OPTIMAL TRANSPORT FOR PARTICLE IMAGE VELOCIMETRY:
REAL DATA AND POST-PROCESSING ALGORITHMS

LOUIS-PHILIPPE SAUMIER, BOUALEM KHOUIDER AND MARTIAL AGUEH

Abstract.
Particle image velocimetry (PIV) is a method used to measure the velocity field of a fluid flow. Traditionally, cross correlation algorithms are used to retrieve the flow velocity. Recently, a new method for PIV based on the $L^2$ optimal mass transportation problem (OT-PIV), is introduced and analyzed in [Saumier, Khouider and Agueh, Optimal Transport for Particle Image Velocimetry, Comm. Math. Sci., Vol. 13, No. 1, 269–296, 2015] for the case of synthetically generated data and for particles of equal mass or brightness. Here, we extend the work of Saumier et al. (2015) to the case of particles with different masses and randomly seeded particles. More importantly, we compare the OT-PIV method with a typical cross-correlation algorithm for the case of real data. Using a combination of theory and numerical experiments, we demonstrate that in the presence of particles with different weights/brightness, the OT method is more accurate for the largest/brightest particles, and it is more faithful when the particles are far enough from each other. This makes it more suitable for the so-called particle tracking regime of PIV, i.e., when the seeding density is low. We demonstrate in particular that for low seeding densities, the OT method performs better than a typical cross-correlation algorithm. Based on these new results and the previous ones in Saumier et al. (2015), we propose a suite of post-processing algorithms for the OT-PIV method. It is found that the OT-PIV with post-processing performs very well on the synthetic and real data when compared to the cross-correlation method.

Key words. PIV, Optimal Transport, Numerical Method, Real Data, Post-Processing

AMS subject classifications. 35J96, 35Q35, 76M27

1. Introduction. Particle Image Velocimetry, or PIV, is a non-intrusive technique used to measure the velocity field of fluid flows in laboratory experiments [1, 14, 24]. Simply put, small particles are seeded in the fluid and are illuminated by a pulsing laser. The distribution of light scattered by these particles is recorded every time the laser pulses, thus creating a discrete sequence of images. The data generated is then analyzed by algorithms typically based on cross-correlation between successive images to retrieve the associated velocity field [10]. The resulting velocity field is deemed a good representation of the actual flow velocity, under the assumption that the particles are small enough so that their effect on the fluid is negligible. When the average distance between particles is much larger than the mean displacement, the technique is sometimes referred to as particle tracking velocimetry (PTV).

Motivated by recent improvements in the numerical resolution of the $L^2$ Monge-Kantorovich problem [2, 3, 5, 7, 16, 17], we introduced in [18] a new PIV method based on the optimal transport (OT) map (in the $L^2$ Monge-Kantorovich sense) as a tool to tackle the PIV problem. Every two successive PIV images are viewed as a pair of initial and final mass distributions in the OT problem, and the resulting transport flow at the particle centres is used to approximate the fluid velocity. We used a continuous model to represent the densities as opposed to considering them as discrete collections of Dirac distributions. The reasons behind this choice are mainly that a continuous problem better allows the tracking of structures created by coalesced groups of particles and that discrete assignment problems can be computationally challenging to solve. Indeed, if $N_p$ is the number of particles involved, then the discrete OT problem scales as $O(N_p^3)$ whereas the continuous method employed here
uses $O(N \log N)$ operations, where $N$ is the numerical grid size (more details are given in [18]). It should nonetheless be noted that $N$ increases rapidly with the dimension of the problem. Also, optimal transportation theory is used in image registration [8, 15], but before [18] it had never been applied to estimate velocity fields from PIV data.

To conduct theoretical analysis for the OT-PIV method in [18], we used a basic model of PIV-like images for synthetic data. More precisely, we approximate point particle (Dirac delta) distributions for the initial and final particle positions, when displaced by a subscribed flow field during a short period of time, by sums of Gaussian distributions with a fixed standard deviation $\sigma$ and an additive background noise constant $1 - r$. In [18], we derived an error bound for the particle final position as predicted by the transport map, for the case of a single particle. Namely, we showed that the error behaves as $O(\sigma(1 - r)/r)$ when $\sigma$ and $1 - r$ are small. We then demonstrated through a combination of theory and numerical experiments that the result can be extended to the case of many particles in 2 or more dimensions.

One shortcoming of these results however comes from the fact that, for the case of many particles, we assumed that all the particles have the same weight in the Gaussian sum density distribution. This is equivalent to assuming that all particles in the PIV image have the same brightness. This assumption is far from being realistic. In practice, particles can vary in size and shape, the light density is not necessarily uniform for every particle, and most importantly, in 2D PIV, the light scattered varies with the perpendicular distance to the observation plane. In addition, only the PTV regime is considered. Before applying the OT-PIV method to real data, it is important to understand its behavior outside this tracking range and in particular when the particles have different weights. To address these issues, we perform the synthetic data exercise for the case of many particles with different weights/brightness. More precisely we modify the Gaussian model [18] by introducing weight parameters $m_i, i = 1, \cdots, M$ (where $M$ is the total number of particles) to represent the relative contributions of individual particles in the total initial and final densities $f$ and $g$, respectively. In the case of $M = 2$, we show that the ratio of the OT-PIV errors associated with the two particles is inversely proportional to their weights. This new result indicates that, in terms of the approximated velocity field, the vectors associated with the brightest particles are more faithful. We also analyze the behavior of the transport map when two particles of different sizes initially far from each other cross paths. The results show that when one of the two particles is much brighter than the other, the flow velocity vector is faithfully retrieved for the brighter particle.

Here, we use this new understanding of the behavior of the OT map when multiple particles of different sizes are involved (inside or outside the tracking range), to propose a suite of post-processing algorithms or filters for the OT-PIV method. For instance, we illustrate the performance and analyze the behavior of the OT-PIV method in various parameter regimes via a few numerical experiments in the case of a synthetic shear flow. From [18], we know that the transport map does not always send the center of a Gaussian to its target center. We show in this work that the effect of this error on the approximated field can be dampened by reassigning the predicted target position to the brightest neighboring pixel. Also, in actual PIV images, the light scattered by a single particle is closer to a Gaussian distribution than it is to a Dirac mass. As shown in our experiments the mass associated with a single particle can be split and sent to two different locations by the transport map and can lead to significant approximation errors. In this paper, we show how the sequence of filters...
introduced here successfully correct the errors in the case of a synthetic shear flow and validate their effect by quantifying the associated actual errors for the case of a synthetic vortex. The OT-PIV method with these post-processing filters are then applied to two realistic sets of PIV images, one taken from the 2001 PIV challenge [13, 19], and the other from an actual laboratory experiment of a fluid flow [25]. Compared to the results obtained by a typical cross-correlation algorithm [20, 21], the method performs equally well for most cases and even better in a few others, especially when the seeding density is low.

