



HAL
open science

TracTrac: A fast multi-object tracking algorithm for motion estimation

Joris Heyman

► **To cite this version:**

Joris Heyman. TracTrac: A fast multi-object tracking algorithm for motion estimation. *Computers & Geosciences*, 2019, 128, pp.11-18. 10.1016/j.cageo.2019.03.007 . insu-02088564

HAL Id: insu-02088564

<https://insu.hal.science/insu-02088564>

Submitted on 3 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Accepted Manuscript

TracTrac: A fast multi-object tracking algorithm for motion estimation

Joris Heyman

PII: S0098-3004(18)31066-5

DOI: <https://doi.org/10.1016/j.cageo.2019.03.007>

Reference: CAGEO 4254

To appear in: *Computers and Geosciences*

Received Date: 7 November 2018

Revised Date: 13 February 2019

Accepted Date: 22 March 2019

Please cite this article as: Heyman, J., TracTrac: A fast multi-object tracking algorithm for motion estimation, *Computers and Geosciences* (2019), doi: <https://doi.org/10.1016/j.cageo.2019.03.007>.

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



TracTrac: a fast multi-object tracking algorithm for motion estimation

Joris Heyman
Univ Rennes, CNRS
Géosciences Rennes, UMR 6118
35000 Rennes, France.
joris.heyman@univ-rennes1.fr

March 28, 2019

1 **Abstract**

2 TracTrac is an open-source Matlab/Python implementation of a robust and efficient
3 object tracking algorithm capable of simultaneously tracking several thousands of
4 objects in very short time. Originally developed as an alternative to particle image
5 velocimetry algorithms for estimating fluid flow velocities, its versatility and robust-
6 ness makes it relevant to many other dynamic sceneries encountered in geophysics
7 such as granular flows and drone videography. In this article, the structure of the
8 algorithm is detailed and its capacity to resolve strongly variable and intermittent
9 object motions is tested against three examples of geophysical interest.

10 **Keywords: Videography, Motion Estimation, Feature Tracking**

11 **1 Introduction**

12 Owing to the popularization of digital cameras in the last 20 years, videography tech-
13 niques are increasingly used in the lab and in the field to measure velocities and
14 trajectories associated to a moving scenery. As earth processes are mostly dynamic,
15 imaging today appears as an affordable way to get spatio-temporal quantification of
16 these motions. Glacier motion, river flow, sediment transport, rock avalanches, wind
17 boundary layers, are some example of geophysical processes, whose understanding
18 rely deeply on how accurately their kinematics can be measured both in time and
19 in space. Yesterday restricted to laboratory studies with important experimental
20 apparatus (lasers, high speed cameras, computing clusters), flow imaging is now
21 expanding to in-situ monitoring of geophysical processes, notably thanks to the new
22 perspectives offered by drone videography [1]. In parallel, significant efforts have
23 been made in the computer vision community to improve and invent new image
24 processing algorithms treating efficiently these image sequences. Applications for
25 video surveillance (such as face recognition) and autonomous vehicles are among
26 the most spectacular achievements of these algorithms, running in real time [3, 14].
27 Curiously, few of these new methods were transferred into user-friendly, flexible and
28 open-source applications available for earth science researcher in their daily work
29 [5]. Processing images often costs much of the scientific effort instead of being a
30 powerful and direct mean to better understand natural processes. The present arti-
31 cle introduces TracTrac (see Computer Code Availability section), an open-source
32 Matlab/Python implementation of an original and efficient object tracking algorithm
33 capable of simultaneously tracking several thousands of objects in very short com-
34 putation time and very basic user knowledge. Its conception makes it equally good
35 for dealing with densely seeded fluid flows (typically treated with Particle Image
36 Velocimetry methods, PIV), granular flow, birds motion or any natural moving scene.
37 The article is structured as follows. After briefly reviewing the computational methods
38 that have been proposed for motion estimation in the past, TracTrac originality and
39 the algorithm structure is detailed. The accuracy and robustness of the algorithm is
40 then tested against a synthetic images occupied by artificial moving objects. Finally,
41 examples of TracTrac flow estimates are presented in real earth science applications
42 (turbulence, granular avalanche and bird flock).

43 **2 Advances in motion estimation techniques**

44 Literature about motion estimation from image sequence is vast and spreads over
45 several scientific disciplines, rendering difficult an exhaustive review. At least two
46 large families of methods have emerged: (i) the methods based on interrogation
47 windows, usually called Particle Image Velocimetry (PIV) and (ii) the methods based
48 on object detection and tracking, typically called Particle Tracking Velocimetry (PTV).
49 Although being not very informative, the choice of these acronyms refers to their

50 initial use in films of flows seeded by tracer particles to establish a map of local
51 velocities. The spectrum of applications of PIV and PTV methods is however much
52 broader, extending to complex moving sceneries. The general idea behind PIV is
53 to quantify motion by cross-correlation of interrogation windows [25]. Dividing
54 the image into smaller boxes (typically of 8 or 16 pixels width), the local motion is
55 obtained by searching the box displacement that maximizes the cross-correlation
56 product of box pixel light or color intensities between two consecutive frames. In
57 contrast, PTV consists in detecting the presence of special features in an image and
58 tracking them through consecutive frames. Special features, also called objects,
59 can be particles (blobs of bright or dark intensities), but also more complex shapes
60 (corners, ball, faces, ...). Where PIV outputs a velocity vector for each interrogation
61 box, PTV provides the trajectory of an object (its position in successive frames)
62 which can be then mapped into a grid to get a dense velocity field [22]. In other
63 words, PTV takes the Lagrangian point of view of motion where PIV is essentially an
64 Eulerian vision. Both techniques have advantages and disadvantages as shown in
65 the following. While originally preferred for its relative simplicity and robustness (a
66 few free parameters involved), PIV inevitably introduces some filtering at fluctuation
67 scales smaller than the box size, which preclude the correct estimation of steep
68 velocity gradients. Partial recovering of interrogation boxes help at increasing velocity
69 map resolution but do not solve the filtering effect. Uncertainties are thus particularly
70 high for flows near walls and turbulent flows in general for which the Kolmogorov
71 scale can be small. In contrast, PTV is only limited by the scale of the tracked features
72 (particles, gradients) as well as their local density so that instantaneous velocity maps
73 are less prone to the box filter effect [10, 11]. Both PIV and PTV dynamic ranges
74 strongly rely on the accurate detection of peaks in the frames (e.g., the location
75 of the feature to track for PTV and maximum cross-correlation product for PIV).
76 Methods have been proposed to reach sub-pixel accuracy in peak location, allowing
77 for the measurement of displacements smaller than one pixel per frame. However,
78 local saturation of images (values equal to 0 or 1) and small particle image size may
79 produce peak locking and biased velocity measurements [6, 15, 16, 19, 20, 23]. To
80 minimize these effects, attention has to be taken to the effective dynamic range
81 reached (high contrasts) and the magnification factor of the lens. PIV methods
82 typically overtake PTV methods if particle displacement becomes large compared to
83 the mean inter particle image distance. A so-called particle spacing displacement
84 ratio $p = \sqrt{S/N}/(v\Delta t)$ (with $v\Delta t$ the particle image displacement, N the number
85 of particles and S the image surface) has been proposed by [13] to describe this
86 effect. To avoid ambiguities in the reconstruction of trajectories, PTV algorithms
87 generally requires high frame rates or low particle image densities, that is $p \geq 1$. Thus,
88 PIV algorithms have often been preferred to probe densely seeded turbulent flows
89 where p can be small compared to 1. Combinations of the two methods have been
90 proposed to gain robustness in the case of large displacements (or large particle image
91 density) to limit the low-pass filtering effect of PIV [7, 21]. Another disadvantage of
92 PIV methods concerns complex sceneries made of moving and non-moving layers
93 (e.g. a flowing river on a fixed bed, a flock of flying birds through trees). The cross-
94 correlation procedure do not differentiate between layers so that the resulting velocity
95 is an average of the fixed and moving elements. In addition, incoherent motion

