

Figure S1.

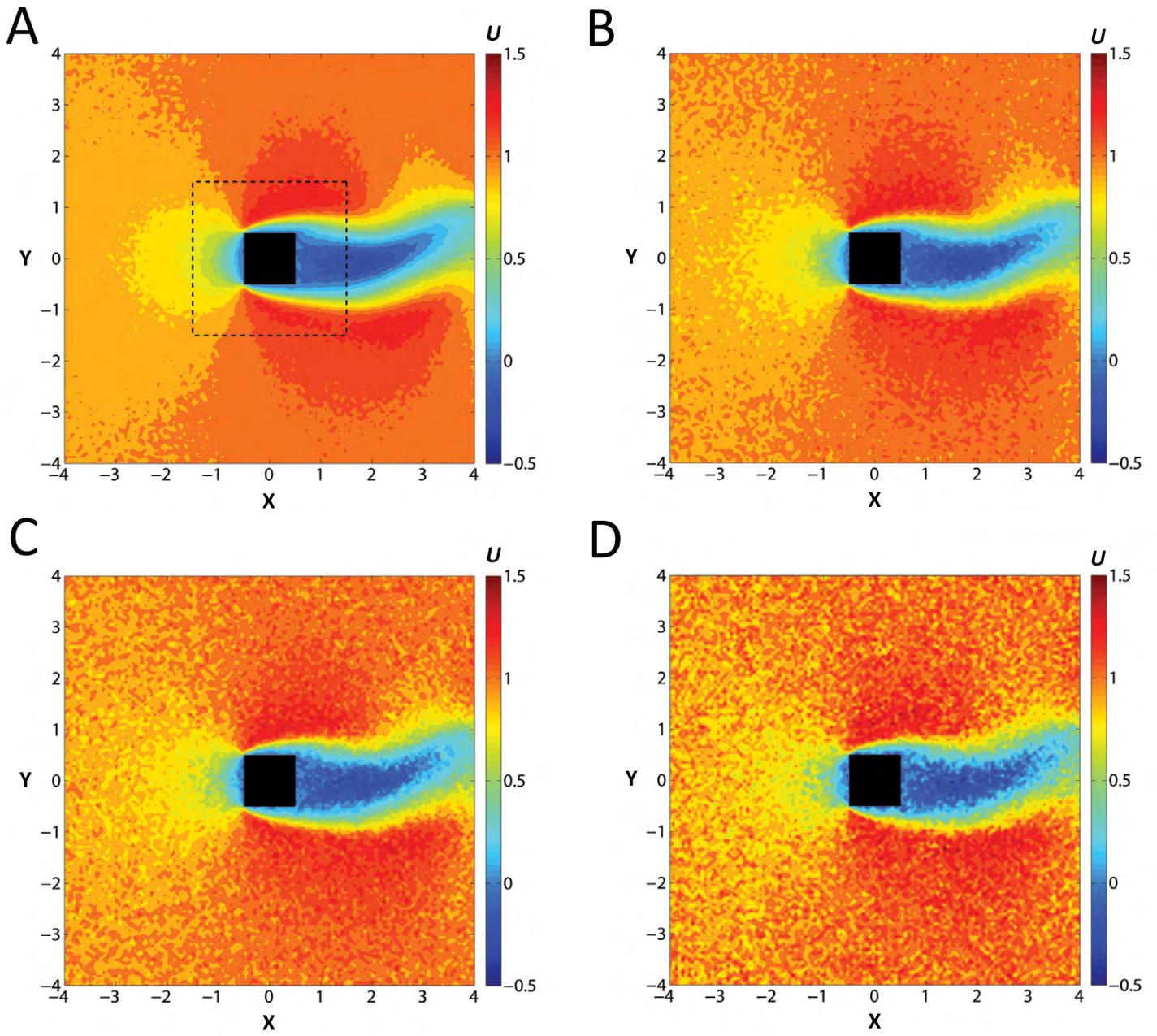


Figure S2.

