Feature-based Analysis of Plasma-based Particle Acceleration Data

Oliver Rübel [1], Cameron G.R. Geddes [2], Min Chen [2], Estelle Cormier-Michel [3] ,
and E. Wes Bethel [1]

[1] O. Rübel and E.W. Bethel are with the Visualization Group, Computational Research Division, Lawrence Berkeley National Laboratory (LBNL)

[2] C.G.R. Geddes and Min Chen are with the Lasers, Optical Accelerator Systems Integrated Studies (LOASIS) program of the Accelerator and Fusion Research Division, LBNL

[3] E. Cormier-Michel is with Tech-X Corp.

DISCLAIMER: This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Acknowledgements: This work was supported by the Director, Office of Science, Offices of High Energy Physics and Advanced Scientific Computing Research of the U.S. Department of Energy under Contract No. DE-AC02-04CH11231 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET) and the COMPASS project. The colliding pulse simulations were supported by DOE, NNSA, NA-22. This research used resources of NERSC which is supported by the Office of Science of the U.S. Department of Energy under Contract NO. DE-AC02-05CH11231. We thank the FastBit, H5Part, VORPAL and VisIt team for ongoing development and maintenance efforts.

Feature-based Analysis of Plasma-based Particle Acceleration Data

Oliver Rübel, Cameron G.R. Geddes, Min Chen, Estelle Cormier-Michel, and E. Wes Bethel

Abstract

Plasma-based particle accelerators can produce and sustain thousands of times stronger acceleration fields than conventional particle accelerators, providing a potential solution to the problem of the growing size and cost of conventional particle accelerators. To facilitate scientific knowledge discovery from the ever growing collections of accelerator simulation data generated by accelerator physicists to investigate next-generation plasma-based particle accelerator designs, we describe a novel approach for automatic detection and classification of particle beams and beam substructures due to temporal differences in the acceleration process, here called acceleration features. The automatic feature detection in combination with a novel visualization tool for fast, intuitive, query-based exploration of acceleration features enables an effective top-down data exploration process, starting from a high-level, feature-based view down to the level of individual particles. We describe the application of our analysis in practice to analyze simulations of single pulse and dual and triple colliding pulse accelerator designs, and to study the formation and evolution of particle beams, to compare substructures of a beam and to investigate transverse particle loss.

Index Terms

feature detection, feature-based analysis, visualization, plasma-based particle acceleration

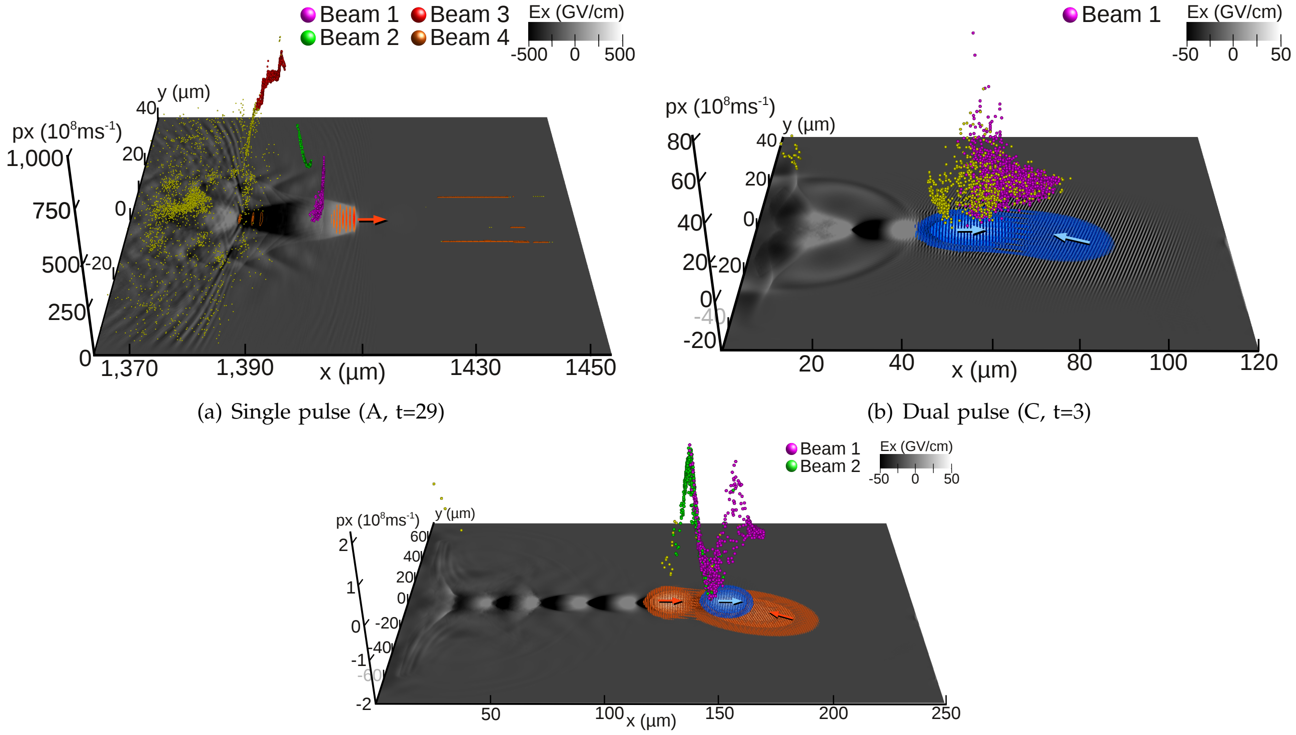
1 Introduction

Particle-in-cell (PIC) simulations are widely used in computational studies in particle physics and solid and fluid mechanics research, e.g., to study granular materials. In particle physics research, PIC simulations are a critical tool for the analysis of complex physical processes, such as magnetic reconnection, and for understanding, modelling and design of conventional and plasma-based particle accelerators and fusion reactors.

A central challenge in the analysis of complex particle simulation data arises from the fact that while hundreds of millions and in many cases billions to trillions [1] of particles are required for accurate simulation, only a small fraction of the particles form particle features of interest. Particle features, such as dense particle bunches in plasmas or the halo of a particle beam in a linac, evolve over time and are highly dynamic. However, due to the massive data size and the high cost for reconstructing full particle traces from the data, particle features are commonly identified instantaneously based on information from single timesteps. This loss of temporal context makes accurate feature detection difficult, requiring complex feature tracking methods in order to reconstruct essential temporal information.

To address this challenging problem, we propose a multi-stage analysis paradigm combining advanced query-driven analyses and machine learning. The first stage of the analysis reduces the amount of data required during subsequent analyses by using advanced temporal queries to identify a coarse subset of particles of potential interest. For this much reduced subset of particles, the second stage then reconstructs the full temporal traces. Using advanced index and query methods enables an efficient implementation of the temporal query and particle tracing [2]. In the third stage, we first process the particle traces to identify characteristic single particle events of interest, such as, changes in state or maxima in energy or transverse amplitude. Based on these per-particle events and the particle traces, we then identify the particle feature(s) of interest using advanced machine learning methods. By preserving the full temporal particle contexts, this analysis paradigm enables more accurate and reliable detection of dynamic particle features.