96 such as Brownian motion or multiple wave celerities cannot be handled by most
 97 PIV methods: at the scale of the interrogation window, the flow is supposed to be
 98 continuous and unidirectional. In contrast, sharp interfaces between moving and
 99 static regions, as well as non-coherent motions can in principle be rendered by PTV
 100 methods [5]. There is today a net enthusiasm for PTV algorithms due to their broader
 101 application range and their higher resolution [5, 9]. They are also directly transferable
 102 to stereoscopic camera setup where the position and trajectory of objects can be
 103 estimated in the three space dimensions [12, 13, 18]. However, most existing PTV
 104 algorithms still suffer from the aforementioned drawbacks: (i) they are limited to large
 105 particle spacing displacement ratio ($p > 2$ in [13] and in [5], while $p > 0.33$ in a recent
 106 study by [7]) and (ii) they are not computationally efficient when many features have
 107 to be tracked (a maximum of 4000 particles per time frame are considered in [7] and
 108 1000 in [5]).

109 **3 TracTrac**

110 The first innovation brought by TracTrac compared to traditional PTV methods is
 111 its efficiency. Indeed, TracTrac uses k -dimensional trees to search and compute
 112 statistics around neighbouring objects [4], allowing very high analysis frame rate
 113 even at large particle image number. The second key feature lie in an original asso-
 114 ciation process of objects between frames, that significantly decreases the number
 115 of erroneous trajectory reconstructions. This process is based upon a sequence of 3
 116 frames (instead of 2 for classical pair association) and a conservative rule rejecting
 117 any association ambiguities [21]. The third advantage of TracTrac is its capacity to
 118 deal with high feature densities at relatively low acquisition frequencies (p down
 119 to 0.25). This is achieved owing to a motion predictor step based on a local spatio-
 120 temporal average of the neighbouring object velocities. Differences between the
 121 motion prediction model and the observed displacement are systematically moni-
 122 tored, allowing filtering outliers based on local and adaptive statistics of the motion
 123 variability. This adaptive filter enables both the quantification of strongly incoherent
 124 motions (of the Brownian motion type [5]) and coherent displacement (governed by
 125 a spatio-temporal continuous deterministic velocity field) in the same image.

126 **3.1 Details of the algorithm**

127 TracTrac rests on three main modules: object detection, motion estimation, error
 128 monitoring.

129 **3.1.1 Object detection**

130 The first module regroups all the computing steps from the raw frame I_t to the
 131 detection of the position of moving objects \mathbf{x}_t . Most of these preprocessing steps
 132 are optional, and may be turned off by the user. The procedure is the following.
 133 First a median box filter can be applied to remove possible noise on I_t . The default
 134 size of this filter was set to 3×3 pixels. Second, the image is divided between a

135 “background” image B_t made of quasi-static regions, and a “foreground image” F_t ,
 136 formed by the moving regions. The latter is computed as $F_t = I_t - B_t$. This operation
 137 allows focusing on the moving part of a scene, while ignoring the static regions. B_t
 138 has to be recomputed at each frame. The method chosen here borrows from the
 139 so-called “median” background subtraction method, where the background image is
 140 taken to be a temporal moving average of pixel values:

$$B_t = \beta \text{sign}(I_t - B_{t-1}) + B_{t-1}, \quad (1)$$

141 where $\beta < 1$ is the background adaptation speed. A large value of β gives backgrounds
 142 that are rapidly adapted to the changes in scene luminosity. On opposite, a small value
 143 of β provides background images that are insensible to rapid luminosity changes.
 144 The default value is set to $\beta = 0.001$. The recurrence relation (1) requires to provide
 145 an initial guess for B_0 , which is computed from an average of the first $1/(2\beta)$ frames

$$B_0 = \frac{2}{\beta} \sum_{i=0}^{\beta/2-1} I_t. \quad (2)$$

146 It is worth noting that PTV methods are able to resolve sharp velocity gradients as
 147 well as out-of-plane velocity gradients so that background subtraction may not be
 148 always necessary. Objects are then identified in the foreground image by a so-called
 149 “blob detection” method. TracTrac integrates two state-of-the-art detectors, namely
 150 Difference of Gaussians (DoG) and Laplace of Gaussian (LoG), both depending on
 151 a single scale parameter δ . The DoG convolves the image with a filter constructed
 152 from the subtraction of two Gaussian of bandwidth 0.8δ and 1.2δ . It acts as a band-
 153 pass filter selecting blobs in the 20% scale range around $\delta\sqrt{2}$. The LoG approach
 154 first convolves the image with a Gaussian filter of bandwidth δ , then applying the
 155 Laplacian operator on the convoluted image. Both approaches yield a filtered image
 156 F'_t with a strong positive response in the presence of objects of scale δ . Positions
 157 of the object centroids $\{\mathbf{x}_i\}$ are obtained by searching for local maximum in F'_t . To
 158 minimize false detections, an intensity threshold ε is fixed under which maxima are
 159 ignored. In TracTrac, the default value of ε is fixed to half standard deviation above
 160 the mean luminosity of F'_t . Sub-pixel resolution of object position is achieved by
 161 fitting a quadratic or a Gaussian function to the pixel intensity values around the
 162 centroid position, and finding then the position of the maximum of this function. For
 163 instance, if a maximum is found in pixel i, j , the sub-pixel position of object will be

$$x = j + \frac{F'_{i,j+1} - F'_{i,j-1}}{2(F'_{i,j+1} - 2F'_{i,j} + F'_{i,j-1})}, \quad (3)$$

$$y = i + \frac{F'_{i+1,j} - F'_{i-1,j}}{2(F'_{i+1,j} - 2F'_{i,j} + F'_{i-1,j})}, \quad (4)$$

$$(5)$$

164 for a quadratic function. The formula is the same for a Gaussian function, replacing F'
 165 by $\ln(F')$. The ensemble of points $\mathbf{x}_i(t)$, $i = 1, \dots, N(t)$ made of the sub-pixel centroid
 166 positions are then tracked through time.