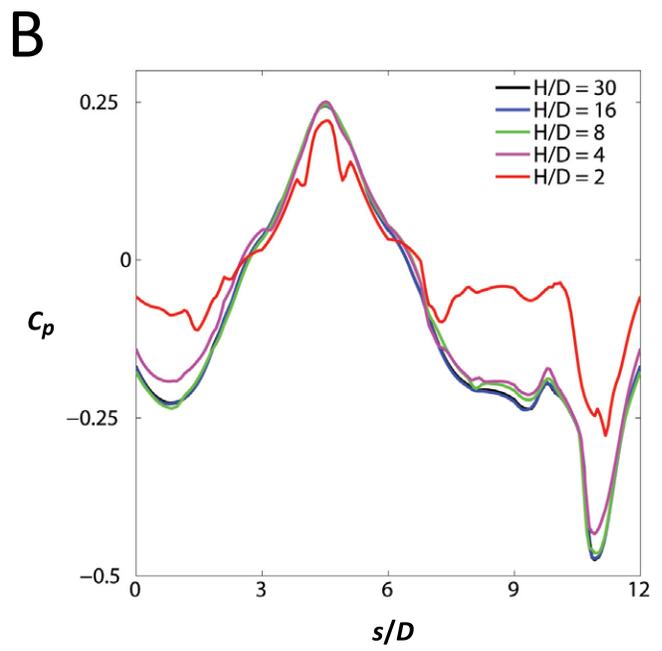
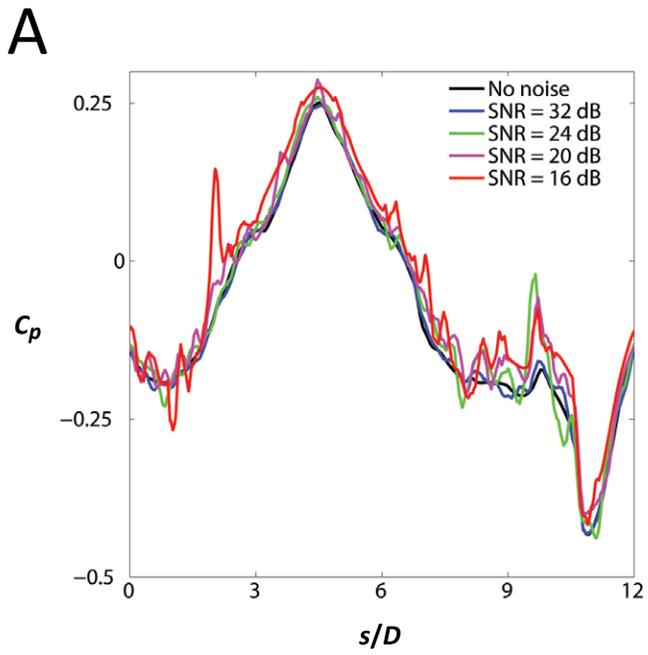


Figure S3.

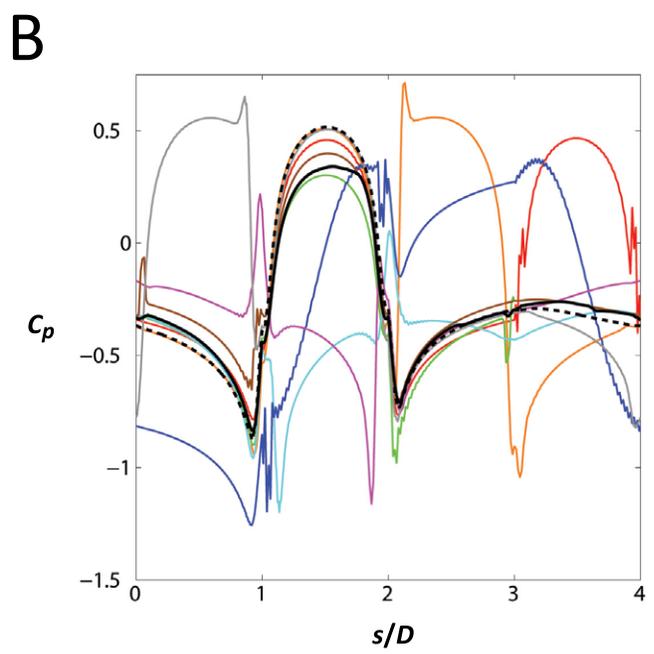
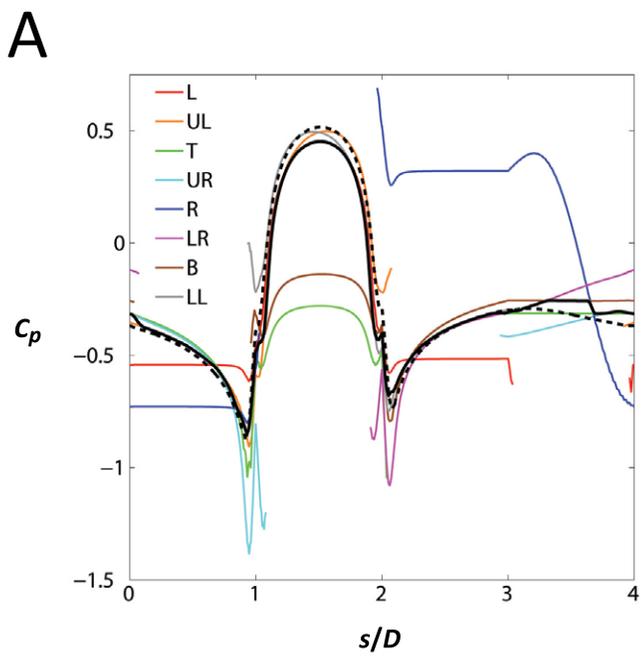


Figure S4.