The paper is organized as follows. In Section 2 we present a brief summary of the optimal transport theory and the main method used to approximate the velocity field of moving particles via optimal transport paths. In Section 3, we analyze the behavior of the transport map for two particles with weighting parameters $m_i$ representing the non-uniform particle brightness, within and outside the tracking range. In Section 4, we consider the case of synthetic data generated by both a simple shear and a vortex, and we introduce and validate two natural post-processing filters based on these two cases. Section 4 concludes with a systematic strategy to tackle real PIV images which is applied to two realistic sets of PIV images. Finally, we compare the OT method with cross-correlation at low seeding density in Section 5, and we give a few concluding remarks in Section 6.

2. Monge-Kantorovich Approximate Field. Let $f, g \in C^2(\Omega)$ be two mass densities, and consider Monge’s $L^2$ optimal transport (OT) problem:

$$
\text{Min} \int_{\Omega} [T(x) - x]^2 f(x) \, dx,
$$

over the set of all transformations $T$ pushing $f$ forward to $g$: $T\#f = g$ [12, 22, 23]. According to [4], when $f$ and $g$ are sufficiently smooth, the map $T$ in (2.1) takes the form $T = \nabla \Psi$, where $\Psi$ is a convex solution of the Monge-Ampère equation:

$$
g(\nabla \Psi(x)) \det(D^2 \Psi(x)) = f(x).
$$

In the context of PIV, it is natural to consider $f$ as the image of the light scattered back by particles at time $t = 0$ and $g$ as the corresponding image at time $\Delta t$, where $\Delta t$ is the time elapsed between the two successive images. The optimal map $T$ might provide a good approximation of the particle trajectories provided $\Delta t$ is small enough, and thus it could be used to approximate the flow velocity. It is well known in the optimal transport theory that the time dependent map

$$
T_t(x) = x + \frac{t}{\Delta t} (T(x) - x), \quad 0 \leq t \leq \Delta t
$$

is the $L^2$ optimal map which sends the density $\rho_0 = f$ to a density $\rho_t$: $T_t\#f = \rho_t$ with $\rho_{\Delta t} = g$ [22]. The associated velocity field is then constant on $[0, \Delta t]$: $v = (T(x) - x)/\Delta t$.

Moreover, it can be shown that the couple $(\rho_t, v)$ satisfies the following equations:

$$
\begin{align*}
\frac{\partial v}{\partial t} + (v \cdot \nabla)v &= 0, \\
\frac{\partial \rho_t}{\partial t} + \text{div}(\rho_t v) &= 0,
\end{align*}
$$

which describe a pressureless gas at zero temperature [22]. For the time period $[0, \Delta t]$, the approximate flow given by Optimal Transport is a pressureless potential flow,
\[ v = \nabla \left( \Psi(x) - |x|^2 \right) / \Delta t \] and the associated trajectories are straight lines. Therefore, when using OT to approximate the velocity field for PIV, the associated trajectories, describing the intermediate particle positions during the time period \( \Delta t \), are in general not realistic. In the more specific case of a potential flow, the approximate trajectories might be closer to the target trajectories, provided the effect of the pressure is not strong. However, even though the OT trajectories are not physically accurate (in general), this method gives good results when it comes to predicting the final positions of the brightest particles, as it is demonstrated in this work and previous work [18]. Indeed, for most particles, the position after \( \Delta t \) units of time is well approximated by \( T(x) \).

3. Gaussian OT Model for PIV. Even though the OT map provides a good approximation of the final positions of particles, there are several sources of error that need to be analyzed. To do so, we will assume as in [18] that the image of the light scattered back by the particles is close to a Gaussian distribution. This is a realistic assumption motivated by the fact that the light scattered back by a point source and captured by a lens is distributed according to the Airy function, which itself is very close to a Gaussian [14]. We therefore select as a model for PIV images sums of weighted Gaussian distributions with standard deviation \( \sigma \), additive noise \( 1 - r \) and weights \( m_i \). Let \( \lambda \) and \( \mu \) be respectively the initial and final positions of a particle. Here, we are interested in the behavior of \( |T(\lambda) - \mu| \). While the effects of the parameters \( \sigma \) and \( r \) were studied in [18] for the case of a single particle, here, we will consider the case of two distinct particles which is more appropriate for the PIV problem. Also, because the initial distance between the two particles is an important parameter, we will consider both the cases when this distance is larger than the flow displacement (case within the tracking range) and the case where this constraint is violated (case outside the tracking range).

3.1. Case of Two Particles within Tracking Range. Consider for now two particles in 1D. We are able to rigorously analyze the one-dimensional case because the transport map can be obtained explicitly in this scenario (the result can be extended as a conjecture to higher dimensions and as in [18] it has been readily validated in the case of 2 dimensions via numerical experiments, results not shown here). Also, due to the nature of the transport problem, this analysis for two particles extends to the case of multiple particles more relevant to the PIV problem. Therefore, let the domain be \( \Omega = [0, 1] \) and the centers of the particles be respectively at \( \lambda_1 \) and \( \lambda_2 \). Let them evolve and arrive at positions \( \mu_1 \) and \( \mu_2 \) in \( \Omega = [0, 1] \) after \( \Delta t \) units of time. We first consider the case for which the particles stay within the tracking range, that is \( \lambda_1, \mu_1 \) are close to each other and far from \( \lambda_2, \mu_2 \) so that the OT map may not mistakenly select \( \mu_2 \) as the image of \( \lambda_1 \) and vice-versa. Let the initial mass density be given by

\[
f(x) = \frac{m_1 f_1(x) + m_2 f_2(x)}{m_1 + m_2},
\]

\[
f_i(x) = \frac{r}{k_{\lambda_i}} e^{-\frac{(x-\lambda)^2}{2\sigma^2}} + (1 - r), \quad k_{\lambda_i} = \int_0^1 e^{-\frac{(x-\lambda)^2}{2\sigma^2}} \, dx \quad i = 1, 2.
\]

Here, \( \sigma > 0 \) is the standard deviation, \( 0 < r < 1 \) is the noise factor and \( 0 < m_1, m_2 \) are the weights of particles one and two. Similarly, define the final density \( g_i \) by replacing \( f_i \) by \( f_i \) and \( \lambda_i \) by \( \mu_i \). We will establish an estimate for the ratio of the corresponding transport errors, \( |T(\lambda_1) - \mu_1|/|T(\lambda_2) - \mu_2| \), in terms of the particle masses \( m_1 \) and \( m_2 \). Note that in practice, the weights \( m_1 \) and \( m_2 \) in
the initial distribution \( f \) may be different from the ones in the final distribution \( g \). However, it is not unrealistic to assume that the weight differences between the initial and final densities remain small if \( \Delta t \) is small enough, which explains the equality assumption. This in principle does not depend on whether or not the particles are inside or outside the tracking range as long as the flow considered is laminar enough (a necessary assumption for 2D-PIV).