167 **3.1.2 Motion estimation**

168 As classical PTV algorithms [7, 13], motion estimation is achieved by associating
 169 detected objects between successive frames, typically by minimization of Euclidean
 170 distance. In TracTrac, at each time t , the set of $N(t)$ detected objects is organized
 171 into a 2-dimensional tree allowing for fast nearest neighbour search [4]. The nearest
 172 neighbours in successive frames are computed for both forward and backward time
 173 association:

- 174 • forward $\mathbf{x}_i(t) \rightarrow \mathbf{x}_j(t+1)$: for each $\mathbf{x}_i(t)$, find its closest neighbour in $\{\mathbf{x}(t+1)\}$,
- 175 • backward $\mathbf{x}_i(t+1) \rightarrow \mathbf{x}_j(t)$: for each $\mathbf{x}_i(t+1)$, find its closest neighbour in
 176 $\{\mathbf{x}(t)\}$.

177 Since objects usually appear and disappear through frame, both computations may
 178 give different results. In order to minimize false associations, only the unequivocal
 179 pairs are kept (i.e., the pairs that point to the same objects regardless of the time
 180 direction of association). In doing so, ambiguous associations are automatically
 181 disregarded. A fragment of trajectory (“tracklet”) is defined if two consecutive and un-
 182 unequivocal associations are made for the same object. In other words, when a position
 183 triplet $x_i(t-1) \leftrightarrow x_j(t) \leftrightarrow x_k(t+1)$ is found without ambiguity, it is considered as a
 184 valid fragment of trajectory, to which is associated a new or existing ID (depending
 185 on whether the object has been already associated to a trajectory ID in the frame
 186 $t-1$). This 3-frame association technique reduces significantly the occurrence of
 187 bad associations. In addition, it enables the computation of second order object
 188 velocities (via central differences) as well as their accelerations:

$$\hat{\mathbf{v}}(t-1) = \frac{\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}(t-2)}{2\Delta t}, \quad (6)$$

$$\hat{\mathbf{a}}(t-1) = \frac{\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}(t-2) + 2\hat{\mathbf{x}}(t-1)}{\Delta t^2} \quad (7)$$

189 This technique does not increase the computational cost significantly since nearest
 190 neighbour associations ($x_j(t) \leftrightarrow x_k(t+1)$) are saved for the following time step. In
 191 the following, the variables pertaining to objects that were associated into tracklets
 192 in the frame t are denoted by a hat (i.e., $\hat{\mathbf{x}}(t)$, $\hat{N}(t)$). The quality of this association
 193 step is often constrained by the maximum object displacement between consecutive
 194 frames, or equally the maximum object velocity divided by the frame rate of the
 195 camera. Indeed, erroneous associations spontaneously arise from aliasing effects
 196 when object displacement is comparable to the average distance separating objects
 197 (for instance, points on a line distant by 10 pixels that travel at 10 pixels per frame
 198 will appear having a null velocity). Motion recognition is relatively easy when $p^{-1} =$
 199 $v\Delta t\sqrt{N/S} \ll 1$ (or equally when $p \gg 1$ [13]). In TracTrac, this condition is relaxed by
 200 the use of a predictor step based on a motion model inherited from previous time
 201 step [7, 17]. In other words, TracTrac first predicts the position objects in the following
 202 frame and then use this prediction to perform the following association. At time
 203 t , the motion model is based on the pool of objects associated to tracklets at $t-1$,
 204 their velocities $\hat{\mathbf{v}}(t-1)$ and their motion predictors $\hat{\hat{\mathbf{v}}}(t-1)$ (where the bar symbol

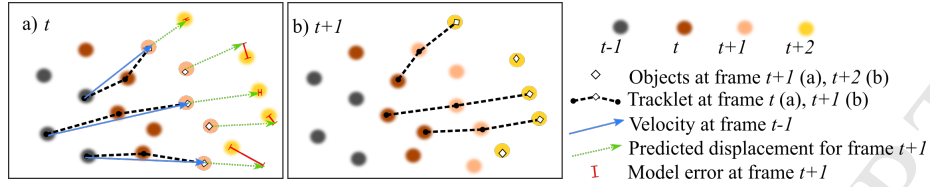


Figure 1: Prediction-Association process between frames t , $t+1$ and $t+2$

stands for quantities related to the motion model). This information is passed to all objects detected in the current frame t by a weighted average over their k -th nearest neighbours taken from the aforementioned pool:

$$\bar{\mathbf{v}}_i(t) = \frac{1}{\min(k, \hat{N})} \sum_{j=1}^{\min(k, \hat{N})} \alpha \hat{\mathbf{v}}_{i,j}(t-1) + (1-\alpha) \hat{\mathbf{v}}_{i,j}(t-1). \quad (8)$$

The weight $\alpha \in [0, 1]$ introduces a finite temporal relaxation of the predicted velocities. For $\alpha \rightarrow 1$, the motion model is only based on the immediate previous frame, while for $\alpha \rightarrow 0$, history of the velocities predicted in earlier frames are used to compute the motion model. Averaging over the k -th nearest neighbours has two advantages. First, it allows filtering the smallest spatial variations of velocities, that can be influenced by noise or erroneous tracklets associations. Second, it naturally adapts to the local density of objects, in contrast to fixed-size kernel smoothing methods: the larger the density, the smaller the filtering scale, and the finer the prediction. Once the motion model is computed, new object position is predicted assuming zero acceleration:

$$\bar{\mathbf{x}}_i(t) = \mathbf{x}_i(t-1) + \bar{\mathbf{v}}_i(t)\Delta t, \quad (9)$$

and the association process is performed by searching among the nearest neighbours between $\bar{\mathbf{x}}(t)$ and $\mathbf{x}(t+1)$ (Fig. 1). These new tracklets can either be saved and the following frame proceed, or used iteratively to refine the motion model and predict once again object displacement. The predictor step is thus implemented as an iterative sequence, using the temporary recovered tracklets as additional velocity vectors considered in the motion model. Convergence is generally obtained after a few iterations, the number of associated tracklet reaching a maximum. Once the desired number of iteration is reached, computation continues with the following frame.