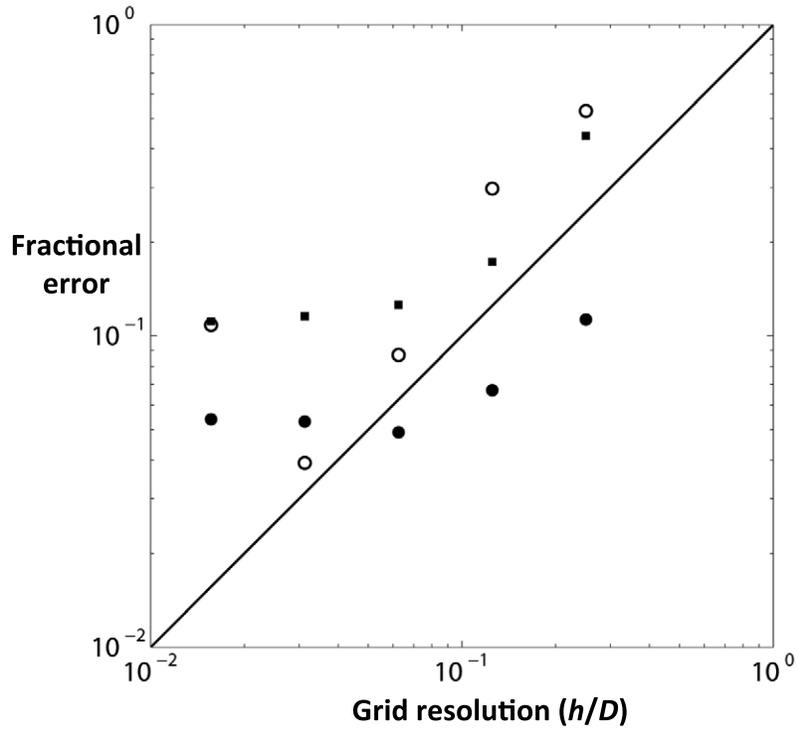


Figure S5.

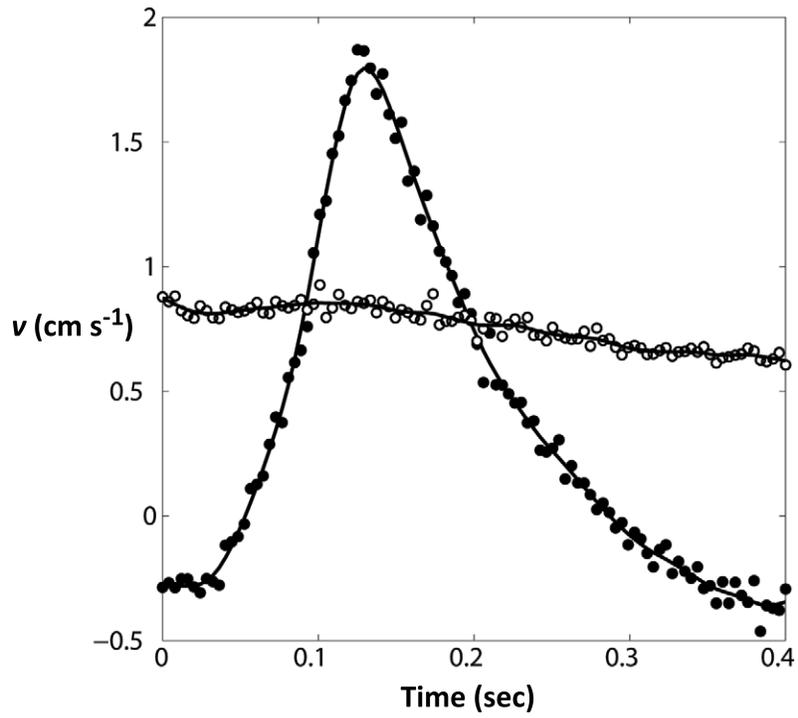


Figure S6.