- *O. Rübel and E.W. Bethel are with the Visualization Group, Computational Research Division, Lawrence Berkeley National Laboratory (LBNL).*
- *C.G.R. Geddes and Min Chen are with the Lasers, Optical Accelerator Systems Integrated Studies (LOASIS) program of the Accelerator and Fusion Research Division, LBNL.*
- *E. Cormier-Michel is with Tech-X Corp..*



(a) Single pulse (A, t=29)

(b) Dual pulse (C, t=3)

(c) Triple pulse (D, t=9)

Fig. 1. Visualizations of three different simulation datasets modeling different accelerator designs showing in the (x, y)-plane: i) a gray-scale rendering of the electric field E_x in the acceleration direction x and ii) iso-contours of the electric field E_y (blue) and E_z (orange) illustrating the location of the laser pulses polarized in y and z, respectively. A (x, y, p_x) particle scatterplot is shown above the (x, y)-plane where color indicates different sets of particles: i) all candidate particles (yellow) and ii) multiple sets of particles found by the analysis to form different main particle beams over the course of a simulation (magenta, green, red and brown). The dataset and timestep used are indicated in brackets in the subfigure captions (see Table 1). Figure *b* and *c* show results at early timesteps during the collision of the laser pulses. Figure *a* shows a medium timestep, because the beam particles enter the simulation window later in time in single-pulse runs.

In this work, we focus on the application of this general analysis paradigm to design a novel algorithm for the automatic detection and extraction of particle beams and temporal beam substructures—here called acceleration features—in plasma-based particle accelerator simulation data. We introduce the application problem in detail in Sec. 2–4. In our analysis (Sec. 5–8), we first introduce a novel definition of acceleration features (Sec. 5.1) and describe an efficient algorithm for detection and extraction of acceleration features (Sec. 5.2). We also present feature grouping (Sec. 6.1) to facilitate analysis of high-level features, e.g., particle beams, and describe a novel interface for validation (Sec. 6.2) and query-based visual exploration of acceleration features (Sec. 6.3). We describe our implementation details in Sec. 7. We validate our methods using two- (2D) and three-dimensional (3D) datasets, which model a large variety of accelerator designs (Sec. 8.1). Next, we discuss the application of our methods to: i) study the formation and evolution of particle beams, ii) analyze transverse particle loss, and to iii) compare temporal substructures of a particle beam (Sec. 8.2). Finally, we study the runtime performance of our feature detection algorithm (Sec. 8.3).

2 Plasma-based Particle Acceleration

Particle accelerators are fundamentally important tools in modern science and play a key role in high-energy physics, medical therapy, and advanced imaging methods for material sciences, chemical sciences, and life sciences. The acceleration gradients that can be achieved using conventional particle accelerator designs are limited due to dielectric breakdown of the acceleration tube and radiation loss. This limitation has led to the development of increasingly large and expensive particle accelerators, such as the Large Hadron Collider at CERN. Plasma-based particle acceleration is a new technology that can produce and sustain acceleration fields thousands of times stronger than conventional accelerators, allowing particles to be accelerated to high energy levels within very short distances (centimeters to meters), potentially significantly reducing size and cost.

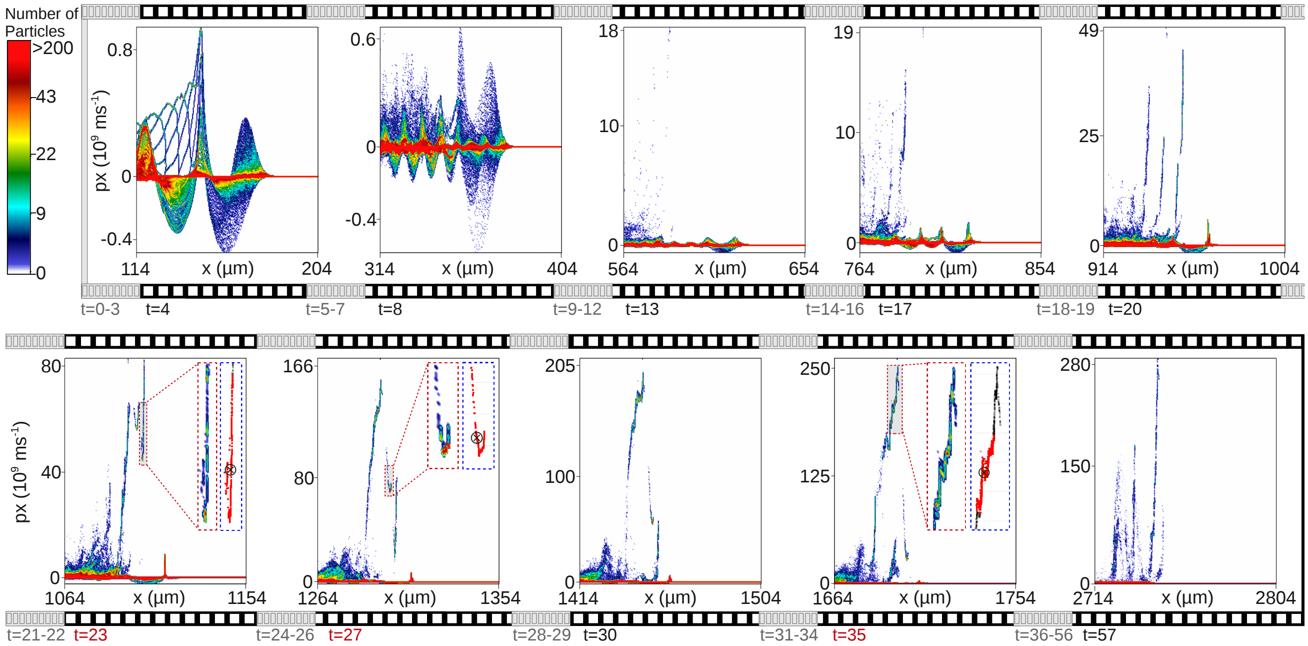


Fig. 2. Time series of phase space (x, p_x) density plots for dataset A. Through manual investigation of the time series, a physicist identified at timesteps $t = 23, 27, 35$ three main particle beams as peaks in phase space density that reach a local maximum in energy (red boxes). Our feature-based analysis identifies the same three principle beams as the three highest ranking groups and corresponding reference features (blue boxes).

Plasma-based accelerators use a short ($\leq 100fs$) ultrahigh intensity ($\geq 10^{18}W/cm^2$) laser pulse to drive waves in a plasma. Electrons in a hydrogen plasma are displaced by the radiation pressure of the laser pulse while the heavier ions remain stationary. This displacement of the electrons in combination with the space-charge restoring force of the ions, drives a wave (wake) in the plasma. Similar to a surfer riding a wave, electrons that become trapped in the plasma wave are accelerated by the wave to high energy levels.

Fig. 1(a) shows an example of such a single-pulse plasma-based accelerator simulation. The laser travels in the positive x direction (from left to right), exciting a plasma wave whose longitudinal (accelerating) electric field amplitude is displayed in a grayscale as a function of x (longitudinal) and y (transverse) space. Particles which exceed the wave's phase velocity v_ϕ, which for these conditions is roughly set by the laser group velocity in the plasma, can be trapped and accelerated over long distances and to high momentum as they co-propagate with the wave. In Fig. 1(a), the particles that become accelerated are plotted above the electric field, showing their distribution in the phase space of longitudinal momentum p_x versus x and y.