3.1.3 Error monitoring and outliers filtering

The motion model used in TracTrac enables a continuous monitoring of the difference between predicted and actual displacements. This information is of particular value since it helps to eliminate outliers from the obtained associations based on statistical criterion. For each unequivocal associations, the log-error norm between the predicted and the true velocity vector is

$$\epsilon_i(t) = \log \left(\left\| \hat{\mathbf{v}}_i(t-1) - \hat{\mathbf{v}}_i(t) \right\| \right). \quad (10)$$

232 which may be considered a real valued spatio-temporal random variable of approxi-
 233 mately Gaussian shape (while the error norm would be log-normal since positive).
 234 Negative ϵ corresponds to high motion model accuracy. The probability distribution
 235 of ϵ depends on the spatial and temporal variability of the background flow to be
 236 measured as well as the quality of the zero acceleration approximation for the motion
 237 model. The local mean model error $\bar{\epsilon}_i$ around the object i is estimated by sampling
 238 log-errors over its k -nearest objects, in the same time as determining model velocities
 239 (8):

$$\bar{\epsilon}_i(t) = \frac{1}{\min(k, \hat{N})} \sum_{j=1}^{\min(k, \hat{N})} \alpha \hat{\epsilon}_{i,j}(t) + (1 - \alpha) \hat{\epsilon}_{i,j}(t). \quad (11)$$

240 The standard deviation of the error around the mean is estimated on the whole
 241 computation window by

$$\bar{\sigma}_\epsilon(t) = \alpha \sqrt{\frac{1}{N(t)} \sum_{i=1}^{N(t)} \left(\hat{\epsilon}_i(t) - \frac{1}{N(t)} \sum_{j=1}^{N(t)} \hat{\epsilon}_j(t) \right)^2} + (1 - \alpha) \bar{\sigma}_\epsilon(t-1), \quad (12)$$

242 Outliers are then detected from tracklets which have $\epsilon_i(t) - \bar{\epsilon}_i(t) > n_\sigma \bar{\sigma}_\epsilon(t)$, with
 243 $n_\sigma \in \mathbb{R}$ a parameter chosen by the user. For instance, $n_\sigma = 1.96$ ensures that all
 244 associated tracklets remain in the 95% confidence interval provided by the model. In
 245 contrast, for $n_\sigma = -1.96$ only remains the 5% of tracklets that best fit the prediction
 246 model.

247 3.1.4 From tracklets to trajectories

248 To each new associated tracklet is given a trajectory ID number. If in the following
 249 frame, a tracklet is found with an object already having an ID, the latter is applied
 250 to the tracklet. This information handover allows reconstructing the whole object
 251 trajectories from elementary tracklets sharing the same ID. At each frame, the infor-
 mation about tracklets are saved by TracTrac in an array with columns: At the end of

Column Number	1	2	3	4	5	6	7	8	9	10	11
Variable	t	ID	\hat{x}_i	\hat{y}_i	\hat{u}_i	\hat{v}_i	$\hat{a}_{x,i}$	$\hat{a}_{y,i}$	\hat{u}_i	\hat{v}_i	$\hat{\epsilon}_i$

Table 1: Correspondence between columns and variables in the TracTrac output ASCII file “*_track.txt”

252 the computation, this array can be saved either in ASCII or in binary format (mat-file
 253 in Matlab, and hdf5 in python). This file is automatically named according to the
 254 video file name with the suffix “*_track.txt”.

256 3.2 User interface

257 The Matlab version of TracTrac includes a graphical interface (GUI) enabling rapid
 258 tracking results for non-expert users. In practice, it can be used to test and optimize

259 the free parameters meanwhile observing in real time their effect on the quality of
 260 the tracking process. In contrast to the Matlab GUI, the Python version of TracTrac
 261 can be launched either as a Python script or as a Python function. This command
 262 line control allows treating iteratively several videos or integrating TracTrac directly
 263 into Python scripts (a list of video files can also be chosen in the Matlab GUI). Full
 264 compatibility is maintained between the two implementations owing to a common
 265 input parameter file whose structure is given as the supplementary material. Details
 266 about the Matlab GUI and the Python commands are also provided in this document.

267 4 Results and discussion

268 4.1 Synthetic flow

269 In order to test TracTrac performances, synthetic images were created, enabling a
 270 comparison of the algorithm predictions with known object trajectories. The flow
 271 was chosen in order to test the algorithm robustness for both strongly unsteady and
 272 non-uniform continuous flow field.

273 4.1.1 Flow description

274 The synthetic trajectories are initiated by N points randomly distributed in the image
 275 (x_0, y_0) . At each frame, a synthetic image is build by applying a Gaussian kernel
 276 of fixed standard deviation on each object centroid. Uncorrelated noise is then
 277 added to the image pixels with an intensity depending on the signal-to-noise ratio
 278 (SNR) chosen (Fig. 2). An image is created at each frame, while advecting the objects
 279 according to the following two consecutive operations: a first one operating in radial
 280 coordinates $(r = \sqrt{x^2 + y^2}, \theta = \tan^{-1}(y/x))$:

$$r_{n+1} = r_n, \quad (13)$$

$$\theta_{n+1} = \theta_n + 4\delta \cos(n\pi/50) \exp(-0.5(r_n/80)^2) - 10\delta \cos(n\pi/25) \exp(-0.5(r_n/50)^2), \quad (14)$$

281 followed by a second step in cartesian coordinates $(x = r \cos \theta, y = r \sin \theta)$:

$$x_{n+1} = x_n + 2\delta \sin(\pi n/100) * y_n + \xi_n, \quad (15)$$

$$y_{n+1} = y_n + \delta x_n + \xi_n, \quad (16)$$

282 where ξ_n is a white noise term whose intensity will be varied. In practice, the time
 283 step $\delta = 0.01$ is chosen to get displacement lengths in the range 0-20 pixels per frames.
 284 Cartesian and polar coordinate systems are centred on the image centre. Periodic
 285 boundary conditions are applied when a point leaves the image field. Snapshot of the
 286 flow velocity vectors were plotted on Fig. 3. Before the next frame, a finite number of
 287 points (0-50%) are associated new coordinates in order to mimic in and out of plane
 288 motion.

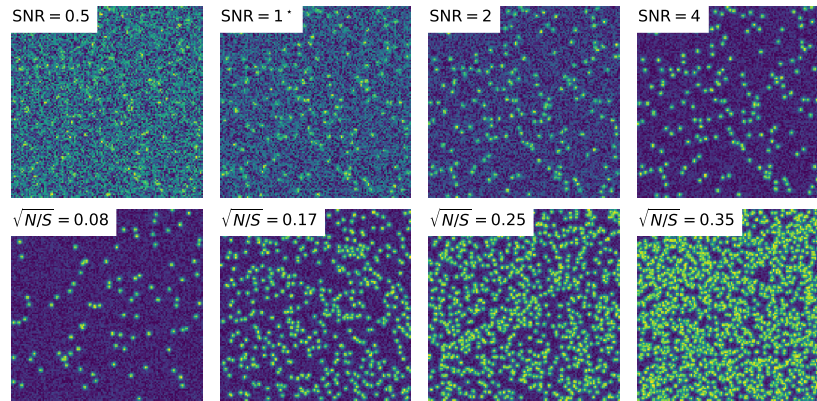


Figure 2: Effect of Signal to Noise Ratio (top) and particle number density on the generated images.

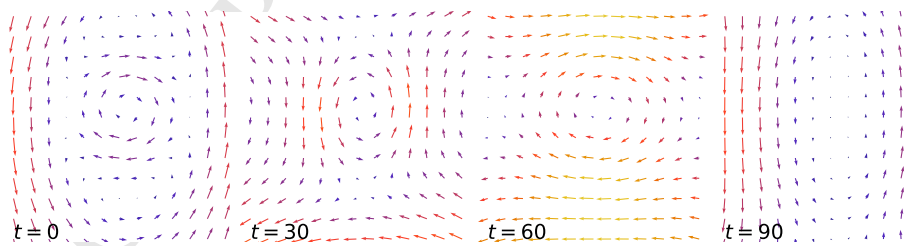


Figure 3: Space dependence of the synthetic vortex flow considered (Eq. 8) at various time instants.