**Theorem 3.1.** Let \( T \) be the OT solution mapping the previously defined one-dimensional density \( f \) to its target \( g \). Fix \( 0 < r < 1, 0 < \epsilon < \lambda_1, \mu_1 < \lambda_2, \mu_2 < 1 - \epsilon < 1 \) for \( \epsilon > 0 \), where \( \lambda_1 \neq \mu_1, \lambda_2 \neq \mu_2 \) and

\[
\max_{i=1,2} |\lambda_i - \mu_i| < \min \left( \min_{1 \leq i \neq j \leq 2} |\lambda_i - \mu_j|, |\lambda_1 - \lambda_2|, |\mu_1 - \mu_2| \right)
\]

so that the particles remain far from each other. Assume that the weights \( m_1, m_2 > 0 \). Then, for \( \sigma > 0 \) small enough, we have

\[
\frac{|\mu_1 - T(\lambda_1)|}{|\mu_2 - T(\lambda_2)|} = O \left( \frac{m_2}{m_1} \right).
\]

**Proof.** The beginning of this proof is similar to the case of one particle in dimension one [18] and thus we will simply summarize the initial arguments. Let \( T(x) \) be the \( L^2 \) OT map sending \( f \) to \( g \). Then \( T : [0,1] \rightarrow [0,1] \) is increasing, bijective, and satisfies \( g(T(x))T'(x) = f(x) \). Integrating this relation over \([0,x] \), we get

\[
G(T(x)) - G(T(0)) = F(x),
\]

where \( F \) and \( G \) are the cumulative distributions functions of \( f \) and \( g \):

\[
F(x) = \frac{1}{m_1 + m_2} \left( \frac{m_1 r}{k_{\lambda_1}} \int_0^x e^{-\frac{(t-\lambda_1)^2}{2\sigma^2}} dt + \frac{m_2 r}{k_{\lambda_2}} \int_0^x e^{-\frac{(t-\lambda_2)^2}{2\sigma^2}} dt \right) + (1-r)x,
\]

and \( G(x) \) is defined similarly. This implies that \( G(T(0)) = G(0) = 0 \) and \( G(T(x)) = F(x) \). Using \( x = \lambda_1 \) in the previous relation and setting \( y = T(\lambda_1) \) leads to \( G(y) = F(\lambda_1) \). Consider now the following Taylor expansion with integral remainder:

\[
\frac{1}{k_{\mu_1}} \int_0^y e^{-\frac{(t-\mu_1)^2}{2\sigma^2}} dt = \frac{1}{k_{\mu_1}} \int_0^{\mu_1} e^{-\frac{(t-\mu_1)^2}{2\sigma^2}} dt + \frac{(y - \mu_1)}{k_{\mu_1}} \int_0^1 e^{-\frac{(2(y-\mu_1)^2)}{2\sigma^2}} dt.
\]

Using this expansion in \( G(y) = F(\lambda_1) \) and adding \( \mu_1(m_1 + m_2)(1-r)/r \) on both sides, we get,

\[
\left[ \mu_1 - T(\lambda_1) \right] \left( \frac{m_1}{k_{\mu_1}} \int_0^1 e^{-\frac{r^2(T(\lambda_1) - \mu_1)^2}{2\sigma^2}} dt + \frac{(1-r)}{r} (m_1 + m_2) \right)
\]

\[
= m_1 \left( \frac{1}{k_{\mu_1}} \int_0^{\mu_1} e^{-\frac{(t-\mu_1)^2}{2\sigma^2}} dt - \frac{1}{k_{\lambda_1}} \int_0^{\lambda_1} e^{-\frac{(t-\lambda_1)^2}{2\sigma^2}} dt \right) + \frac{(1-r)}{r} (m_1 + m_2)(\mu_1 - \lambda_1)
\]

\[
+ m_2 \left( \frac{1}{k_{\mu_2}} \int_0^{T(\lambda_1)} e^{-\frac{(t-\mu_2)^2}{2\sigma^2}} dt - \frac{1}{k_{\lambda_2}} \int_0^{\lambda_1} e^{-\frac{(t-\lambda_2)^2}{2\sigma^2}} dt \right).
\]

We observe as in Lemma 3.2 in [18], that the difference

\[
\epsilon_{11} = \frac{1}{k_{\mu_1}} \int_0^{\mu_1} e^{-\frac{(t-\mu_1)^2}{2\sigma^2}} dt - \frac{1}{k_{\lambda_1}} \int_0^{\lambda_1} e^{-\frac{(t-\lambda_1)^2}{2\sigma^2}} dt =: I(\mu_1) - I(\lambda_1)
\]

is small enough, which explains the equality.
decays exponentially in \( \sigma \), provided \( \lambda_1, \mu_1 \) are far from the boundary (both integrals account for roughly half the mass under the exponentials). Indeed, by the mean value theorem, there exists \( \xi \) between \( \lambda_1 \) and \( \mu_1 \) such that \( I(\mu_1) - I(\lambda_1) = I'(\xi)(\mu_1 - \lambda_1) \).

By explicitly computing \( I' \), centering the Gaussian at 0 instead of \( \xi \) in the resulting denominator and then using a change of variable in the integral, we have

\[
I'(\xi) \leq \frac{e^{-\frac{\xi^2}{2\sigma^2}} \int_\xi^{+} e^{-(\frac{\xi^2}{2\sigma^2})} d\xi + e^{-\frac{\xi^2}{2\sigma^2}} \int_{-\xi}^{0} e^{-(\frac{\xi^2}{2\sigma^2})} d\xi}{\left( \int_0^{1} e^{-(\frac{\xi^2}{2\sigma^2})} d\xi \right)^2} \leq \frac{e^{-\frac{\xi^2}{2\sigma^2}}}{\sigma \int_0^{\frac{\xi}{\sigma}} e^{-(\frac{\xi^2}{2\sigma^2})} d\xi} = \frac{e^{-\frac{\xi^2}{2\sigma^2}}}{\sigma \xi}.
\]

This proves the exponential decay in \( \sigma \) for \( \sigma \) small enough, as claimed. Next, the difference

\[
\epsilon_{12} = \frac{1}{k_{\mu_2}} \int_0^{T(\lambda_1)} e^{-\frac{(t-\mu_2)^2}{2\sigma^2}} dt - \frac{1}{k_{\lambda_2}} \int_0^{\lambda_1} e^{-\frac{(t-\lambda_2)^2}{2\sigma^2}} dt
\]

also decays exponentially in \( \sigma \) (the Gaussians are integrated over their tails). More specifically, because the integrands are increasing functions of \( t \), we have

\[
|\epsilon_{12}| \leq \left[ T(\lambda_1) \frac{e^{-\frac{(T(\lambda_1)-\mu_2)^2}{2\sigma^2}}}{k_{\mu_2}} + \lambda_1 \frac{e^{-\frac{(\lambda_1-\lambda_2)^2}{2\sigma^2}}}{k_{\lambda_2}} \right].
\]