Achieving reproducible high-quality particle beams—i.e., particle beams with high mean energy, high charge, and low energy spread—is a challenging task and requires control of the complex injection (trapping of particles) and acceleration processes in plasma-based particle accelerators. Over the course of the last decade(s), physicists have developed a large range of methods for improving the reliability and quality of plasma-based accelerators [3]. Controlled manipulation of the initial plasma density [4] allowed LOASIS [5] researchers to achieve high-quality electron beams [6] and energies of up to 1 GeV using a centimeter long plasma [7].

Single-pulse plasma-based accelerator designs rely on self-trapping of electrons, i.e., the beam electrons are self-trapped/injected by plasma wave breaking rather than being injected directly or indirectly via a secondary injection mechanism. More recent experiments have employed multiple colliding laser pulses to more directly control particle injection [8]. As shown in Fig. 1(b), dual pulse experiments employ two colliding laser pulses, a driving pulse and a secondary pulse. Similar to single-pulse experiments, a single driver pulse drives the main plasma wave, here traveling in x direction. However, the driver pulse is typically of lower intensity than in single-pulse experiments, avoiding self-trapping of particles. A secondary laser pulse is then used which, when overlapped with the driver pulse, generates a beat wave which kicks particles within the overlap volume into the trapped orbit of the main plasma wave [8]. One limitation of the dual-pulse design, however, is that controlled injection of particles is only possible close to the driver pulse. Triple-pulse designs (see Fig. 1(c)) overcome this limitation by colliding the secondary injector pulse not with the main driver pulse but with a third pulse, located behind the driver pulse and traveling in the same direction as the driver. To avoid interaction between the injector and driver pulse, the driver pulse is polarized in the y direction and the other two pulses in the z direction.

To better understand the complex acceleration processes not accessible to analytic theory, LOASIS researchers model a variety of plasma-based accelerator experiments computationally using the VORPAL [9] simulation code. Explicit PIC simulations [10] using VORPAL, self-consistently model the interactions between the laser pulse(s), plasma and particle bunches. In the acceleration process, a large portion of the laser energy is transferred to the plasma, i.e., the laser and plasma evolve together. Controlling this process is crucial for the design of a reliable, high-quality plasma-based particle accelerator. Accurate modeling of plasma-based accelerators is computationally expensive and the large disparities in scale between wavelength of the laser and the plasma wave make simulation of the entire hydrogen plasma at once impractical. To save resources, simulations employ a moving window approach, in which only a region around the laser pulse is simulated at each timestep and the simulation window is moved (in x direction) along with the laser at the speed of light. The relative x location within the simulation window (x_{rel}) of the laser pulse and a relativistic particle beam are, hence, quite stable.

To evaluate the effectiveness of our methods against a representative set of physics cases, we use VORPAL simulation datasets (Table 1) of varying size and temporal resolution, modeling a large variety of plasma-based accelerator designs. A and B are comparable single-pulse simulations with different temporal resolution. C and F model similar dual-pulse accelerators in 2D and 3D, respectively. Datasets D and E model triple-pulse accelerators with varying plasma channels. The simulation stores for each particle the spatial location (x, y, z), momentums in x, y and z direction (p_x, p_y, p_z), identifier (id), and weight (wt). The momentums are described in the simulation by $\gamma * v$ (the momentum vm divided by the mass) and are in units of ms^{-1}.

		Per Timestep ≈		**Total**	
	Type	**Size** (MB)	**#Particles**	**#Steps**	**Size** (MB)
A	S (2D)	50	610,000	58	2,902
B	S (2D)	50	610,000	226	11328
C	D (2D)	165	2100000	53	8756
D	T (2D)	270	3,300,000	36	9,708
E	T (2D)	269	3,300,000	36	9,690
F	D (3D)	91,923	1,036,000,000	79	7,261,923

TABLE 1
Description of the simulation datasets used for evaluation. S = Single, D = Dual, T = Triple pulse simulation.

3 Analysis Problem

While large numbers of particles are required for accurate simulation of plasma-based particle accelerators, only a small fraction ($< 1\%$) of the particles are accelerated to high energy levels and subsequently form particle beams of interest. Informally, a particle beam is defined by a compact bunch of accelerated particles (i.e., particles with high p_x values) condensed in physical space (x, y, z) and momentum space (p_x, p_y, p_z). While being trapped in the plasma wave, the particles forming a beam are accelerated over a period of time until they eventually outrun the wave and decelerate again. During the acceleration process, different sets of particles of a beam reach their peak in energy at different points in time, defining distinct temporal beam substructures. A central challenge in the analysis of plasma-based particle accelerator simulation data is to accurately detect and extract all particle beams and their temporal substructures, here called acceleration features.

Traditionally, the detection and classification of particle beams is performed manually. As illustrated in Fig. 2, a physicist manually investigates animations of the complete time series. Based on this information, the physicist first identifies a reference timestep at which a beam is most condensed as it reaches its peak in energy. To extract the particles of interest from the data, the physicist then defines a set of thresholds in x and p_x. As Fig. 2 illustrates, the majority of the particles appear at low energy levels (indicated by p_x) and are not of interest. Furthermore, multiple distinct particle beams of interest may form during the course of a simulation. Manual analysis of all beam-like structures is challenging and time-consuming and identification of complex beam substructures due to differences in the temporal evolution of the particles is not possible.

4 Related Work

Visualization and statistical analysis are commonly used to facilitate knowledge discovery from large, complex accelerator simulation data [11]. A large number of analysis frameworks—such as, ROOT [12], IDL [13], VisIt [2], [14], and many others—are available for this purpose. Fonseca et al. [15] described the use of the OSIRIS [16] framework for particle tracing in plasma-based accelerator simulations. In this context, a researcher selects the particles of interest manually and then re-executes the simulation to reconstruct the particle traces of interest at a higher temporal resolution. Martins et al. [17] applied these methods to study ion dynamics

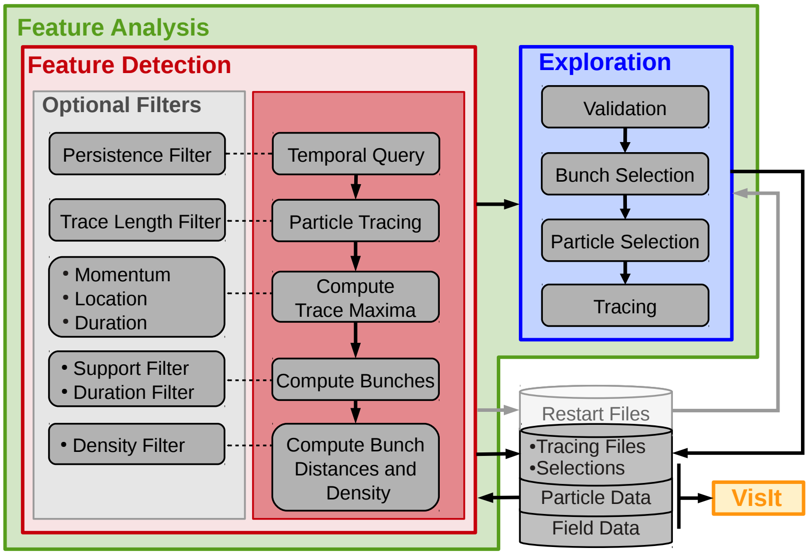


Fig. 3. Illustration of the analysis pipeline design.

and acceleration in relativistic shocks. In contrast to these tools and methods, we focus on the problem of automating the analysis of particle-based features in plasma-based accelerator simulation data.