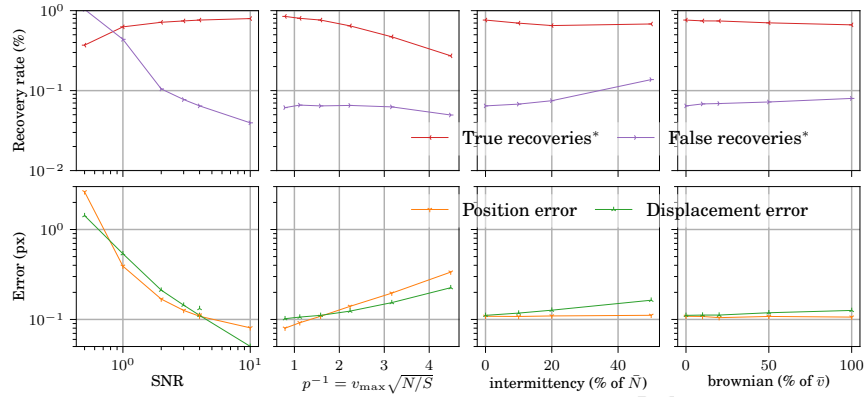


Figure 4: Algorithm accuracy function of 4 variables: Signal to Noise Ratio (SNR), object separation compared to mean displacement ($p^{-1} = \bar{v}\sqrt{N/S}$, with \bar{v} the mean displacement and N the number of objects in the area S), object appearance and disappearance (“Intermittency”) and velocities random fluctuations (“Brownian”). True recoveries rates are defined according to a maximum position error of d and a maximum displacement error of $v/2$. False recoveries are found for the opposite criterion. Default parameters are: SNR=4, $N=1000$, Intermittency=0 and Brownian=0.

289 4.1.2 Accuracy

290 Accuracy was measured owing to 4 indexes: mean percentage of true and false
 291 object detection as well as mean absolute error in object position and displacement
 292 estimation. These indexes were computed for various image qualities and flow
 293 properties. To better isolate TracTrac performances, no pre-processing step was
 294 performed on the synthetic images (e.g. background subtraction, median filter). First,
 295 TracTrac accuracy is compared to the Signal to Noise Ratio (SNR, defined as blob
 296 peak magnitude over magnitude of an underlying uniform noise). Results presented
 297 in Fig. 4 show that increasing SNR significantly increases tracking quality: for $\text{SNR} \geq 4$
 298 (a typical value in PIV experiments), less than 5% of false detections are made, while
 299 mean position and displacement error are below 0.2 pixels, a value comparable with
 300 recent PTV methods [7]. Another quality factor is given by the ratio of maximum
 301 displacement length to the mean distance between neighbour objects expressed as
 302 $p^{-1} = \bar{v}_{\max}/\sqrt{N/S}$ (the inverse of the ratio defined by [13]). PTV algorithm are usually
 303 limited to $r \ll 1$ to avoid object association ambiguities between frames [13, 17].
 304 Thanks to the motion predictor, the association rule, and the outlier filter, false
 305 detections remain below 6% for ratios $p^{-1} \approx 4.5$, while position and displacement
 306 error are below 0.5 pixels. To the author knowledge, such large values of p^{-1} have not
 307 yet been reported in the literature.

308 Appearance and disappearance of objects through time, referred to as “intermit-
 309 tency” in Fig. 4, often occur due to out of transverse velocities in 3D flows observed on
 310 2D planes. While this phenomenon complicates the association process, the number
 311 of false tracklets remains limited to 12% at high intermittency levels (50% of the object

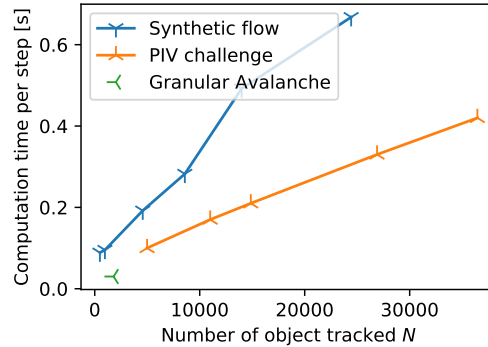


Figure 5: Computation time per video frame depending on the number of objects tracked. Computations are made with the Python implementation of TracTrac on a HP ELiteBook 840 laptop with processor Intel i7.

312 disappearing at each frame), suggesting a good adaptation of the algorithm to out of
 313 plane motions. While intermittency does not affect position error, it increases slightly
 314 the mean displacement error (owing to false associations, with mean displacement
 315 errors smaller than 0.5 pixel for a level of intermittency of 50%).

316 The last factor considered is the stochasticity of the underlying flow field, which
 317 cannot be predicted by deterministic motion predictors [5]. To investigate this effect,
 318 white noise was summed to object velocities in proportion of the deterministic flow
 319 velocity magnitude. Fig. 4 shows that both false detection and displacement error
 320 remain limited for fluctuations levels comparable with the average magnitudes (9%
 321 and 0.3 pixel respectively). This good performance is ensured by the continuous and
 322 local monitoring of prediction errors, that allows the computation of a local threshold
 323 to filter outliers, threshold which is directly influenced by the local motion statistics.

324 4.1.3 Efficiency

325 As shown by Fig. 5, TracTrac algorithm provides a computational time that grows only
 326 linearly with the number of object to track. This is mainly due to the implementation
 327 of k-d tree structures for nearest neighbour search. This allows for 25000 objects to
 328 be tracked in less than 0.7 seconds per frame (Fig. 5).

329 5 Application to geosciences

330 In this section, TracTrac specificities are highlighted through 3 examples of particular
 331 interest to geoscientists. We provide a comparison of TracTrac results with another
 332 open-source PIV software, PIVlab [24]. The latter was parametrized to compute
 333 velocity fields on 16×16 pixel interrogation windows.