Then, by assumption (3.1) and the optimality of the transport map \( T_0 \), \( T(\lambda_1) \) will be close to \( \mu_1 \) which is itself far from \( \mu_2 \), we get \( |T(\lambda_1) - \mu_2| > |\lambda_2 - \mu_2| \) and \( |\lambda_1 - \lambda_2| > |\lambda_2 - \mu_2| \) (for example), which in turn yields

\[
|\epsilon_{12}| \leq e^{-\frac{(\lambda_2-\mu_2)^2}{2\sigma^2}} \frac{2}{k_{\mu_2}^2} - e^{-\frac{(\lambda_2-\mu_2)^2}{2\sigma^2}} \frac{2}{k_{\lambda_2}^2}.
\]

This shows that \( \epsilon_{12} \) decays exponentially in \( \sigma \) for \( \sigma \) small enough. By applying the same technique, we obtain an identity similar to (3.2) for \( \mu_2 - T(\lambda_2) \) and the same observations apply to the dominating terms. Gathering these estimates, we have

\[
\frac{\mu_1 - T(\lambda_1)}{\mu_2 - T(\lambda_2)} = \left( \frac{m_1 \epsilon_{11} + \frac{(1-r)}{r}(m_1 + m_2)(\mu_1 - \lambda_1) + m_2 \epsilon_{12}}{m_1 \epsilon_{21} + \frac{(1-r)}{r}(m_1 + m_2)(\mu_2 - \lambda_2) + m_2 \epsilon_{22}} \right) \times \left( \frac{m_2}{m_1} \right) \left( \frac{k_{\lambda_1}}{k_{\mu_2}} \right) \left( \int_0^{1} e^{-\frac{r(T(\lambda_2)-\mu_2)^2}{2\sigma^2}} dt + k_{\mu_2}(1-r) \frac{m_1 + m_2}{m_2} \right)
\]

where \( \epsilon_{21} \) and \( \epsilon_{22} \) are defined similarly to \( \epsilon_{11} \) and \( \epsilon_{12} \). Notice that the ratio \( k_{\mu_1}/k_{\mu_2} \approx 1 \) for small \( \sigma \). Moreover, we know that \( T(\lambda_1) - \mu_1 \) decays linearly in \( \sigma \) in the case of one particle [18]. Here, since the two particles are assumed to be far from each other, then the properties of the OT map will preserve this decay, and thus there exist constants \( C_1 > 0 \) and \( C_2 > 0 \) such that

\[
(3.3) \quad \int_0^{1} e^{-\frac{r(T(\lambda_2)-\mu_2)^2}{2\sigma^2}} dt \approx \int_0^{1} e^{-C_{1}t^2} dt \quad \text{and} \quad \int_0^{1} e^{-\frac{r(T(\lambda_1)-\mu_1)^2}{2\sigma^2}} dt \approx \int_0^{1} e^{-C_{1}t^2} dt.
\]

From (3.3) and the decay in \( \sigma \) of the \( \epsilon_{ij} \), there exists \( C_3 > 0 \) such that

\[
\frac{|\mu_1 - T(\lambda_1)|}{|\mu_2 - T(\lambda_2)|} \leq C_3 \frac{|\mu_1 - \lambda_1| m_2}{|\mu_2 - \lambda_2| m_1} \left( \int_0^{1} e^{-C_{1}t^2} dt + k_{\mu_2}(1-r) \frac{m_1 + m_2}{m_2} \right) \left( \int_0^{1} e^{-C_{1}t^2} dt + k_{\mu_1}(1-r) \frac{m_1 + m_2}{m_1} \right)
\]
Finally, using that both $k_{\mu_1}$ and $k_{\mu_2}$ are at least $O(\sigma)$, we conclude that

$$
\frac{|\mu_1 - T(\lambda_1)|}{|\mu_2 - T(\lambda_2)|} \leq C_4 \frac{|\mu_1 - \lambda_1|}{|\mu_2 - \lambda_2|} \frac{m_2}{m_1} \left( \int_0^1 e^{-C_1 t^2} dt \right) \leq C_5 \frac{m_2}{m_1}
$$

for appropriate constants $C_4, C_5$, and for $\sigma$ small enough.

![Images of particles]

**Fig. 1.** Results for the OT method applied to two particles for different values of their respective weights. The resulting field obtained via our OT algorithm is displayed at the particles’ initial locations.

### 3.2. Case of Two Particles outside Tracking Range.

Let us now consider the case where the two particles evolve in a flow so that the final position of each particle is closer to the other particle’s initial position (we are thus outside the tracking range). If the masses of the two particles $m_1$ and $m_2$ are the same, then we expect the OT solution to mismatch the particles and give the wrong vectors. However, this is not necessarily the case if the weights are not the same. In order to see this, we computed the solution of the OT problem in the case of a crossover between two 2D particles with initial positions $\lambda_1 = (0.35, 0.4), \lambda_2 = (0.65, 0.6)$ and final positions $\mu_1 = (0.65, 0.4), \mu_2 = (0.35, 0.6)$, respectively. The results are presented in Figure 1. We see that the error in this regime is also smaller for brighter particles. These results and the ones of the previous section will be used as a principle to discard the vectors associated with the small particles (in terms of brightness), i.e. particles which do not carry enough weight in the OT initial and final distributions.


In this section we present the results of numerical experiments demonstrating some strengths and weaknesses of the OT method for PIV and PTV. We begin by analyzing situations where the data are given analytically by our Gaussian model, and use the results to gain intuition and introduce a series of post-processing strategies in order to improve the accuracy of the retrieved velocity field. Then we present experiments of actual PIV images and compare our method with a typical cross-correlation algorithm [20, 21]. This algorithm is based on successive applications of the cross-correlation on interrogation windows displaced at every pass, which is typical of most traditional cross-correlation based PIV methods [14]. For all our numerical experiments, we use the procedure given in [17] to solve the OT problem.

#### 4.1. Synthetic Data.

**4.1.1. Case of Shear Flow.** While the detailed trajectories and flow properties of the OT map and those associated with the actual flow field are both available for synthetic examples, the use of the L^2 OT method to recover an approximate representation of the latter aims at precisely matching the initial position with the final position of the particles given by successive PIV images and not per se predicting the
trajectories of the particles. In fact, the OT and the fluid flow can behave very differently even in cases where the effect of pressure can be neglected. Here we illustrate this fact through the simple example of a shear flow.