Several efforts have explored automating different aspects of the analysis process. Bagherjeiran et al. [18] described graph-based orbit classification in plasma simulations. Love et al. [19] conducted an image space analysis of coherent structures in plasma simulations, and Hlína et al. [20] described an image-based study of dynamic patterns and their velocities in thermal plasma jets. None of these methods, however, address particle dynamics as a function of time nor inspect particle bunches and Hlína et al. and Love et al. focus on structures of the plasma itself rather than particles.

Ushizima et al. [21] described clustering-based beam detection aimed at automating the classification of single high-quality particle beams in 2D plasma-based accelerator simulations. Rübel et al. [22] introduced beam-path analysis for the classification of potentially multiple high-quality particle beams. Ushizima et al. and Rübel et al. both identify particle beams as regions of high particle density in phase space at discrete points in time. These per-timestep features are correlated across time to enable feature tracking and to improve the quality of the feature extraction. Rübel et al. [22] described the use of the average path of a feature over time—i.e., the beam path—to extract the temporal phases of a beam and to define particle-to-beam distances. We use the concept of beam paths to improve particle selection and to compute feature characteristics and statistics.

In this work, we generalize and further formalize the concept of acceleration features by making temporal coherency an integral part of the feature definition, which allows us to i) avoid the complex and error-prone feature tracking and ii) detect temporal beam substructures and other beam-related features that could not be detected using previous methods. Based on the results from the feature detection, we describe a new approach towards feature-based visual analysis [23] and exploration of acceleration features with a particular focus on exploration of possibly large numbers of features [24] using feature-grouping and feature queries.

5 Feature Detection

The goal of this work is to formalize the definition of acceleration features (Sec. 5.1), to develop an algorithm that enables automated detection and extraction of these features (Sec. 5.2), and to enable physicists to effectively explore the space of acceleration features using dedicated visualization and interaction methods (Sec. 6). Fig. 3 illustrates the structure of our analysis pipeline. In this section we focus on the feature detection step shown in the red box of Fig. 3. In this step, we process the particle data and extract all acceleration features of interest. Later, we discuss the feature exploration step (Fig. 3, blue box) in Sec. 6.

5.1 Feature Definition

Similar to the manual beam selection illustrated in Fig 2, previous automatic beam detection methods identify particle beams as regions of high particle density in (x, p_x) or (x, y, p_x) phase space [21], [22]. Density-based criteria, however, are inherently static in that they do not capture properties of temporal coherency. Hence,

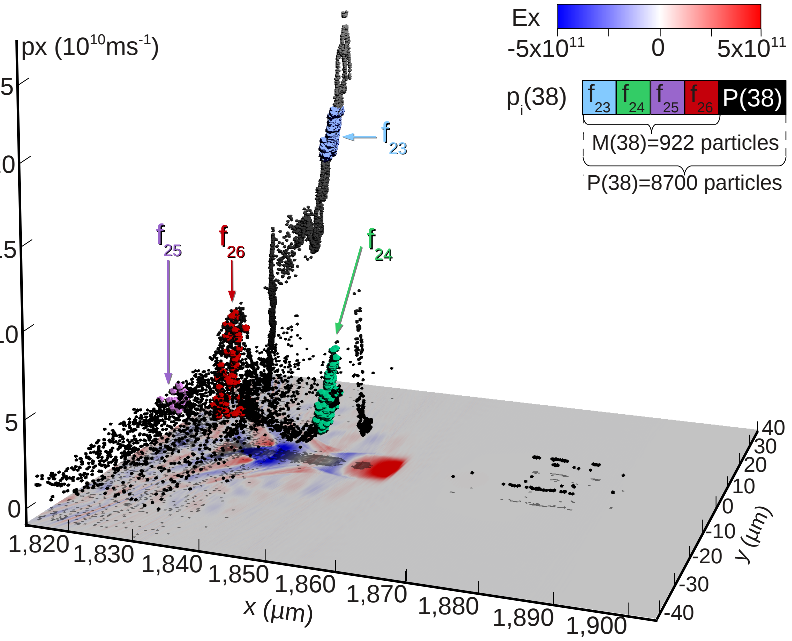


Fig. 4. A (x, y, p_x) particle plot of $t = 38$ of dataset A showing i) all particles used in the analysis $P(38)$ (black), ii) four acceleration features f_{23} to f_{26} detected at $t = 38$ (colored particles) with $M(38) = f_{23} \cup f_{24} \cup f_{25} \cup f_{26}$, and iii) the electric field in x direction E_x (bottom plane). By focusing on $M(38)$, rather than $P(38)$, we can easily separate the different features, even in this very complex scenario, with multiple features appearing at the same time and at overlapping spatial locations (green/blue particles).

density-based beam detection algorithms are unable to distinguish important beam substructures due to temporal variations in the acceleration process. Informally, we extend the density-based beam criterion as follows:

Definition (V1) *An acceleration feature (particle bunch) defines a group of particles that are: i) condensed in space and energy and ii) reach a peak in energy (indicated by p_x) within the same timeframe.*

Formally, let $p_{i,x}(t)$ be the momentum in the acceleration direction x at timestep t of the particle with index i. Furthermore, let $M(t)$ be the set of particles with a local maximum in energy at timestep t;

$$M(t) = \{i \mid p_{i,x}(t) > p_{i,x}(t + \Delta t)) \; \forall \; \Delta t \in [-1, 1, ..., a]\}, \tag{1}$$

with $a \geq 1$. Choosing $a > 1$ allows filtering of temporarily local maxima that do not persist for more than a timesteps before the particles continue to accelerate. An acceleration feature f_k is then defined as:

$$f_k = \{i \mid p_i(t) \in c_k(h_{x,y,p_x}(M(t))\}. \tag{2}$$

$h_{x,y,p_x}(M(t))$ defines the 3D histogram in (x, y, p_x) space of all particles contained in $M(t)$. Each voxel of $h_{x,y,p_x}(M(t))$, hence, describes how many particles of the set $M(t)$ are located within the corresponding region of (x, y, p_x) space. $c_k(h_{x,y,p_x}(M(t))$ then defines the k'th connected non-zero region of $h_{x,y,p_x}(M(t))$. Each region c_k defines a maximal set of non-zero voxels of $h_{x,y,p_x}(M(t))$ that are connected in space and surrounded by empty voxels.

We define the support S of a feature f_k as the number of particles contained in f_k (i.e., $|f_k|$). Note, in practice, a particle may undergo several phases of acceleration, i.e., for any given pair of timesteps (t_i, t_j), $M(t_i) \cap M(t_j)$ may not be empty. For example, a particle may become trapped in a different period of the plasma wave after it was accelerated by an earlier wave-period and may participate in several distinct acceleration features at different points in time.

5.2 Algorithm

Our feature definition leads to a direct algorithm for feature extraction outlined in Fig. 3 (red box).