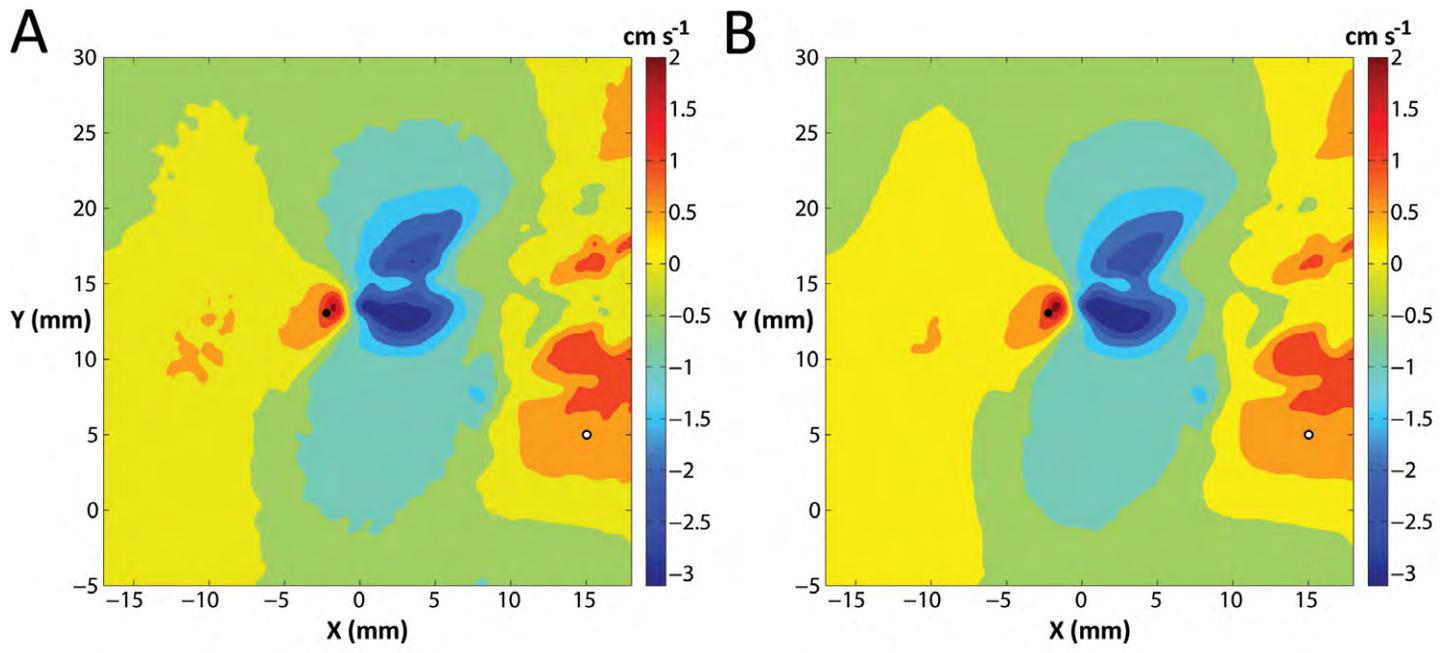


Figure S7.

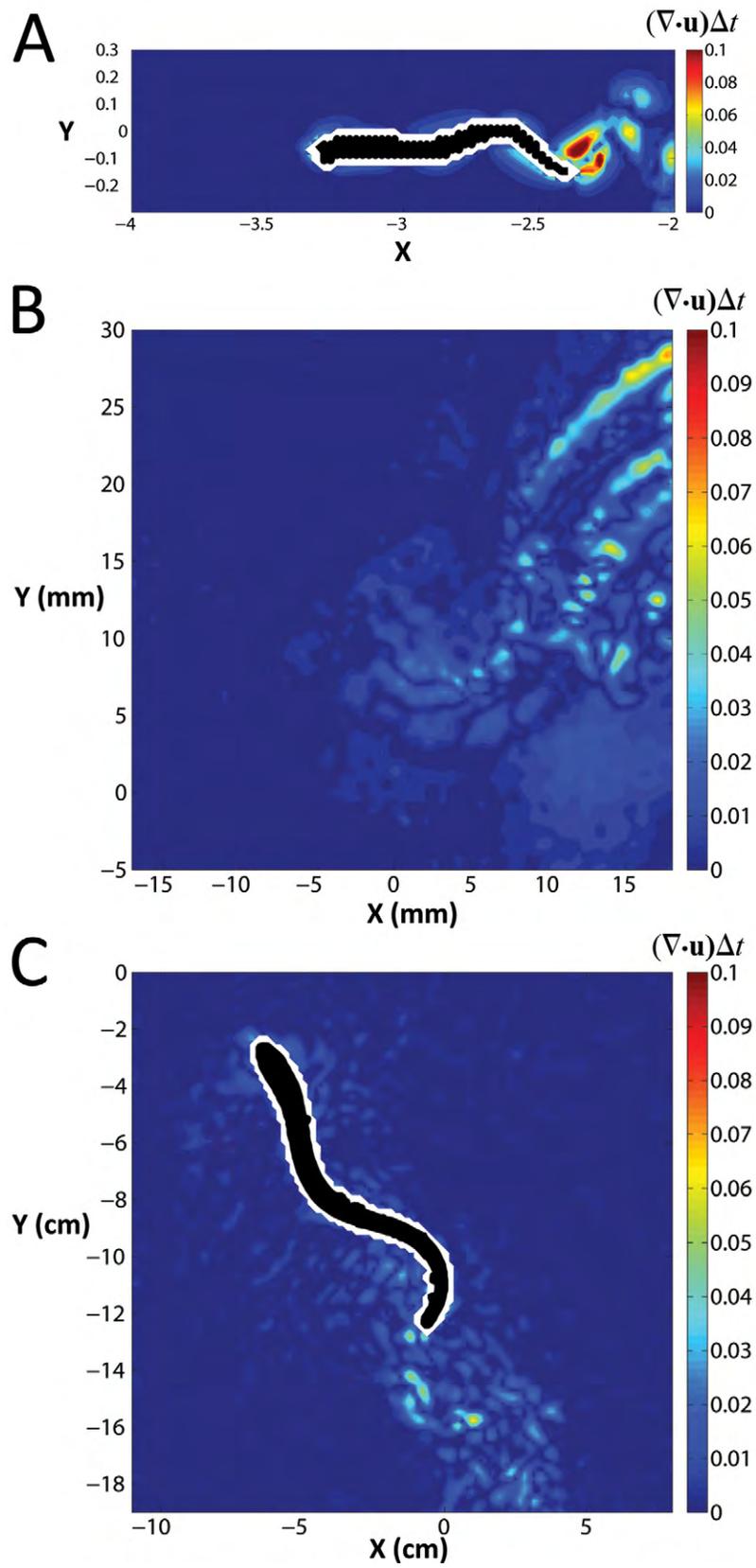


Figure S8.