Consider the Eulerian equations satisfied by the OT time-dependent density $\rho$ and velocity $v$:

$$\begin{align*}
\frac{\partial v}{\partial t} + (v \cdot \nabla)v &= 0 \\
\frac{\partial \rho}{\partial t} + \text{div}(\rho v) &= 0.
\end{align*}$$

Notice that a constant shear (of the form $v(x_1, x_2, t) = (\gamma(x_2), 0)$ for example, which is by construction a solution to the incompressible Euler equations for fluid flows) satisfies the first equation in the system. The second equation just becomes

$$\frac{\partial \rho}{\partial t} + \gamma(x_2) \frac{\partial \rho}{\partial x_1} = 0$$

and signifies that the initial density is simply transported horizontally according to $\gamma(x_2)$. However, since the optimal map $T$ is known to be the gradient of a convex function, we also know that $v$ is a potential flow and thus cannot accurately represent a shear flow. Therefore, it is not possible with the OT method to go beyond the simple first-order accurate in time approximation corresponding to the matching problem.

Let the domain $\Omega = [0, 1]^2$ and the velocity field be defined by

$$v(x_1, x_2, t) = \begin{cases} 
(\gamma, 0) & \text{if } x_2 \geq 0.5 \\
(-\gamma, 0) & \text{if } x_2 < 0.5
\end{cases}$$

for a given constant parameter $\gamma$. This field is obviously not continuous, but we will use it to approximate a continuous horizontal shear with a sharp transition between the two interfaces. Consider the initial distribution of particles $f(x_1, x_2) = \rho(x_1, x_2, 0)$ to be a sum of Gaussian distributions evenly distributed in a region of $[0, 1]^2$ far enough from the boundaries and let the corresponding final density be given by $g(x_1, x_2) = \rho(x_1, x_2, \Delta t)$, representing the same sum of Gaussian distributions but for the particles displaced by the shear flow above after the time period $\Delta t$. We can observe from Figure 2 the result of the OT algorithm applied to such two densities. We see that if the particles do not move further than the tracking regime (flow displacement smaller than mean distance between particles) and if we stay far enough from the shear’s transition zone, then the OT algorithm recovers almost perfectly the given vector field. Note that this remains true for the “in between” regime, that is when the final positions are exactly in between two initial positions (Figure 2 b). When the shear is stronger and for the same time interval, particles exit the tracking range, and mismatches occur almost everywhere. Actually, a vertical transfer of mass now occurs at the edges since the vertical distance between particles is now smaller than the horizontal distance.

Let us now make two observations on the errors made by the method for the “in between” regime of Figure 2 b). First, the least accurate vectors are located at the shear’s transition zone (especially at the edges). At these positions, the optimal solution to the $L^2$ Monge problem actually splits the mass of the Gaussians into pieces, distributing it towards neighboring target positions of particles. Because these masses correspond to physical tracers, this solution does not make physical sense. It would
Fig. 2. Results for the OT method applied to the shear flow for different values of the shear strength $\gamma$. The full circles represent the initial positions of particles whereas the transparent circles represent their final positions. The resulting field obtained via our OT algorithm is displayed at the particles’ initial locations.

Fig. 3. The No Split filter applied to experiment b) in Figure 2, for different values of $k_{ns}$ (strength of filter).

therefore be appropriate to remove some of these vectors from the final approximation (at the very least the ones at the edges).

In addition, we also observe in Figure 2 b) that there is an error even in the vectors far from the shear’s transition zone. These vectors’ directions are quite accurate, but their length is always smaller than the lengths of the target vectors. This is due to the OT map which does not send the center of a Gaussian directly to its target center in the tracking range, as seen in Section 3. In this particular case, this error can be corrected by simply assigning the image of this center to the brightest neighboring pixel. Let us now present two post-processing filters designed to reduce these errors.

4.1.2. No Split Filter. As previously mentioned, the shear experiment shows that the OT solution will sometimes split the mass associated to one particle into several parts to be distributed among neighboring particles. This observation naturally leads to a post-processing filter where we identify and remove the vectors associated to a particle whose mass is being split.

**No Split Filter**

Let $\lambda_i$ be the initial position of a particle on the numerical grid, and let $\Lambda_{\lambda_i}$ be the set of all 8 neighboring pixels. Then, if there exists $x \in \Lambda_{\lambda_i}$ for which

$$k_{ns}|x - \lambda_i| < |T'(x) - T'(\lambda_i)|,$$

the vector $v_i = \frac{T'(\lambda_i) - \lambda_i}{\Delta t}$ is rejected.
The positive parameter $k_{ns}$ determines the strength of the filter and is typically taken in between 1 and 10. Note here that mass splitting is possible because we are considering continuous densities (not to be confused with mass splitting of Dirac Delta distributions, which is not allowed by the push-forward constraint imposed on the transport problem). Figure 3 shows the effect of this filter on experiment b) displayed in Figure 2. We see on this figure that this simple filter effectively removes spurious vectors for the shear experiment. We shall also observe its effectiveness on real examples in the subsequent sections.

4.1.3. Particle Target Position Correction. The other natural post-processing filter motivated by previous observations consists in changing the predicted target position of tracers to the position corresponding to the brightest surrounding pixel.

**Particle Target Position Correction Filter**

If $\lambda$ is the position of a bright particle and $T(\lambda)$ its predicted final position and if $\Gamma$ is the set of all pixels close to $T(\lambda)$, in the sense that

$$|y - T(\lambda)|_1 < c$$

for $y \in \Gamma$ and $c$ a constant, then, we select $y \in \Gamma$ such that the final density $g(y)$ is the greatest and we assign it to the final position of the particle at $\lambda$.

We therefore correct the errors created by the OT procedure which does not map the center of the initial Gaussian distribution exactly to the center of the target Gaussian (provided the right assignment is achieved). Let us first test this strategy on the example b) of Figure 2. The results are presented in Figure 4 for $c = 40$ (the total image is 512 by 512 pixels). We see that this correction eliminates the error made by the transport map for every particle except one. For that tracer, the brightest neighboring pixel to the image by the transport map does not correspond to the same initial tracer and the algorithm leads to a mismatch. This error will be more frequent when a larger number of tracers will be considered in the next experiments. However, in this specific case, we can already eliminate this vector with the no split filter as previously shown.

4.1.4. Case of a Vortex. In order to further validate the effect of the filters we presented, we now take a fixed number of Gaussian-like particles of the type presented in 3, with equal masses randomly distributed in $\Omega$. Note that scenarios where particles have different masses will be considered in the next sections. Assume that these particles evolve according to the velocity field given by

$$v(x_1, x_2) = \left(x_2 - 0.5, 0.5 - x_1\right) \exp\left(-10((x_1 - 0.5)^2 + (x_2 - 0.5)^2)\right)$$
OT FOR PIV: REAL DATA AND POST-PROCESSING

(a) Field obtained with a perfect match (the only error is due to finite differences).

(b) Field obtained with the transport algorithm.