Temporal Query: In practice, we are only interested in high-energy features several times above the plasma wave's phase velocity v_ϕ, defined here by requiring that a particle's p_x momentum must exceed $10^{10} ms^{-1}$. Focusing on the much smaller subset of accelerated particles improves the memory and I/O footprint as well as runtime performance of the analysis. To extract the particles that achieve high energies, the algorithm evaluates the temporal query:

$$P = \{i \mid \exists p_{i,x}(t)) > 10^{10} ms^{-1},\ t \in [0,n]\}. \tag{3}$$

Particle tracing: Next, the algorithm traces the greatly reduced subset of particles returned by the temporal query P over time. This step requires the evaluation of an id-based equality query at each timestep to compute:

$$P(t) = \{p_i(t) \mid i \in P\}. \tag{4}$$

with $t \in [0,n]$. The sets $P(t)$ contain all information required for computing the particle traces TR with

$$TR = \{tr_i \mid i \in P\},\ \text{with } tr_i = \bigcup_{t\in[0,n]} p_i(t) \tag{5}$$

Compute Trace Maxima: Based on the particle traces TR, the algorithm then computes for each timestep t the set of particles with a local maximum in energy at t, i.e., $M(t)$, $t \in [1,n]$ (Eq. 1). As illustrated in Fig. 4 (colored particles), the particles contained in $M(t)$ form distinct clusters in (x, y, p_x) phase space.

Compute Bunches: The goal of the next step then is to classify the clear particle clusters defined by $M(t)$, each of which corresponds to a unique acceleration features f_k. In principle, any density-based clustering algorithm could be used to perform this task. Following our feature definition, we first compute for each set $M(t) \neq \emptyset$ the corresponding 3D histogram $h_{x,y,p_x}(M(t))$ in (x, y, p_x) space. Due to the transverse symmetry of the plasma wave and the roughly equal width of particle bunches in y and z, it is in practice sufficient to focus on (x, y, p_x) space for the bunch computation even for 3D input data. We use a 3D region growing algorithm to label the connected non-zero regions of the 3D histograms $h_{x,y,p_x}(M(t))$, each of which corresponds to a particle cluster in $M(t)$ defining an acceleration feature f_k.

Previous, density-based algorithms typically used the full set of particles of the temporal query $P(t)$ (Eq. 4)—or alternatively the subset of particles that suffice the condition $p_x > 10^{10} ms^{-1}$ at the current timestep—to separate particle bunches based on the (x, y, p_x) particle density. As illustrated in Fig. 4, unlike the clear clustering structure we observe for the particles in $M(t)$ (colored particles), the particles in $P(t)$ (black particles) often form continuous patterns, making density-based feature detection a challenging task. Using $M(t)$ allows for a more reliable and accurate detection of acceleration features. Furthermore, because different parts of a particle bunch often reach their peak energy (and p_x) at different points in time, using $M(t)$ allows us for the first time to automatically identify these distinct temporal features of a particle bunch.

Compute Bunch Distances and Density: To gather additional information about the multiple detected acceleration features f_k and to facilitate selection of particles in their vicinity, we use a modified version of the path-distance approach introduced by Rübel et al. [22]. Using the particles of the feature f_k as reference, we compute a single center trace describing the average location and momentum of the reference particles at all timesteps defined via:

$$r(f_k) = \{rp(f_k, t) | t \in [0,n]\} \tag{6}$$

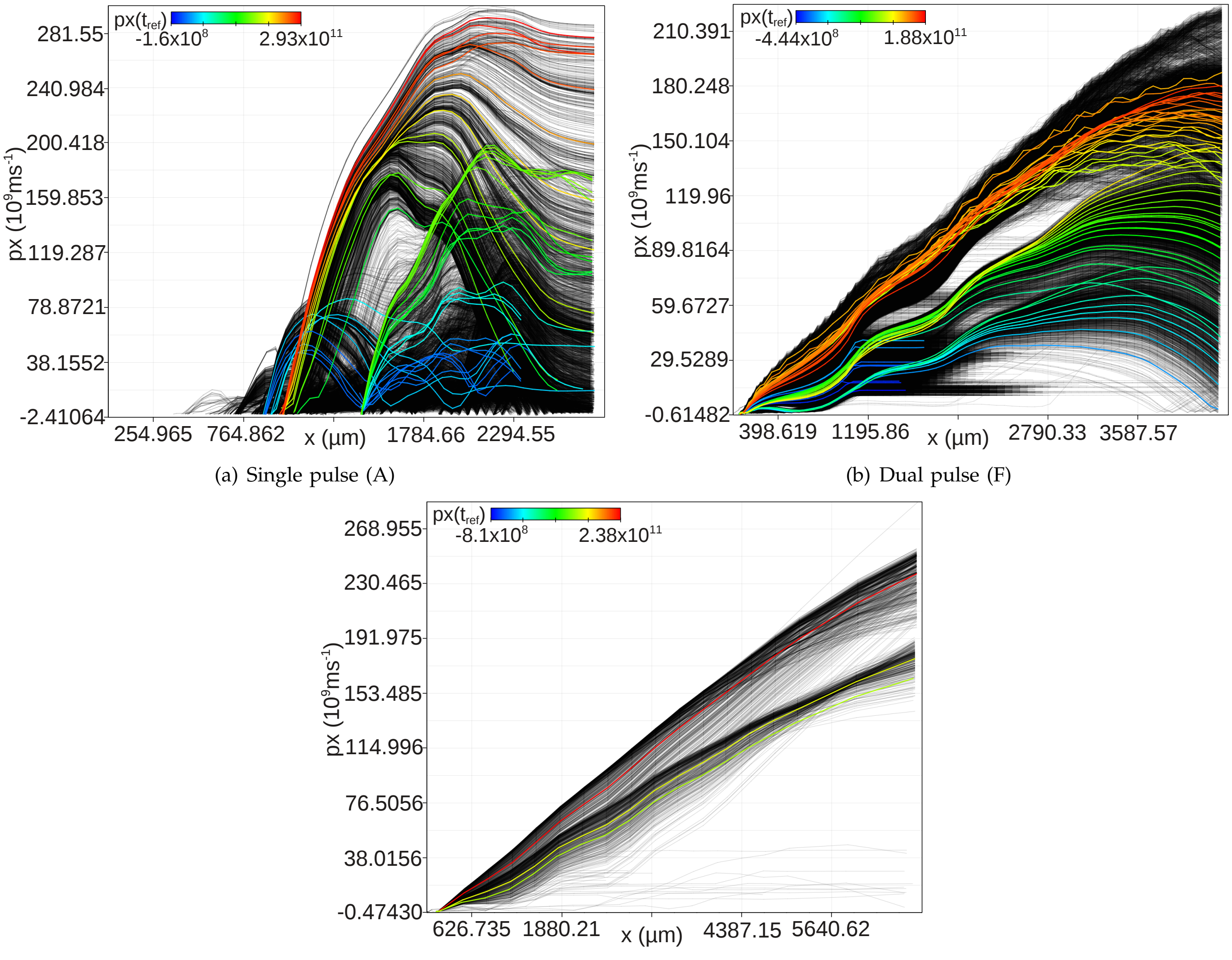
with $rp(f_k, t) = \frac{\sum_{p_i \in f_k} p_i(t)}{S}$ and $S = |f_k|$ being the number of particles contained in f_k. Fig. 5 shows as an example all particle traces TR (Eq. 5) and the center traces $r(f_k)$ of the main features found for dataset A, F and D.