334 5.1 Turbulent flow

335 The first example concerns 1000 frames of a turbulent duct flow past a series of hills,
 336 similar to aquatic bedforms or aeolian dunes (Fig.6a). The data was presented in
 337 the 4-th PIV challenge as an example of time dependent flow with strong velocity
 338 gradients and out of plane motions (intermittency) [9]. It is available online at
 339 www.pivchallenge.org/pivchallenge4.html. The flow is visualized through a 2-D
 340 laser sheet which illuminates seeded particles in a plane. TracTrac processing on
 341 such data can be appreciated in Fig. 6b and in the supplementary video online.
 342 This test case provides an example of the algorithm capabilities to compute time-
 343 average Eulerian flow quantities within high resolution (average flow magnitude
 344 and turbulent kinetic energy are presented in Fig.6c). In the dark regions where no
 345 object could be detected, the average values are kept empty (e.g., blank pixels in the
 346 right size of Fig.6c). The accuracy of the algorithm is specifically demonstrated by
 347 Fig. 6d where the streamwise time-average velocity profile close to the above wall
 348 is plotted. The profile closely follows the expected logarithmic law of the wall over
 349 several measurement points, and allows deducing the value of the local wall shear
 350 velocity ($u_* \approx 0.035$ m/s). While comparable to the TracTrac values in the bulk flow,
 351 the PIVlab-computed time-average velocity profile do not allow a clear identification
 352 of the inertial layer where the log-law applies. This is caused by the filtering effect
 353 imposed by the interrogation windows which bias velocity gradients close to the wall
 354 boundary. Wall boundary layer typically present a linear increase of the shear stress
 355 while approaching the wall, which permits deducing the shear velocity independently
 356 of the log-law of the wall. In the inertial layer, the total shear stress $\tau = u_*^2 \rho$ (ρ
 357 is the water density) is approximated by the turbulent stresses $-\rho \langle u' v' \rangle$ (viscous
 358 stresses $\nu \rho \partial_y \langle u \rangle$ are negligible outside of the viscous layer, $\nu = 10^{-6} \text{m}^2 \text{s}^{-1}$ being the
 359 kinematic viscosity of water). Extrapolation of the Reynolds stresses at the wall thus
 360 provides an estimation of the shear velocity $u_* = \sqrt{\tau/\rho}$. Fig. 6d shows that TracTrac
 361 predicts a similar wall shear velocity by this method, confirming its ability to measure
 362 precisely all the contributing scales of turbulence. In contrast, the Reynolds stresses
 363 predicted by PIVlab, while qualitatively similar, are much smaller than TracTrac
 364 values. This is once again an effect of the low-pass filter imposed by interrogation
 365 windows. This analysis is confirmed by a comparison of the root mean squared (RMS)
 366 streamwise velocity profile at $x = 100\text{px}$ with the average of several PIV software
 367 presented in [9]. Fig. 7 shows that TracTrac RMS are significantly higher than typically
 368 measured by traditional PIV software, including PIVlab.

369 Finally, it is worth pointing that the sub-pixel resolution of TracTrac algorithm also
 370 enables the observation of the viscous sub-layer in the mean velocity profile (Fig. 6d,
 371 at $yu_*/\nu < 30$). The latter has a theoretical size of $\nu/u_* \approx 28\mu\text{m}$, which corresponds
 372 to 0.15 pixel in the images and can thus, in theory, be visualized by TracTrac.

373 Computational time for the hill test case were reported in Fig. 5. In this figure,
 374 the level of the object detection threshold was sequentially decreased to artificially
 375 increase the number of detected object and confirm the quasi linear dependence of
 376 the computational time on object number. PIVlab takes about 2 seconds to compute
 377 15741 velocity vectors at each time step, where TracTrac takes less than 0.5 seconds
 378 to provide the double of vectors. A factor 8 is thus observed between computation

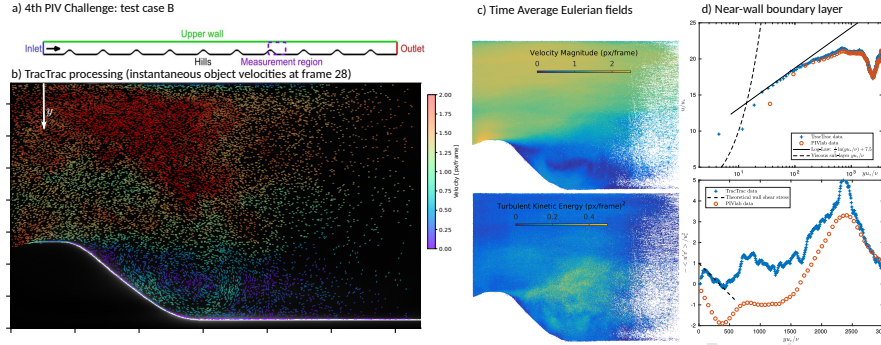


Figure 6: TracTrac results on the 4-th PIV Challenge data of time resolved turbulent flow past a hill [9]. (a) Geometry of the flow. (b) Instantaneous object velocities obtained by TracTrac. (c) Time average Eulerian field: velocity magnitude (up) and turbulent kinetic energy (down). (d) Turbulent profiles close to the wall above the hill: mean streamwise velocity (up) and Reynolds stresses (down)

379 time of PIVlab and TracTrac.

380 5.2 Granular avalanche

381 The second example focuses on the avalanche of granular material (glass beads of
 382 1mm diameter) along an inclined plate confined between two lateral walls (Fig. 8a).
 383 The experiment was made at Institut de Physique de Rennes, France, as part of a larger
 384 project aiming at modelling the rheology of dense inertial flow of granular media [8].
 385 The purpose of this example is to highlight the role of the motion predictor step and
 386 the associated monitoring of prediction errors to resolve locally heterogenous flow
 387 regions. In this experiment, the image density of objects is about 0.13 object/pixels,
 388 with displacements up to 6 pixels/frames, giving locally a ratio $p^{-1} = \bar{v}/\sqrt{N/S} \approx 0.8$.
 389 Instantaneous top and side views of the granular flow are shown on Fig. 8b with
 390 a color scale proportional to the monitored error between motion prediction and
 391 corrected displacement, showing local variations in the error values. As beads are
 392 generally bouncing against the walls, these regions present higher deviations from
 393 the mean motion than the bulk of the flow. This is confirmed by transverse and
 394 vertical profiles (Fig. 8c) that show higher average prediction errors on the side walls
 395 and at the bottom of the plate (at $z = 160\text{px}$) than in the bulk of the flow. This increase
 396 is also observed in the mean kinetic agitation (defined here as $\sqrt{u'^2 + v'^2}$).

397 By continuously monitoring the local mean prediction error, the algorithm gen-
 398 uinely adapts to the Brownian nature of object motion close to the side walls. As
 399 a consequence, the threshold for outlier filtering (see Sec. 3.1.3), $\bar{\epsilon} + 1.5\sigma_{\epsilon}$, locally
 400 adapts to the flow characteristics and allows for an correct estimation of object mo-
 401 tion statistics in all regions.

402 In contrast, PIVlab underestimates the kinetic agitation of the flow close to the

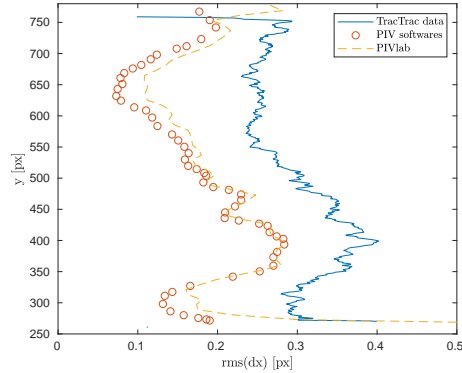


Figure 7: Root mean square streamwise velocity profiles estimated at the transverse section $x = 100\text{px}$ of the hill test case. The average of five PIV algorithms (Dantec, DLR, INSEAN, IOT and IPP) are represented with circles (adapted from the Fig. 21 of [9]), together with the PIVlab estimates (dashed line) and TracTrac values (blue line).