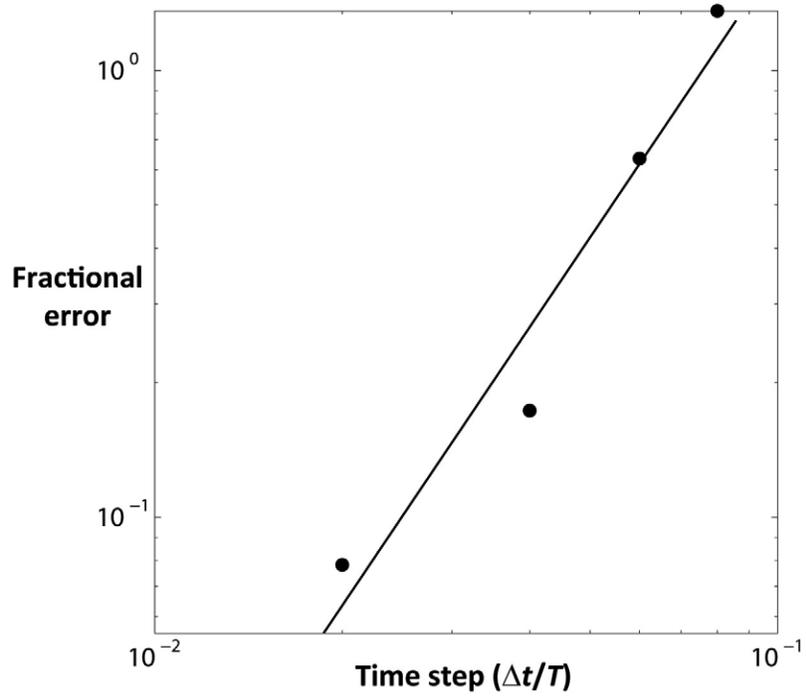


Figure S9.

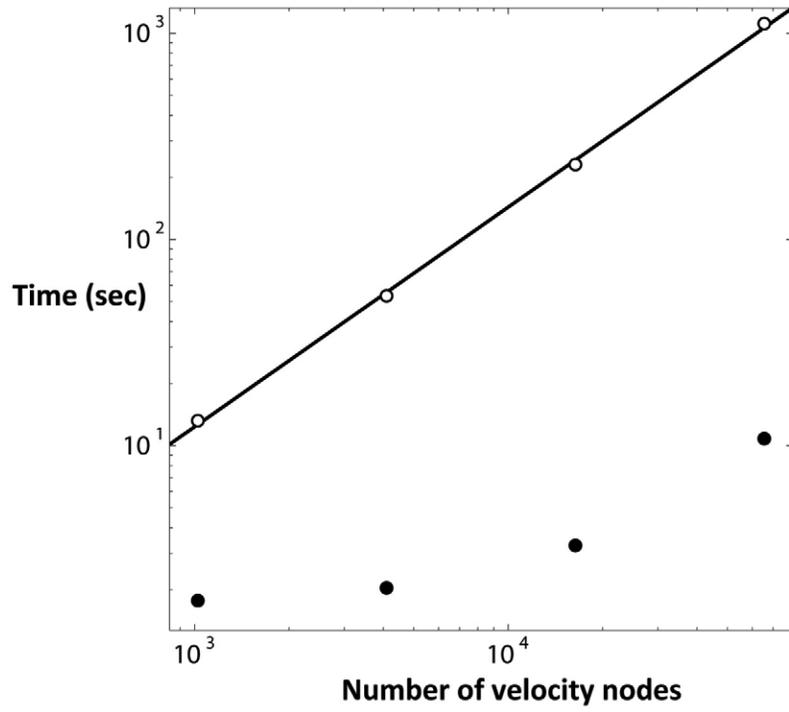


Figure S10.

APPENDIX 1: ADDITIONAL CHARACTERIZATION OF QUASI-STEADY ALGORITHM

Effect of median polling

To illustrate the contribution of each integration path to the final pressure estimate, Fig. S1 plots the pressure on the body surface (at $0.1D$ away from the fluid–solid interface, to avoid the spatial filter artifact) and on two additional concentric square contours in the domain (e.g. the dashed contour in Fig. S2A), as computed using each of the eight families of integration paths. The results illustrate the benefit of median polling versus an average of the integration path results. For example, only five of the integration path families intersect the upstream face of the bluff body without passing through the body itself. The median of these curves is in good quantitative agreement with the correct surface pressure (Fig. S1A). The pressure profiles for the two concentric square contours in the domain (i.e. Fig. S1B, Fig. S1C) indicate that the contribution of each family to the final pressure estimate is spatially non-uniform. This is illustrated qualitatively in Fig. S1D, which is a contour plot that colors each point in the domain according to the path family that corresponds to the median pressure at that point. Because there are eight path families, the median is always the average of the two intermediate values (where none of the paths is undefined due to intersection with the solid body). To reveal the individual integration path family contributions, a ninth pressure value equal to the mean of the eight path families is included in Fig. S1D, so that the median pressure is from either a single integration path family or the mean. The contour plot indicates that each integration path family contributes to the final pressure field estimate, but the contributions are often spatially localized. The pressure estimates for the R family of integration paths are noticeably less accurate than the other families (e.g. Fig. S1A) and yet, as illustrated in Fig. S1D, these paths determine the pressure estimate in the far wake. This leads to the observed poorer pressure estimate in that region of the flow (e.g. Fig. 1B). The underlying source of this effect is discussed below in the section examining the effect of boundary conditions.