Using the center traces, we next identify the different temporal phases of the features: i) pre-formation, ii) formation, iii) acceleration, iv) deceleration, and v) post-deceleration. The acceleration phase is the timeframe during which a feature is most coherent and is of most interest to our automated feature detection. The other temporal phases, and in particular the formation and deceleration phase, are of interest during later analysis to, e.g., compute the timepoint of minimum energy spread, which may occur later during deceleration. We define the start timepoint a_s of the acceleration phase as the first timestep t at which the center trace exceeds the base acceleration threshold of $p_x > 10^{10} ms^{-1}$. The end timepoint a_e of the acceleration phase is given by the reference timepoint t_{ref} at which the feature f_k was detected. The duration of the acceleration phase is defined by $A = a_e - a_s + 1$. For a more detailed discussion of the other temporal phases see [22],p22.

Based on the center traces $r(f_k)$ and all particle traces TR, we then compute the average distance over time in momentum space d_m and physical space d_s between each traced particle p_i and each acceleration feature f_k:

$$d_s(p_i, f_k) = \frac{\sum_{t\in[a_s,a_e]} ||s(p_i(t)) - s(rp(f_k, t))||_2}{A}\ m, \tag{7}$$

$$d_m(p_i, f_k) = \frac{\sum_{t\in[a_s,a_e]} ||p(p_i) - p(rp(f_k, t))||_2}{A}\ ms^{-1}. \tag{8}$$



(a) Single pulse (A)

(b) Dual pulse (F)

(c) Triple pulse (D)

Fig. 5. (x, p_x) particle trace plot showing: i) the traces of all particles identified by the temporal query (black) and ii) the center traces $r(f_k)$ of the main features colored by their reference p_x level using the indicated color scale. (a) 11275 traces and 39 features (b) 37,486 traces and 61 features. (c) 1122 traces and 3 features.

$s(p_i(t))$ and $p(p_i(t))$ refer to the spatial $\{x,y,z\}$ and momentum $\{p_x,p_y,p_z\}$ coordinates of particle $p_i(t)$, respectively. Using two independent distance functions avoids non-intuitive normalization and yields distances at physically meaningful scales. This is crucial to ensure comparability of distance values across features and datasets, allowing physicists to intuitively define meaningful thresholds for d_s and d_m

Feature characteristics: To facilitate the exploration and analysis of accelerations features, we compute a set of derived feature statistics. In the following we provide a brief overview of the main feature characteristics used.

The feature detection itself already provides us with a rich set of information about the individual features f_k, like: i) the support S defined by the number of particles contained in f_k, ii) the reference timestep t_{ref} at which a feature reaches its peak energy, iii) the peak p_x level at t_{ref}, iv) the reference location at t_{ref}, as well as v) the start (a_s), end (a_e), and duration (A) of the acceleration phase.

Via the distance fields d_s and d_m, we then define the set of particles in close proximity to the feature over time:

$$\begin{aligned} D(f_k) = \quad & \{p_i |\ d_s(p_i, f_k) < 2 * 10^{-6} m \ \&\& \\ & d_m(p_i, f_k) < 10^{10} ms^{-1}\}. \end{aligned} \tag{9}$$

The thresholds $d_s < 2 * 10^{-6} m$ and $d_m < 10^{10} ms^{-1}$ effectively define the maximum allowed spatial and momentum spread of the vicinity set, respectively. Based on prior physical knowledge and data analysis results we have chosen these thresholds conservatively, to ensure that no particles from other main features bias the analysis, while still providing a good intuition of the larger particle bunch. Example vicinity sets for dataset F are shown later in Fig. 8. Similar to $D(f_k)$, we also define the per-timestep vicinity sets $D(rp(f_k, t))$ based on the per-timestep distances of the particles to a feature.

Based on the vicinity sets, we derive additional information about the features. The feature density $R = |D(f_k)|$ defines the amount of particles in the vicinity of a feature over time, providing important information about how condensed a feature is in both physical and momentum space. In the analysis, we use R to rank features, enabling fast identification of the most condensed features. To estimate the spatial and momentum (energy) spread of the particle feature at peak energy, both of which are key beam parameters, we compute the standard deviation in physical space CS and momentum space CM of all particles in the vicinity of the feature at the reference timepoint t_{ref}.

Parameters: The main parameter affecting the feature detection itself refers to the resolution of the histograms $h_{x,y,p_x}(M(t))$ used in the bunch computation. Here we can take advantage of the fact that the subset of particles with a local maximum in energy at a given timestep t (Eq. 1, Fig. 4) shows a clear clustering structure, which allows us to use a coarse binning in the region growing to avoid possible over-segmentation. For all experiments discussed in this study, we use 60 bins for the dimensions (x, y, p_x), which has shown to be in practice sufficient to reliably distinguish all features f_k. In our experiments, region-growing-based segmentation of the histograms $h_{x,y,p_x}(M(t))$ has been reliable, but other density-based segmentation approaches, such as mean-shift clustering, may be practical alternatives.

The ability of the algorithm to detect temporal subfeatures of particle beams inherently depends on how time is discretized by the simulation output. Smaller step sizes between timesteps enable the algorithm to detect finer temporal subfeatures. As we show later, the algorithm performs well even when just a small number of timesteps are available (see Sec. 8). While the algorithm may retrieve a larger number of features for data with high temporal resolution, the number of main particle beams is independent of the temporal resolution of the data. As described later in Sec. 6.3, automatic grouping of acceleration features allows us to identify features that are part of the same main particle beam, so that the user can focus on a well-defined set of high-level features of interest. Furthermore, temporal subsetting can be used in the case of data with high temporal resolution to avoid detection of small features that are due to high-frequency variations in time, such as the oscillating motion of particles in the plasma wave.

Filtering: To avoid detection of small, spurious features, the feature detection provides various optional filters associated with the different analysis steps. In practice, the different filters may be disabled without major impact on the analysis, i.e., disabling filters mainly results in the retrieval of additional features. To avoid removal of possibly relevant features, the filters are, therefore, typically kept at conservative default levels indicated below, and are not modified between different analysis runs. In practice, filtering is performed later during data exploration (Sec. 6.3).