(c) OT field with the No Split filter and $k_{ns} = 5$.

(d) OT field with the particle end-position correction and $c = 3$.

(e) Field obtained with the DCC algorithm superposed to the initial density.

(f) Field obtained by filtering the field in e) with a velocity threshold set at 2 standard deviations away from the mean velocity in either components.

We then use their final positions after $\Delta t = 0.25$ units of time to create a second distribution with Gaussians. The initial and final densities are thus taken as $f$ and $g$ (200 tracers were seeded to create the densities). We applied the transport algorithm with different filters on these initial and final images and the results are presented in Figures 5 a), b), c) and d).

The velocity field a) shows the result of a perfect match between initial and final positions, i.e. the only error is the error due to the finite differences approximation. In b), we see the field obtained with the OT algorithm. The initial position of particles was not assumed to be known, and every pixel was considered to be a particle if it was brighter than a certain brightness threshold (0.8 in this case). This simple strategy to detect particles will be explained in more details in the next sections. By doing so, we obtain a vector for 198 of the 200 particles initially seeded. Note that some particles are not individually detected because they are too close to another particle. The field c) shows the result obtained by applying the No Split filter to field b) with the parameter $k_{ns} = 5$ (30 vectors are removed). Finally, we see in d) the field given by applying the particle end position correction filter to b).

To quantify the improvement done by the two filters considered, let us first compare the $l^2$ norms of the errors between the fields obtained and the “perfect match”
field. More specifically, we look at

$$|e|_2 = \frac{1}{N_p} \sqrt{\sum_{i=1}^{N_p} |e_i|^2}$$

where $|e_i|^2 = (v_{OT_1}^i - v_1^i)^2 + (v_{OT_2}^i - v_2^i)^2$ is the $l^2$ error between the $i$-th OT vector $v_{OT}^i = (v_{OT_1}^i, v_{OT_2}^i)$ and the $i$-th target vector $v^i = (v_1^i, v_2^i)$, and $N_p$ is the number of particles considered. The results are presented in Table 1. We see that the No Split filter effectively removes some bad vectors. In addition, even if the particle end-position correction filter does not improve the quality of every single vector (the $l^\infty$ error significantly), it decreases the global $l^2$ error significantly. To look more specifically at the effect of the parameters on the $l^2$ error, we present in Table 2 this error for different values of the parameters $k_{ns}$ and $c$, but this time for the exact field given by 4.3 and not the finite differences field. We see that the No Split filter significantly reduces the $l^2$ error, at the expense of removing more and more vectors. On the other hand, the particle end position correction filter does not improve the error as much, but does not remove any vectors. It has a greater effect when the No Split filter is not too strong, but when the value of $c$ is greater than 9, the $l^2$ error starts increasing in some cases due to mismatches. Therefore, a mild application of both filters is desirable.

We can also compare these errors with the ones obtained using a cross-correlation algorithm, namely the Direct Cross-Correlation (DCC) algorithm of the PIVlab toolbox in MATLAB [20, 21]. Results of this algorithm for the images under study in this section are presented in Figures 5 e) and f). Table 3 gives the $l^2$ and $l^\infty$ errors in between the exact field (4.3) and the images of Figures 5 e) and f). The reader should keep in mind that both methods give rise to different numerical grids, and thus the norms are computed on their respective grids and should be considered as approximations of the underlying continuous $L^2$ and $L^\infty$ norms. However, for interpolation purposes, it is worth noticing that the DCC algorithm needs a coarser grid than the OT algorithm to achieve a similar error for the current experiment. Another observa-

### Table 1

$l^2$ and $l^\infty$ norms of the error between the fields displayed in Figures 5 a), b), c) and d). $N_p$ represents the number of vectors used in each case.

<table>
<thead>
<tr>
<th>Fields Considered</th>
<th>$l^2$ Error</th>
<th>$l^\infty$ Error</th>
<th>$N_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite Differences and OT</td>
<td>$4.92 \times 10^{-3}$</td>
<td>$3.37 \times 10^{-2}$</td>
<td>198</td>
</tr>
<tr>
<td>Finite Differences and OT with $k_{ns} = 5$</td>
<td>$4.51 \times 10^{-3}$</td>
<td>$2.90 \times 10^{-2}$</td>
<td>168</td>
</tr>
<tr>
<td>Finite Differences and OT with $c = 3$</td>
<td>$4.40 \times 10^{-3}$</td>
<td>$3.53 \times 10^{-2}$</td>
<td>198</td>
</tr>
</tbody>
</table>

### Table 2

$l^2$ errors between the exact field (4.3) and the OT field for the given parameter values. $N_p$ represents the number of vectors used in each case.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>No Filter</th>
<th>$k_{ns}$</th>
<th>5.0</th>
<th>3.0</th>
<th>1.5</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$5.4 \times 10^{-3}$</td>
<td>$5.2 \times 10^{-3}$</td>
<td>$4.6 \times 10^{-3}$</td>
<td>$4.4 \times 10^{-3}$</td>
<td>$4.2 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$5.3 \times 10^{-3}$</td>
<td>$5.1 \times 10^{-3}$</td>
<td>$4.5 \times 10^{-3}$</td>
<td>$4.3 \times 10^{-3}$</td>
<td>$4.2 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>$5.2 \times 10^{-3}$</td>
<td>$5.0 \times 10^{-3}$</td>
<td>$4.4 \times 10^{-3}$</td>
<td>$4.3 \times 10^{-3}$</td>
<td>$4.2 \times 10^{-3}$</td>
<td></td>
</tr>
<tr>
<td>$N_p$</td>
<td>198</td>
<td>168</td>
<td>119</td>
<td>101</td>
<td>31</td>
<td></td>
</tr>
</tbody>
</table>
Table 3

$l^2$ and $l^\infty$ norms of the error between the fields displayed in Figures 5 e) and f). $N_p$ represents the number of vectors used in each case. For the DCC algorithm to be effective, a reasonable number of particles per window needs to be used and thus less vectors are produced when compared to the OT-PIV method.

<table>
<thead>
<tr>
<th>Fields Considered</th>
<th>$l^2$ Error</th>
<th>$l^\infty$ Error</th>
<th>$N_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact Field and DCC</td>
<td>$1.22 \times 10^{-2}$</td>
<td>$5.41 \times 10^{-1}$</td>
<td>49</td>
</tr>
<tr>
<td>Exact Field and DCC with filtering</td>
<td>$4.20 \times 10^{-3}$</td>
<td>$9.49 \times 10^{-2}$</td>
<td>45</td>
</tr>
</tbody>
</table>

The table shows the $l^2$ and $l^\infty$ norms of the error between the fields displayed in Figures 5 e) and f). The number of vectors used in each case is also provided. For the DCC algorithm to be effective, a reasonable number of particles per window needs to be used and thus less vectors are produced when compared to the OT-PIV method.