403 side walls (Fig. 8c). An advantage of PTV over PIV also appears in the low density
 404 gaseous region that develops above the dense granular flow in the bottom view
 405 (for $x = 0$ to 75px). In this region, the kinetic agitation estimated by PIV increases
 406 artificially because interrogation windows are often empty, leading to erroneous
 407 velocity estimates. This effect is not occurring in TracTrac, since velocity is computed
 408 in a Lagrangian basis only where objects are detected.

409 5.3 Bird flock

410 In the last example, the fly of a bird flock recorded by Attanasi et al. [2] is used to
 411 highlight the versatility of the algorithm and its robustness for many types of motions
 412 (Fig. 9). In this example, bird motion is three-dimensional so that, in the image, bird
 413 trajectories can occlude each other. However, TracTrac rules out fake connections
 414 when ambiguity arises in the nearest neighbour association, producing sure track-
 415 lets. These tracklets can then be recombined with cost optimization algorithm to
 416 reconstruct each individual entire trajectory.

417 Another aspect well highlighted by this example is the equal ability of a single
 418 size, isotropic convolution kernel (here the differential of Gaussians) to predict the
 419 velocity of objects that are not always of isotropic neither Gaussian shape (the birds
 420 wings for instance). It is particularly true in videos where moving features are not
 421 particles as in the two first examples, but consist of a deforming texture (the water
 422 surface of a flowing river for instance). In these situations, an isotropic convolution
 423 will still be able to isolate local features of interest in the image; features which can be
 424 tracked to provide an estimation of local velocities. In general, it is enough for images
 425 to have strong, dense and aleatory intensity gradients to provide good features to
 426 track, and reliable tracking results.

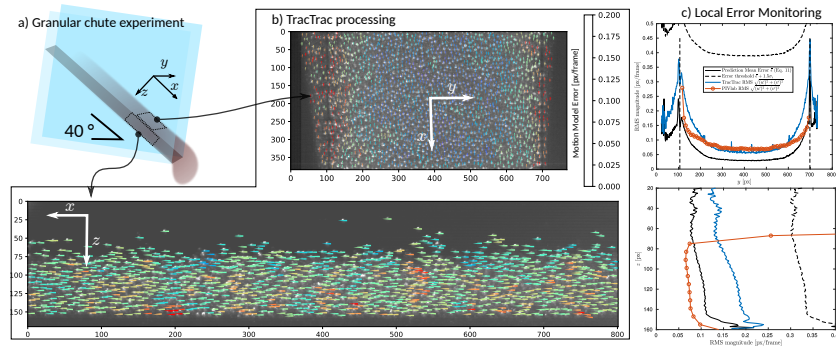


Figure 8: TracTrac genuine error monitoring revealed by a granular avalanche experiment.

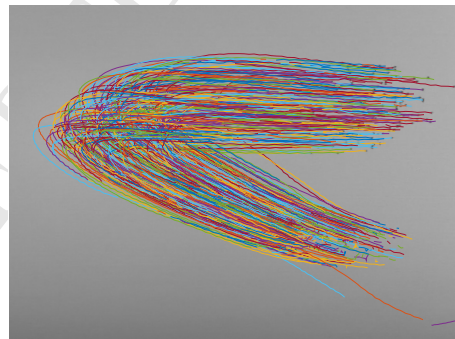


Figure 9: Birds trajectories obtained by TracTrac superimposed on the video of bird flock by Attanasi et al. [2]

6 Conclusion

In this article, I present an open source PTV algorithm called TracTrac, dedicated to motion recognition in geophysics. The main advantages of this algorithm are

1. A fast implementation through k-d tree nearest neighbour search, enabling the use of PTV for applications usually restricted to PIV.
2. An iterative prediction-correction procedure capable of following large object displacements in fluctuating and heterogeneous flow fields ($p^{-1} = 4.5$).
3. A robust 3-frame association process that limits velocity bias.

In particular, it has been shown that the algorithm provides much higher details of turbulent statistics than other open-source PIV software [24]. This result is crucial, since the measure of microscopic velocity fluctuations and sharp local gradients are often essential to correctly model geophysical processes (in turbulence and granular flows for instance).

All TracTrac source files are freely available (see Computer Code Availability section). Among the possible future developments, 3-dimensional tracking via stereoscopic videography may be easily implemented in the current algorithm. Other improvements such as the recognition of size and other specific features of objects can provide stronger constraints to the association process without increasing significantly the computation time.

Computer Code Availability

The TracTrac Matlab and Python source code are freely available at <https://perso.univ-rennes1.fr/joris.heyman/trac-trac-source.zip>. Compiled versions are also available on SourceForge at <https://sourceforge.net/projects/trac-trac>.

Acknowledgments

I am deeply acknowledging Alexandre Valance, Hervé Tabuteau, Renaud Delannay and Philippe Boltzenhagen (Institut de Physique de Rennes) for funding the scientific project involving the granular chute experiment. I am also grateful to the 3 anonymous reviewers whose comments significantly contributed to improve the manuscript.

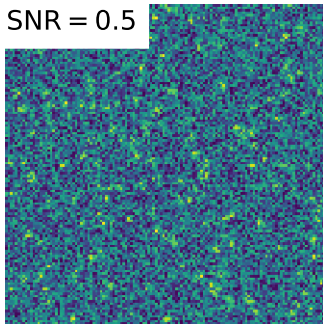
References

- [1] A. A. Aguirre-Pablo, M. K. Alarfaj, E. Q. Li, J. F. Hernandez-Sanchez, and S. T. Thoroddsen. Tomographic particle image velocimetry using smartphones and colored shadows. *Scientific Reports*, 7(1):3714, 2017.