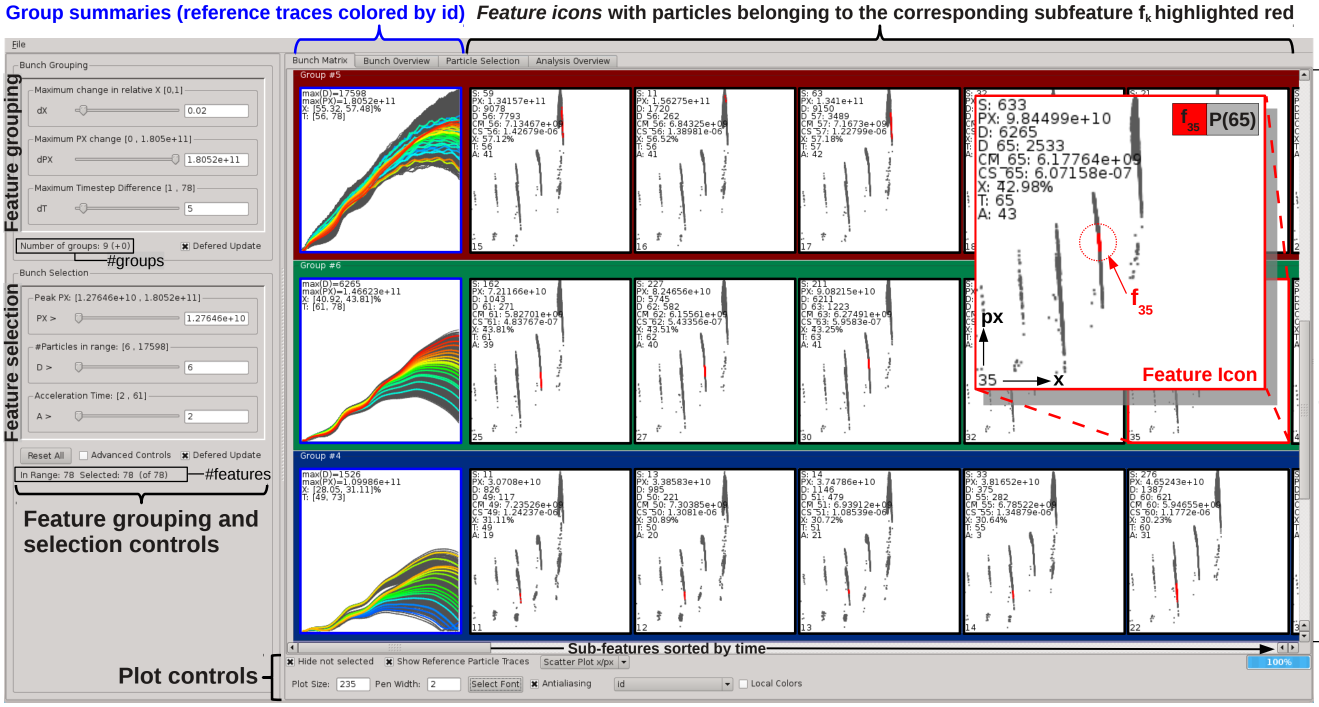
The first set of filters are associated with the temporal query and particle tracing and apply to particles. First, the temporal query computes a count for each particle, indicating how often a particle exceeds the threshold of $p_x > 10^{10} ms^{-1}$. The persistence filter then removes particles from the temporal query result P (Eq. 3) that do not persist for at least s_p timesteps at high energies (default value $s_p = 5$). Similarly, the trace-length filter—applied during particle tracing—removes particles that quickly exit the simulation. This filter is similar to the persistence filter but is independent of a particle's p_x value (disabled by default).

The second main set of filters apply to the sets of trace maxima $M(t)$ (Eq. 1). While the temporal query captures only high-energy particles, along their traces over time, the particles may exhibit local, low-level maxima in energy. We first consider only maxima with $p_x > 10^{10} ms^{-1}$. The second maxima filter refers to the duration parameter a (see Eq. 1) and is used to remove spurious maxima that do not persist in time. The default value $a = 3$ is chosen conservatively to ensure that all features of possible interest are preserved even in the case of datasets with low temporal resolution. The third maxima filter refers to the transverse particle location. Within the simulation, the laser pulse—and the particle beams of interest—are centered transversely at $y = 0$ and $z = 0$. Particle maxima in energy at transverse locations far from the center are, therefore, typically not of interest (default filter value $-3 * 10^{-6} m < y < 3 * 10^{-6} m$).

The third set of filters are associated with the computation of the bunches and refer to removal of features f_k. During the computation of the bunches f_k and associated vicinity sets $D(f_k)$, we can optionally remove features f_k with: i) low support $S = |f_k|$, ii) low density $R = |D(f_k)|$ defined by the number of particles in the vicinity of the feature, and iii) short duration of acceleration A. In our default setup, we disable the density filter and use very conservative values of $S > 10$ and $A \geq 2$ for the support and duration filter.

6 Data Analysis and Exploration

Based on the information from the feature detection we define an efficient top-down workflow towards feature-based exploration of plasma-based particle acceleration data. Rather than exploring the data bottom-up—starting from the level of the entire particle data to develop a view of the particle beams of interest—we



Bunch Grouping

Maximum change in relative X [0,1]: dX 0.02

Maximum PX change [0 , 1.805e+11]: dPX 1.8052e+11

Maximum Timestep Difference [1 , 78]: dT 5

Number of groups: 9 (+0) ☒ Defered Update

Bunch Selection

Peak PX: [1.27646e+10 , 1.8052e+11]: PX > 1.27646e+10

#Particles in range: [6 , 17598]: D > 6

Acceleration Time: [2 , 61]: A > 2

☐ Advanced Controls ☒ Defered Update

In Range: 78 Selected: 78 (of 78)

☒ Hide not selected ☒ Show Reference Particle Traces Scatter Plot x/px

Plot Size: 235 Pen Width: 2 ☒ Antialiasing id ☐ Local Colors

Fig. 6. Path Analysis GUI showing the feature selection interface (dataset F). Left: Controls for grouping and selection of particle bunches. Bottom: Controls to define plot settings. Right: Matrix view with features grouped based on their x location. Each row (group) represents a distinct particle beam and its various substructures. Frame color of *feature icons* is used to indicate group summary plots (blue, first column), subfeature plots (black/red), and reference subfeature plots (red).

enable physicists to explore the data top-down, starting from a high-level, feature-based view down to the level of individual particles. As part of this process we identify higher level acceleration features (e.g., particle beams) via feature grouping (Sec. 6.1). The visual feature exploration then proceeds as follows. The first step typically involves validation of the automatically detected features (Sec. 6.2). Afterwards, the user explores the set of acceleration features in more detail and performs grouping and sub-selection of features to identify the main features of interest (Sec. 6.3). Both, feature grouping and selection are often performed simultaneously in an iterative fashion. Once the user has identified a particular feature of interest, the next step involves extraction of the particles associated with the feature for further analysis (Sec. 6.4).

To assist the user in the feature exploration process, we have developed a dedicated feature browser, discussed in detail in the following subsections. Fig. 6 shows a screen-shot of the main GUI. The tool supports different sets of views aimed at facilitating exploration and validation of the acceleration features extracted by the feature detection. The different views used for validation, feature selection and particle selection are arranged in tabs. Controls for the grouping of features and the selection of features and particles are shown on the left. The selection and grouping controls consist of sliders and associated numeric fields.

6.1 Feature Grouping

Within the same particle beam, different sets of particles often reach their peak energy at different points in time. These characteristic substructures of a beam are detected as separate features f_k by the feature detection. For further analysis, and to ease data exploration, it is often useful to group features f_k to form macro-features G_g, each of which represents a higher level feature (e.g., particle beam).

A particle beam is defined by an evolving set of particles trapped in a single period of the plasma wave. The subfeatures of a particle beam, hence, appear as consecutive acceleration features in time located at similar relative x positions (x_{rel}) within the moving simulation window. We, therefore, group temporarily consecutive features f_k by their relative x location (x_{rel}) using Algorithm 1.