4.2. Applications to Realistic Data. We now present a strategy to apply our technique to more realistic PIV data and display the results of numerical experiments. First, we smooth the PIV grayscale images to reduce the noise coming from out-of-plane loss of particles in the images. This will also transform the particles distributions into Gaussian-like functions. We employ a cutoff in frequency space as a low-pass filter. More specifically, we cut the higher frequencies of both images by computing the Discrete Fourier Transform (DFT) of the images using the Fast Fourier Transform (FFT) algorithm. We then truncate the data in frequency space by keeping only the frequencies located inside a square centered at the origin with sides of length twice a certain proportion $\alpha_t$ of the Nyquist frequency ($\alpha_t = 0.25$ for the case considered). Note that this smoothing step is not always necessary (depending on the amount of out-of-plane loss of particles). Then, we directly take the resulting images as discretizations of the initial and final densities.

After applying the OT algorithm to the two smoothed images, we obtain a vector field where every vector corresponds to a grid point on the numerical grid. However, many of these grid points do not correspond to a particle’s location and for general fluid flows, the vectors obtained using OT would not necessarily be a good approximation of the velocity field at that location. Moreover, according to the results of Section 3, the vectors obtained at particle locations should be faithful mostly for the brighter particles. Let us therefore introduce the post-processing parameter $\beta \in [0, 1]$ which we take as a brightness threshold used to identify the brighter particles on the initial image. We then filter the OT vector field to keep only the vectors associated to a point on the image with brightness level higher than $\beta$, because these points will correspond to brighter particles on the image. The value of $\beta$ will have to be different depending on the strength of the smoothing employed and depending on the accuracy desired for the vectors obtained in the approximate field. We will also apply the No Split and the Particle End-Position Correction filters to the approximate fields. In addition, the resulting vector field will be nonuniformly sampled because the brighter particles are not necessarily uniformly distributed. In order to compare it with the fields on uniform grids obtained with other methods such as cross-correlation, we also interpolate our vector field on a uniform grid. To do this, we use MATLAB’s TriScatteredInterp function which builds the Delaunay triangulation associated with the data points. This triangulation creates a surface which we use to approximate
Fig. 6. Test Case 1. Figure c) displays the result of the multi-pass DCC algorithm applied to a) and b) (the window size is 32 pixels and the step size is 16 pixels). A Gaussian 2X3 points subpixel approximation was also employed.

Fig. 7. Test Case 1. Results of the OT algorithm applied to the images with $\alpha_t = 0.25$ and different values of $\beta$.

Let us now present the results of a series of numerical experiments on two test cases. For the first case, we selected images from the 2001 PIV challenge case B [13, 19]. These images were created synthetically to simulate PIV-like images generated by particles evolving in a strong vortex, with typical out-of-plane loss of particles and varying particle sizes. Several sets of images were created with different seeding densities and particle sizes to provide tests for PIV algorithms. We focus on the case B3 which corresponds to a medium seeding density and to small particles. Then, in test case 2, we use our method on a set of real PIV images showing a slightly turbulent air flow with small water droplets [25].

**Test case 1**

First, we consider a vortex created to simulate real PIV-like images which contain different brightness intensity levels and out-of-plane loss of particles (2001 PIV
Challenge B3 [13, 19]). Figure 6 presents the two 512 × 512 pixels test images representing the initial and final distributions of particles in the synthetic fluid as well as the resulting field obtained using the DCC Cross-Correlation algorithm of the PIVlab toolbox in MATLAB [20, 21]. In Figure 7, we present the results of the OT algorithm for this case, for different values of $\beta$. When a higher brightness cutoff $\beta = 0.7$ is selected, fewer vectors remain in the final field, but these vectors more accurately follow the underlying target vector field. As this threshold is lowered to $\beta = 0.5$, more vectors are selected at the price of a loss in accuracy for some of them. Therefore, as anticipated, vectors corresponding to brighter particles are more accurate.

Let us now look at the effect of the No Split filter in Figures 8 a), b) and c). We see that higher values of the filter’s strength $k_{ns}$ result in several spurious vectors effectively removed. However, increasing the strength of the filter (decreasing $k_{ns}$) removes most of the vectors close to the center of the vortex, which makes sense
since we are getting near the stagnation point where particles are moving much faster and mismatches by the OT algorithm are thus more likely to occur. Notice that the cross-correlation method also experiences difficulties close to the vortex’s center.

We also present the result obtained by the particle end-position correction post-processing method in Figures 8 (d), (e), (f) and (g). As we have seen previously, this filter improves the overall quality of the approximation, but introduces significant errors, especially for vectors on the outer edge of the vortex for particles whose displacement is on the order of one pixel. Indeed, as we associate the tracers’ end positions with brightest neighboring pixels, we loose the subpixel precision given by the transport algorithm. This could potentially be fixed by a subpixel approximation similar to the ones employed in traditional cross-correlation techniques. This will be the subject of future research. Figures 8 (h) and (i) then shows the velocity fields obtained by interpolating linearly the OT field obtained with a cutoff in frequency on the same uniform grid as the one used in the cross-correlation field for the sake of visual comparison. It is not possible for us to perform a quantitative analysis of the effect of interpolation for different fields obtained with different $\beta$, but this will also be the subject of future research (more details about this will be given in the conclusions’ section). Finally, note that when applied to the other cases of the 2001 PIV challenge B, the OT-PIV procedure performed even better for larger particles and for lower seeding densities relative to the performance of OT-PIV presented here.

**Test case 2**

We now select data from a real experiment, as opposed to the different types of vortices presented before. The images we used are the first two in the sequence given in [25], which consists of PIV images showing a slightly turbulent air flow seeded with small water droplets. We observe in Figure 9 that the results are better for brighter particles, as expected. The interpolated field is not too far from the field obtained using the cross-correlation algorithm, although larger errors are present close to the boundary due to the periodic boundary conditions employed in the resolution of the transport algorithm. This boundary issue can be dealt with by embedding the physical domain containing particles in a larger numerical domain, thus padding the boundaries with a region containing no particles, but the algorithm required to solve the related Monge-Ampère equation needs to handle vanishing densities in order to do this. For both OT and cross-correlation fields, there are parts of the flow for which the approximation is closer to reality than the other method, but the cross-correlation field is overall closer to the solution for this particular example.