Effect of global measurement error

Perhaps the most important test of the algorithm is its robustness to global measurement errors, such as those associated with empirical measurements. Fig. S2 illustrates the streamwise velocity contours for data sets with increasing levels of Gaussian white noise superimposed on the u and v velocity components. The highest levels of noise, corresponding to the lowest signal-to-noise ratios, are higher than typical PIV data but possibly representative of instantaneous two-dimensional data collected in a highly turbulent flow field, where out-of-plane motion can reduce data quality. Comparison of the pressure profiles on a square contour centered on the bluff body and with side length $3D$ so that it passes through the salient flow features (i.e. Fig. S2A) indicates that, with the exception of the highest noise level tested, the quantitative pressure estimates remain consistent with the noise-free result despite relatively high noise (Fig. S3A). Error in the pressure estimate is not additive with the increasing noise level because errors do not accumulate uniformly on the eight paths that arrive at each point in the domain. Hence, median polling remains an effective filter irrespective of the noise magnitude, up to the second-highest noise level tested. At higher noise levels, contour plots of the pressure estimate begin to exhibit spatial discontinuities reminiscent of the median contributions in Fig. S1D. Because the pressure estimates from each integration path family begin to diverge in the presence of high noise levels, median polling in this case leads to spatially discontinuous changes in pressure. Results of this sort are an indication that measurement noise in the input velocity data has become unacceptably large.

Effect of boundary conditions

As mentioned previously, a major assumption implicit in the present algorithm is that the pressure on each integration path is zero at its originating point on the boundary, to avoid the need for a computationally expensive iteration scheme to solve for the boundary pressure as part of the field solution (Liu and Katz, 2006). Although this assumption can be reasonable for large domains, it is prudent to investigate the dependence of the pressure estimate on the domain size. Fig. S3B plots the pressure on a square contour centered on the bluff body (see Fig. S2A) for domains ranging in size from $H/D=2$ to 30, where H is the half-width of the domain. The results indicate that the accuracy of the algorithm (and, hence, the assumption regarding the boundary pressure) is not significantly compromised until the domain shrinks to $H/D=2$. This limitation is important to keep in mind when designing experiments that will make use of the present algorithm.

Notwithstanding the demonstrated efficacy of the aforementioned assumption regarding the boundary pressure, examination of the individual pressure estimates on each family of integration paths reveals that some individual estimates are severely compromised by this assumption. Most notably, the R family of integration paths originate at the downstream boundary of the domain, where vortices shed by the bluff body exit the measurement window and create a non-zero pressure on that boundary. Hence, this family of pressure estimates is significantly less accurate than the others, as seen in Fig. S1A, for example. The benefit of the median polling approach is that this estimate is usually discarded in determining the final pressure estimate. In contrast, previous methods would include pressure estimates affected by

the downstream boundary in the final averaged pressure estimate, and therefore require additional computational effort to resolve the correct pressure on that boundary via iterative processes. However, the present method does suffer in that the pressure in regions close to the downstream boundary is based either on integration paths that originate at the downstream boundary where the pressure is non-zero (i.e. R, UR and LR families) or on long integration paths from the other boundaries. The relatively large error accumulated on the long integration paths can make them an even poorer estimate of the local pressure near the R boundary (cf. Fig. S1D); hence, the median pressure in this region is less accurate than in the rest of the domain. This limitation is inherent in the present method and should be kept in mind when using the technique for flows with large velocity gradients at any of the boundaries.

Effect of fluid viscosity

It is useful to examine the role of the viscous term in Eqn 2, as many previous pressure estimation methods neglect this term. Fig. S4A plots the pressure estimates on the body surface for each integration path family as in Fig. S1A, but for a pressure estimate that neglects the viscous term in the Navier–Stokes equation. The effect is most noticeable in integration paths orthogonal to the mean flow (i.e. T and B), especially near the upstream face of the bluff body. This result can be understood by considering the contributions to the pressure gradient from the streamwise and transverse material acceleration components, Du/Dt and Dv/Dt , relative to the contributions from the Laplacian of the streamwise and transverse velocity components in the viscous term. As the flow approaches the upstream face of the bluff body, the material acceleration is dominated by streamwise fluid particle deceleration Du/Dt . However, the pressure computed on integration paths that are orthogonal to the mean flow (i.e. T and B) is independent of this term. Instead, on these paths the pressure depends to a significant degree on the local velocity curvature (i.e. second spatial derivative) as the flow is turned around the bluff body. This effect is captured in part by the Laplacian of the transverse velocity. Hence, its neglect leads to an underestimate of the pressure on those integration paths. The net effect of the neglected viscous term is minimal due to the median polling approach implemented presently, i.e. the T and B paths do not represent the median pressure estimate on the upstream face of the bluff body and are therefore not a factor in the final pressure estimate in that region of the flow.