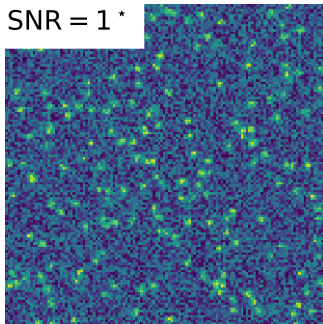
- 460 [2] A. Attanasi, A. Cavagna, L. Del Castello, I. Giardina, T.S. Grigera, A. Jelic, S. Melillo,
461 L. Parisi, O. Pohl, E. Shen, and M Viale. Information transfer and behavioural
462 inertia in starling flocks. *Nature Physics*, 10:691, Jul 2014.
- 463 [3] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple
464 instance learning. In *2009 IEEE Conference on Computer Vision and Pattern
465 Recognition*, p983–990, Jun 2009.
- 466 [4] J. L. Bentley. Multidimensional binary search trees used for associative searching.
467 *Commun. ACM*, 18(9):509–517, Sep 1975.
- 468 [5] N. Chenouard, I. Smal, F. de Chaumont, M. Maška, I.F. Sbalzarini, Y.o Gong,
469 J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R Cohen, W. J Godinez, K.
470 Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. E G Magnusson, J.
471 Jaldén, H. M Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.-Y.
472 Tinevez, S. L Shorte, J. Willemse, K. Celler, G. P van Wezel, H.-W. Dan, Y.-S. Tsai,
473 C. O. de Solórzano, J.-C. Olivo-Marin, and E. Meijering. Objective comparison
474 of particle tracking methods. *Nature Methods*, 11:281, Jan 2014.
- 475 [6] K. T. Christensen. The influence of peak-locking errors on turbulence statistics
476 computed from piv ensembles. *Experiments in Fluids*, 36(3):484–497, Mar 2004.
- 477 [7] C. Cierpka, B. Lütke, and C. J. Kähler. Higher order multi-frame particle tracking
478 velocimetry. *Experiments in Fluids*, 54(5):1533, May 2013.
- 479 [8] J. Heyman, P. Boltenhagen, R. Delannay, and A. Valance. Experimental investi-
480 gation of high speed granular flows down inclines. *EPJ Web Conf.*, 140:03057,
481 2017.
- 482 [9] C. J. Kähler, T. Astarita, P. P. Vlachos, J. Sakakibara, R. Hain, S. Discetti, R. La Foy,
483 and C. Cierpka. Main results of the 4th international piv challenge. *Experiments
484 in Fluids*, 57(6):97, May 2016.
- 485 [10] C. J. Kähler, S. Scharnowski, and C. Cierpka. On the resolution limit of digital
486 particle image velocimetry. *Experiments in Fluids*, 52(6):1629–1639, Jun 2012.
- 487 [11] C. J. Kähler, S. Scharnowski, and C. Cierpka. On the uncertainty of digital piv and
488 ptv near walls. *Experiments in Fluids*, 52(6):1641–1656, Jun 2012.
- 489 [12] H. G. Maas, A. Gruen, and D. Papantoniou. Particle tracking velocimetry in
490 three-dimensional flows. *Experiments in Fluids*, 15(2):133–146, Jul 1993.
- 491 [13] N. A. Malik, Th. Dracos, and D. A. Papantoniou. Particle tracking velocimetry in
492 three-dimensional flows. *Experiments in Fluids*, 15(4):279–294, Sep 1993.
- 493 [14] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for
494 driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelli-
495 gent Transportation Systems*, 7(1):20–37, Mar 2006.
- 496 [15] D. Michaelis, D. R Neal, and B. Wieneke. Peak-locking reduction for particle
497 image velocimetry. *Measurement Science and Technology*, 27(10):104005, 2016.

- 498 [16] H. Nobach, N. Damaschke, and C. Tropea. High-precision sub-pixel interpo-
499 lation in particle image velocimetry image processing. *Experiments in Fluids*,
500 39(2):299–304, Aug 2005.
- 501 [17] K. Ohmi and H-Y. Li. Particle-tracking velocimetry with new algorithms. *Mea-
502 surement Science and Technology*, 11(6):603–616, May 2000.
- 503 [18] N. T. Ouellette, H. Xu, and E. Bodenschatz. A quantitative study of three-
504 dimensional lagrangian particle tracking algorithms. *Experiments in Fluids*,
505 40(2):301–313, Feb 2006.
- 506 [19] E.F.J. Overmars, N.G.W. Warncke, C. Poelma, and J. Westerweel. Bias errors in
507 piv: the pixel locking effect revisited. In *15th Int Symp on Applications of Laser
508 Techniques to Fluid Mechanics*, Lisbon, Portugal, 05-08 Jul 2010.
- 509 [20] T. Roesgen. Optimal subpixel interpolation in particle image velocimetry. *Ex-
510 periments in Fluids*, 35(3):252–256, Sep 2003.
- 511 [21] D. Schanz, S. Gesemann, and A. Schröder. Shake-the-box: Lagrangian particle
512 tracking at high particle image densities. *Experiments in Fluids*, 57(5):70, Apr
513 2016.
- 514 [22] J.F.G. Schneiders, I. Azijli, F. Scarano, and R.P. Dwight. Pouring time into space.
515 In *11th International Symposium on Particle Image Velocimetry, PIV15*, Santa
516 Barbara, CA, USA, 2015.
- 517 [23] I. Smal, M. Loog, W. Niessen, and E. Meijering. Quantitative comparison of
518 spot detection methods in live-cell fluorescence microscopy imaging. In *2009
519 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*,
520 1178–1181, Jun 2009.
- 521 [24] W. Thielicke and E. J. Stamhuis. PIVlab – towards user-friendly, affordable and
522 accurate digital particle image velocimetry in MATLAB. *Journal of Open Research
523 Software*, 2, Oct 2014.
- 524 [25] J. Westerweel, G. E. Elsinga, and R. J. Adrian. Particle image velocimetry for
525 complex and turbulent flows. *Annual Review of Fluid Mechanics*, 45(1):409–436,
526 2013.

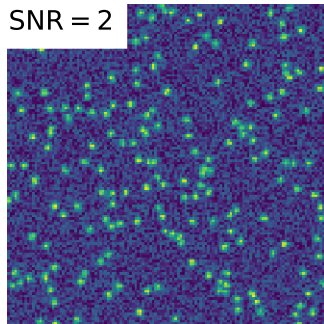
SNR = 0.5



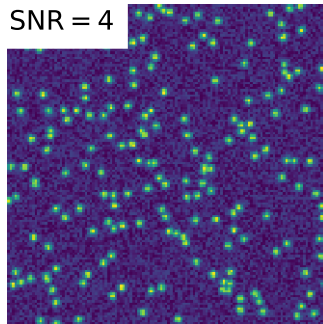
SNR = 1 *



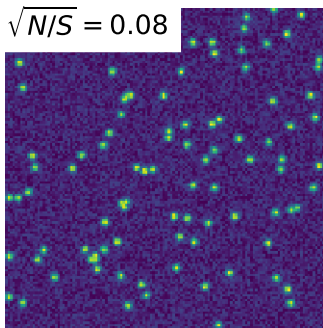
SNR = 2



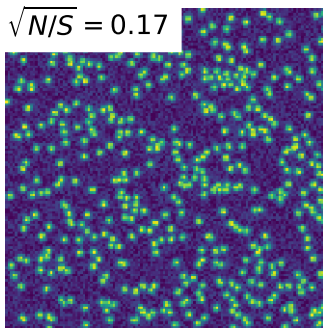
SNR = 4



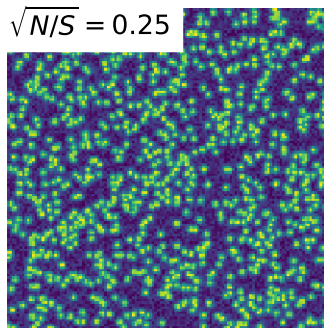
$\sqrt{N/S} = 0.08$



$\sqrt{N/S} = 0.17$



$\sqrt{N/S} = 0.25$



$\sqrt{N/S} = 0.35$