**5. Comparison with Cross-Correlation for Low Seeding Density.** Cross-correlation algorithms are not always accurate for a region at low seeding density, or equivalently for high-resolution fields. The OT procedure appears to perform better in those cases. In order to demonstrate this, we have generated 3 images with randomly placed tracers of equal weights, and we let them evolve in a preselected field (a vortex similar to the one used in Section 4.1.4). We used the DCC algorithm of MATLAB’s PIVlab toolbox as a cross-correlation algorithm to compare with [20, 21]. The results are presented in Figure 10. We see that in every case, the OT procedure produced less spurious vectors than the cross-correlation method employed. These errors are a direct result of the use of interrogation windows: particles moving from one window to another are more likely to cause a mismatch at low seeding density, even when multiple passes to move the location of the interrogation windows are employed. Due to the low seeding density, these errors remained present even when the size of the interrogation window was varied for the current experiment. When the seeding density
is increased, we see that less spurious vectors are produced by the cross-correlation method because the effect of one particle crossing windows is diminished by the other particles which remain in the same window (possibly only after the window has been moved). Note that to ensure the fairness of this qualitative comparison, no post-processing was applied to any of the fields presented in Figure 10 (for both OT and DCC).

6. Concluding remarks. Based on our previous work [17, 18], we presented a new PIV method based on the theory of optimal mass transportation (OT-PIV) to retrieve the flow velocity of a fluid. While traditional PIV algorithms are typically based on cross-correlations between the successive images recorded, our results indicate that when augmented with proper post-processing filters, the OT-PIV method yields equally valid results in most cases, and even better in some. By analyzing both theory and experiments, we showed that, in the case of uneven particle brightness, the OT map performs better on the brighter particles than it does on the faded ones. Using only the locations of particles whose brightness in the initial image exceeds a certain threshold to construct the flow velocity drastically improves the accuracy of the later. We also found that, because the light distribution around a particle is closer to a Gaussian distribution with a non-zero standard deviation than it is to a Dirac-delta function, the mass associated with a single particle can be split or spread around two or more particle locations. We thus introduced a no split filter which consists in discarding particles whose mass is spread by the transport map over a region.

Fig. 9. Test Case 2. Results of the OT algorithm applied to the first two images of package P3 of [25] ($\alpha_t = 0.35$).
exceeding a certain limit. This improves the flow in regions of very high densities or rapid rotations prone to mismatching between particles by the OT map. Also, as already documented in [18], the OT map does not send the initial particle position to its corresponding final position, but rather move it to a nearby location. Thus by remapping each initial particle position to the closest particle position in the final distribution within a certain radius, we recover the true final position of that particle, unless there is a mismatch.

The results obtained in this paper also provide insight on the sensitivity of the post-processing filters to the associated parameters $\beta$, $k_{ns}$ and $c$. The brightness cutoff parameter $\beta$ depends on the seeding density and on the relative brightness of particles with respect to their neighbors. Indeed, for images at lower seeding densities, a lower $\beta$ can be safely employed. This is the case when the particles mostly remain within the tracking range, which we can validate by verifying if the average displacement is smaller than the average distance between particles. Theorem 3.1 provides a theoretical justification for this observation. It could also be possible to employ a nonuniform $\beta$ which could be taken as a threshold on the relative brightness of a particle with respect to its immediate neighbors. However, even when particles are quite isolated from each other, if the noise level in the pictures is high when compared to the brightness of a particle, the parameter $\beta$ cannot be taken too low as there might be significant mass exchanges between noise and particles ($\beta$ should be at least greater than a multiple of $1 - r$). The parameter $\beta$ essentially provides a trade-off between accuracy and quantity of information. An interpolation process may be used to obtain a uniform field, but it will potentially introduce more errors in regions where less vectors are selected. Lowering the value of $\beta$ would result in more vectors being selected and thus the interpolation errors would decrease, but these vectors will
have a greater error due to the method. The extension of the OT field obtained from
the bright particle grid to the whole domain will be the subject of future research.

Concerning the no split filter parameter $k_{ns}$, its value also depends on the seeding
density. For lower densities within the tracking range, a lower value of $k_{ns}$ (close to
1.5 for example) can be selected as we do not expect particles to be influencing each
other too much, unless there is a clear mismatch. For higher densities, we saw from
numerical experiments that taking $k_{ns}$ too low removes too many vectors in regions
of greater turbulence, and thus a value closer to 3 or 4 is more desirable. Finally, the
parameter $c$ depends on the resolution of the images at hand (the number of pixels
per particle), and it should be selected so that the maximum correction corresponds
to the radius of a typical particle, in pixels.

One apparent shortcoming of the OT-PIV method with post-processing filters
presented above lies in the fact that two out of three of the proposed filters sim-
ply discard particles that are problematic. This may have the undesired feature of
limiting the flow scales recovered in this fashion; hence, some of its fine-small scale
structure will be lost. To extend the OT-PIV method to highly turbulent flows where
high seeding densities are necessary, multi-pass algorithms similar to the ones em-
ployed in traditional cross-correlation methods could be employed. Moreover, given
the continuum nature of the transport flow, one is tempted to extend this approxi-
mation outside the particle locations, but as already pointed out in [18], the flow field
associated with the OT problem solves pressureless and inviscid transport equations
which are different from the Navier-Stokes equations. High seeding densities are thus
required in order to capture any fine features the fluid flow may possess.

It is worthwhile giving a few remarks on the computational cost of the OT method
compared with the cross-correlation algorithm. Our OT method is based on the nu-
merical solution of the Monge-Ampère equation, and as shown in [17] this method has
a computational complexity of $O(N \log N)$, which is similar to the cross-correlation
algorithm since it is also based on the Fast Fourier Transform. For the actual com-
puting times, the previous MATLAB-based version of the OT PIV method developed
in [18] required several hours. However, our new FORTRAN code requires only a
few minutes of CPU time for a typical resolution of $512 \times 512$ grid points. This new
computation time is comparable to the computation time of the cross-correlation
algorithm [20, 21]. For example, for the images of Figure 8, the DCC algorithm
took only 6 seconds while the OT algorithm required 130 seconds on an Intel(R)
Core(TM) i7-3537U CPU 2.00GHz. The OT method is still slower than the DCC
cross-correlation algorithm, but at least now the times are more realistic for day-to-
day usage. We must stress that our FORTRAN code is not professionally optimized,
unlike the cross-correlation package [20, 21]. This FORTRAN code is available from
the authors upon request.

We finally point out that due to the nature of the problem, any direct method
will be limited to small displacements; this is not an issue of OT alone. Too large
displacements will unavoidably lead to particle mismatch and thus to incorrect velocity
fields. As previously mentioned, future research on Optimal Transport for Particle
Image Velocimetry will involve an extension of the current procedure to a convection
method based on finding the Navier-Stokes solution that is closest to the inferred
flow field. The approximate field will therefore be more physically realistic outside
particle locations, solving the Navier-Stokes equations as opposed to the pressureless
Euler equations for the current method. Some work has already shown possible ways
to perform such extension (e.g. [9]), but it will need to be adapted to the PIV problem
under study here.

Acknowledgement. This work is part of L.P. Saumier’s PhD dissertation. He is supported by fellowships from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Victoria. The research programs of B. Khouider and M. Agueh are supported in part by NSERC.

REFERENCES