Effect of fluid–solid interfaces

An aspect of the present algorithm that can be cumbersome is the treatment of the fluid–solid interface to eliminate the effect of integration paths that pass through solid objects in the flow. For example, for moving objects, this approach requires the identification of the fluid–solid interface in each data frame. To illustrate the effect of the interface treatment in the algorithm, Fig. S4B plots pressure estimates on the body surface, where the algorithm has been implemented without treating the fluid–solid interface. The accuracy of the algorithm is noticeably affected due to additional spurious pressure estimates on paths that cross the body. However, it is noteworthy that the final pressure estimate is still qualitatively consistent with the correct pressure field. It may therefore be acceptable to bypass the fluid–solid interface treatment where only a quantitative approximation of the pressure field is sought. The results of the analysis in Fig. 4B, which did not identify the boundary of the medusa as was done for the numerical data, suggest that suitable pressure estimates can still be achieved where treatment of the fluid–solid interfaces is not practical.

Spatial convergence

The spatial convergence of the quasi-steady algorithm was evaluated by computing the pressure on a square contour immediately adjacent to the region of undefined pressure on the bluff body, and by integrating the pressure to compute the net force in the streamwise and lateral directions. Fig. S5 plots the fractional error in these calculations [using the pressure from the numerical simulation (CFD) as the true value, i.e. $|F_{\text{CFD}} - F_{\text{estimate}}|/F_{\text{CFD}}$] versus the grid resolution. At relatively large grid spacing, the log–log curve has a slope of 1, indicating the expected first-order spatial convergence of the method. As the grid spacing is further reduced, the error decreases more slowly. This effect is due to a combination of inherent model error and numerical round-off error. A convergence plot for calculations of the time-averaged streamwise force is included. Its departure from first order convergence at small grid spacing confirms that the quasi-steady approximation is not solely responsible for errors at small grid spacing. For grid spacing less than $D/16$, the error falls below 5% for the instantaneous pressure and approaches 10% for the time-averaged pressure.

APPENDIX 2: ADDITIONAL CHARACTERIZATION OF UNSTEADY ALGORITHM

Effect of temporal filter

Fig. S6 plots the time series of v component data at two selected points in the jellyfish PIV data set. Despite the relatively smooth spatial distribution of velocity as illustrated in Fig. 4A and Fig. S7A, there is non-trivial scatter in the temporal data at both spatial locations. Flow accelerations computed by using finite differences of adjacent velocity fields would be subject to large errors because the local slope varies

considerably between adjacent pairs of velocity fields. A temporal filter of the data is therefore essential in this case. Fig. S6 indicates the corresponding smoothing splines that were fitted to the data and subsequently used to compute the material acceleration. The splines capture the true transient behavior of the flow while eliminating the high-frequency noise. Comparison of Fig. S7A and Fig. S7B illustrates that the spatial distribution of velocity is relatively unaffected by the temporal filter. It is prudent to note that if a flow exhibits real, high-frequency oscillations in the velocity, e.g. in turbulence, it will be essential that the PIV measurements are of sufficiently high temporal resolution such that the smoothing spline does not discard those temporal trends. In those cases, it is important that the frequency of PIV measurements satisfies the Nyquist sampling criterion with respect to the time scale of turbulence fluctuations, while concurrently avoiding sampling at frequencies high enough to incur numerical round-off errors in the calculation (Beckwith et al., 2007).

Effect of out-of-plane flow

Given that two-dimensional PIV data represent a projection of three-dimensional flow, it is useful to characterize the impact of that limitation on the accuracy of the present methods. As in prior work (Stamhuis and Videler, 1995), Fig. S8 characterizes the out-of-plane motion by computing the divergence of a two-dimensional velocity field extracted from the three-dimensional numerical simulation of the self-propelled swimmer and of the PIV data sets examined in Fig. 4. The divergence is made dimensionless by multiplying it by the time step between adjacent velocity fields, as this time scale is most relevant for calculation of the material acceleration. The plots effectively quantify the fractional change in the volume of an idealized infinitesimal fluid particle between adjacent velocity fields. Because the flows are incompressible, the fractional volume change would be identically zero if the flows were two-dimensional. Deviation from zero values can therefore be attributed to velocity gradients perpendicular to the plane of the velocity field, i.e. out-of-plane flow.

The results in Fig. S8 indicate that the three-dimensional numerical simulation exhibits greater out-of-plane flow than the PIV measurements. Given the demonstrated accuracy of the algorithm in the case of the three-dimensional numerical data, we can conclude that the algorithm is robust to out-of-plane effects at the magnitudes found in typical PIV measurements. To be sure, the divergence metric does not capture out-of-plane flow where there is no flow gradient in the perpendicular direction. However, in such cases, the PIV would itself be difficult to acquire, as the seed particles would not remain in the plane of the laser sheet sufficiently long to enable temporal cross-correlation of their positions.

Temporal convergence

The temporal convergence of the unsteady algorithm was evaluated by plotting the fractional error in the pressure at the head of the self-propelled swimmer at an instant of high acceleration [using the pressure from the numerical simulation (CFD) as the true value, i.e. $|p_{\text{CFD}} - p_{\text{estimate}}|/p_{\text{CFD}}$] versus the time step between velocity fields (Fig. S9). Although the available data were limited to time steps from $0.02T$ to $0.08T$, the results are consistent with temporal convergence that is higher than second order, except as the smallest step size is approached, where further reduction in error is limited by inherent model error and numerical round-off error. At a temporal spacing of $0.02T$, the error in the pressure at the head is $\sim 8\%$.

