and evolutionary algorithms come of Age. Commun ACM 37: 113–119

Guezennec YG; Brodkey RS; Kent JC (1994) Algorithms for fully automated three-dimensional particle tracking velocimetry. Exp Fluids 17: 209–213

Johnson DS (1990) Local optimization and the traveling salesman problem. M. S. Paterson (ed), Proc 17th Colloquium on Automata, Languages, and Programming, Lecture Notes in Computer Science, Vol. 443, pp 446–460, Springer, Berlin

Kasagi N; Nishino K (1991) Probing turbulence with three-dimensional particle-tracking velocimetry. Exp Thermal Fluids Sci 4: 601–610

Lester BP (1993) The art of parallel programming. Prentice-Hall, Englewood Cliffs, NJ, pp 44–47
Meinhart CD; Adrian RJ (1995) Measurement of the zero-pressure gradient turbulent boundary layer using particle image velocimetry. AIAA 95-0789 Wash. DC: AIAA
Meng H; Hussain F (1991) Holographic particle velocimetry: a 3D measurement technique for vortex interactions, coherent structures and turbulence. Fluid Dyn Res 8: 33–37
Meng H; Anderson WL; Hussain F; Liu DD (1993) Intrinsic speckle noise in in-line particle holography. J Opt Soc Am 10: 2046–2058
Meng H (1994) Development of holographic particle velocimetry techniques for three-dimensional vortical flows. Ph.D. Dissertation, University of Houston
Meng H; Hussain F (1995a) Instantaneous flow field in an unstable vortex ring measured by holographic particle velocimetry. Phys Fluid 7: 9–11
Meng H; Hussain F (1995b) In-line recording and off-axis viewing techniques for holographic particle velocimetry. Appl Opt 34: 1827–1840
Michalewicz Z (1992) Genetic Algorithm + Data Structures = Evolution Programs. Springer, New York
Okamoto K; Hassan YA; Schmidl WD (1995a) New tracking algorithm for particle image velocimetry. Exp Fluids 19: 342–347
Okamoto K; Schmidl WD; Hassan YA (1995b) Spring model tracking algorithm for three dimensional particle image velocimetry. Flow Visualization and Image Processing of Multiphase Systems
Petty CB; Leuze MR; Grefenstette JJ (1987) A parallel genetic algorithm. Proc 2nd Int Conf Genetic Algorithms: 155–161
Schichting H (1979) Boundary-layer theory, 7th Ed. McGraw-Hill, New York, 102–107
Simmons S; Meng H; Hussain F; Liu D (1993) Advances in Holographic Particle Velocimetry. In: Optical diagnostics in fluid and thermal flow, SPIE 2005, San Diego, July 11–16
Tanse R (1989) Distributed genetic algorithms. Proc 3rd Conf Genetic Algorithms, pp 434–439
Tsukiji T; Ohyama R; Kaneko K (1994) Particle tracking method using genetic algorithm and its application to electrically operated flow field. Proc FLUCOME '94, pp 81–86
Wernet MP (1995) Fuzzy logic particle tracking velocimetry. In: Optical diagnostics in fluid and thermal flow, SPIE 2005, pp 701–708
Wernet MP (1995) Fuzzy inference enhanced information recovery from digital PIV using cross-correlation combined with particle tracking. SPIE 2546, pp 54–64
Yamamoto F; Uemura T; Tian H; Ohmi K (1993) Three-dimensional PTV based on binary cross-correlation method (algorithm of particle identification). JSME Int J Series B: Fluids Thermal Eng 36: 279–284
Zhang J; Tao B; Katz J (1996) Three dimensional velocity measurements using hybrid HPIV. Proc 8th Int Symp on Appl of LASER Tech to Fluid Mech., Vol I, Lisbon July
Zimin V; Meng H; Hussain F (1993) An innovative holographic particle velocimeter: multibeam technique. Opt Lett 18: 1101–1103