When the unsteady algorithm is applied to a sequence of velocity fields that are spaced too closely in time, leading to increased numerical error, the results appear similar to those described in the context of global measurement error (Appendix S1) in which the pressure contours exhibit spatial discontinuity reminiscent of Fig. S1D.

Computational cost

Fig. S10 plots the time required for a single 3 GHz processor to apply the temporal filter and to compute the pressure field for velocity fields from 32×32 to 256×256 nodes, which is a practical upper limit for typical PIV measurements due to camera pixel resolution. The time required for the temporal filter scales linearly with the number of nodes in the velocity field. The cost is independent of the number of velocity fields in the sequence of up to the tested several hundred frames of data. The computational cost of the subsequent pressure calculation scales sublinearly in the range tested, and it is significantly lower than the cost of the temporal filter in absolute terms. For example, for a 128×128 velocity field, each pressure field is computed in ~ 3 s, as compared with 46 s using an existing iterative algorithm (Liu and Katz, 2006).

Fig. S1. (A) Pressure on surface of bluff body estimated using quasi-steady algorithm. Measurement contour is offset by $0.1D$ from the fluid–solid interface. The pressure coefficient $c_p = p/(\rho U^2)$. s is the local surface coordinate and increases in the counter-clockwise direction from the upper right corner of the bluff body. Dashed black line, pressure from numerical simulation; solid black line, pressure estimated using quasi-steady algorithm; solid colored lines, pressure estimates based on each family of integration paths.

Colors correspond to paths in Fig. 5 and to the legend in D. (B,C) Pressure on square contours centered on the bluff body and with side length $2D$ and $3D$, respectively (e.g. Fig. S2A). s is the local surface coordinate and increases in the counter-clockwise direction from the upper right corner of each square contour. The difference in abscissa in panels A–C reflects the different contour perimeters. (D) Contour plot that colors each point in the domain according to the path family that corresponds to the median pressure at that point. To reveal the individual integration path family contributions, a ninth pressure value equal to the mean of the eight path families is included, so that the median pressure is from either a single integration path family or from the mean.

Fig. S2. Streamwise velocity contours for flow field with Gaussian white noise added to reduce the signal-to-noise ratio (SNR). (A) SNR=32 dB. (B) SNR=24 dB. (C) SNR=20 dB. (D) SNR=16 dB. Dashed square in A indicates contour on which quasi-steady pressure estimates are compared in Fig. S3A.

Fig. S3. (A) Quasi-steady pressure estimate on contour shown in Fig. S2A for varying signal-to-noise ratio. s is the local surface coordinate and increases in the counter-clockwise direction from the upper right corner of the square contour. (B) Quasi-steady pressure estimate on contour shown in Fig. S2A for varying measurement domain size. H is the half-width of the measurement domain.

Fig. S4. (A) Pressure on surface of bluff body estimated using quasi-steady algorithm without viscous term. Measurement contour is offset by $0.1D$ from the fluid–solid interface. s is the local surface coordinate and increases in the counter-clockwise direction from the upper right corner of the bluff body. Dashed black line, pressure from numerical simulation; solid black line, pressure estimated using quasi-steady algorithm; solid colored lines, pressure estimates based on each family of integration paths. (B) Pressure on surface of bluff body estimated using quasi-steady algorithm without treatment of fluid–solid interfaces to remove integration paths that pass through the solid body.

Fig. S5. Spatial convergence of the algorithm. Log–log plot of the fractional error in instantaneous streamwise (closed circles), instantaneous lateral (open circles), and time-averaged streamwise (closed squares) force coefficients versus grid resolution for numerical simulation of two-dimensional flow past the bluff body. Solid line indicates a slope of 1 corresponding to first-order convergence. Deviation from first-order convergence at small grid resolution is due to a combination of model error and numerical round-off error.

Fig. S6. Time series of v component data at two selected points in the jellyfish PIV data set. Symbols indicate original PIV data at corresponding locations identified in Fig. S7. Solid curves indicate respective smoothing splines.

Fig. S7. (A) Contour plot of v component of original velocity measurement during middle of jellyfish bell contraction. (B) Contour plot of v component temporal spline-filtered velocity measurement during middle of jellyfish bell contraction. Location of animal is similar to that indicated in Fig. 4A, although earlier in the bell contraction phase. Closed circle near bell margin and open circle in wake indicate locations of temporal profiles in Fig. S6.

Fig. S8. Contour plots of normalized two-dimensional divergence for (A) three-dimensional numerical simulation of self-propelled swimmer, (B) PIV measurement of freely swimming jellyfish (cf. Fig. 4A), (C) PIV measurement of freely-swimming lamprey (cf. Fig. 4C). Dimensional divergence is normalized by multiplying by the time step between sequential velocity fields in each case.

Fig. S9. Temporal convergence of the algorithm. Log–log plot of the fractional error in pressure at the head of the simulated three-dimensional self-propelled swimmer versus time step between velocity fields (closed circles). Solid line indicates a slope of two corresponding to second-order convergence.

Fig. S10. Computational cost of the algorithm, as quantified by the time required for a single 3 GHz processor to apply the temporal filter (open circles) and to compute the pressure field (closed circles) for velocity fields from 32×32 to 256×256 nodes. Solid line indicates slope of 1.