Algorithm 1 depends on the parameters, dX, dPX and dT. A good value for the maximum spatial distance, dX, depends on the wavelength of the plasma wave (and hence the laser frequency), but is typically on the order of 2% of the size of the simulation window. The maximum temporal lack between temporally consecutive features, dT, depends on the temporal resolution of the data but is typically set to span a large fraction of the

Algorithm 1 Group features
User-defined input parameters:
dX: maximum distance in relative x location x_{rel},
dPX : maximum difference in energy indicated by p_x,
dT : maximum temporal lack
Input data: Features f_k, $k \in [0, m]$, sorted in time
Output: Set of groups G_g, $g \in [0, ng]$

```
$G_0 \leftarrow f_0$        // Initialize the group $G_0$
$ng \leftarrow 1$           // Initialize the number of groups $ng$
// Iterate over the features $f_k$ to define the group assignment
for $k \in [1, m]$ do
    $cg \leftarrow -1$            // Initialize the current group (undefined)
    $cdxrel \leftarrow dX$      // Initialize the current distance in $x_{rel}$
    // Iterate over all groups to find the group that is closest in
    // $x_{rel}$ and that suffices the $dT$, $dPX$, and $dX$ thresholds
    for $g \in [0, ng]$ do
        if $(t_{ref}(f_k) - t_{ref}(G_g.last)) \leq dT$ then
            if $(p_{x,ref}(f_k) - p_{x,ref}(G_g.last)) \leq dPX$ then
                $dxrel \leftarrow (x_{rel,ref}(f_k) - x_{rel,ref}(G_g.last))$
                if $dxrel \leq cdxrel$ then
                    $cdxrel \leftarrow dxrel$
                    $cg \leftarrow g$
    // If a group assignment was found for the feature $f_k$
    if $cg \neq -1$ then
        $G_{cg} \leftarrow G_{cg} \cup f_k$      // Add $f_k$ to $G_{cg}$
    else
        $ng \leftarrow ng + 1$          // Increase the number of groups
        $G_{ng} \leftarrow f_k$            // Create a new group for $f_k$
```

total number of timesteps (≈ 10–20%). Restricting the maximum difference in energy, dPX, is needed only in rare cases where two features co-exist at the same location at different energy levels. For datasets A, e.g., a new feature forms at a high energy level while a previous beam reaccelerates at a lower energy level (Fig. 4). Fig. 6 shows as an example three main groups of dataset F, each of which corresponds to a main particle beam trapped in the first, second and third period of the plasma wave.

6.2 Feature Overview and Validation

To provide an overview of the analysis, we use a particle trace plot of (x, p_x) phase space (Fig. 5). This *bunch overview* plot shows the center-traces $r(f_k)$ of all features and the traces of all particles used. The color of the center traces is used to visualize general feature-properties (e.g., feature id or density R) or time-dependent statistics, such as the number of particles in the vicinity of the feature ($|D(rp(f_k, t)|$), the estimated spatial and momentum spread (CS, CM), or the mean momentum and spatial coordinates ($rp(f_k, t)$).

Detailed validation of the multiple detected acceleration features f_k is then performed via a set of temporal statistic plots and particle plots of phase space (x, p_x) and physical space (x, y) (see Fig. 7). The temporal statistics bar charts (Fig. 7, right) show for each timestep: i) the number of particles with energies above the plasma wave's phase velocity defined by the query $p_x(t) > 10^{10} ms^{-1}$, ii) the maximum p_x level (not shown), and iii) the number of features found. These plots provide simple means for identifying various events of interest, such as different phases of injection, acceleration or particle loss (Fig. 7, top right). The scatter-plots then show all particles used in the analysis and the features f_k found at the current timestep highlighted in color (Fig. 7, left). Instead of the features f_k, a user may also chose to highlight a user-defined subset of particles in the scatter-plots. The current timestep can be set manually or the plots can be animated to play through the time series.

6.3 Feature Selection

The initial validation step provides an overview of the general acceleration behavior of the data and the features extracted by the analysis. The *bunch matrix* view provides a detailed summary of all features and

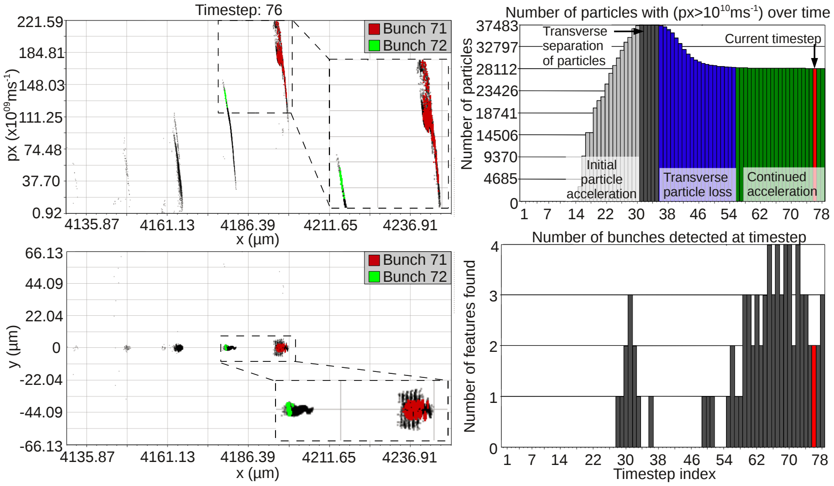


Fig. 7. Validation plots (dataset F) showing: i) particle plots in (x, p_x) and (x, y) space (left) showing the feature(s) f_k found at the current timestep and ii) temporal statistic plots showing the number of particles with $p_x > 10^{10}ms^{-1}$ (top right) and the number of bunches (bottom right). The red bars in the temporal statistics plots indicate the current timestep.

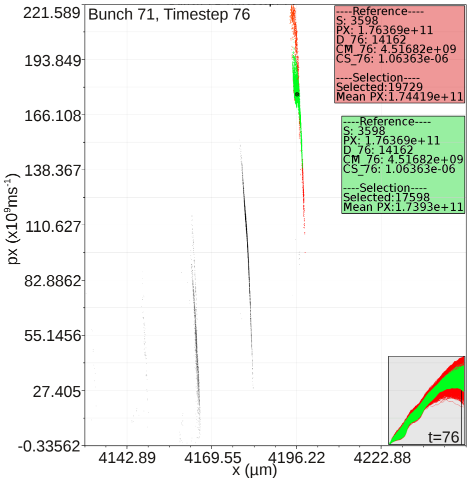


Fig. 8. Particle selection plot showing the default vicinity query $(d_s < 2 * 10^{-6}m \quad AND \quad d_m < 10^{10}ms^{-1})$ ($D(f_{71})$, green) and the extended beam query $(d_s < 4 * 10^{-6}m \quad AND \quad d_m < 4 * 10^{10}ms^{-1})$ (red).

facilitates the identification of the main features of interest through grouping of features and feature queries. In this view, each feature is represented by a single *feature icon* arranged in a matrix-layout with icons sorted in time. Each *feature icon* (see inset of Fig. 6) shows a (x, p_x) particle trace plot or an (x, p_x) or (x, y) scatter-plot of all relevant particles ($P(t)$, Eq. 4)(gray) with the corresponding feature f_k highlighted in red. In the particle trace plot, the (x, p_x) traces of the reference particles f_k are shown in gray and the center trace $r(f_k)$ is shown in color using the same coloring schemes as in the *bunch overview* view discussed in the previous subsection. The reference timestep t_{ref} at which the feature f_k reaches its peak energy is indicated via a vertical line at the corresponding reference x location (not shown). The (x, p_x) and (x, y) scatter-plot shows all particles $P(t_{ref})$ at the reference timestep t_{ref} with the reference particles of f_k highlighted in red. In addition to the plot, further feature characteristics—such as feature density R and estimated energy spread CM etc.—are summarized in the legend of the plot (see Sec. 5: *Feature characteristics*, for details).

Fig. 6 shows an example of the *bunch matrix* view with the grouping of features enabled. In the *bunch matrix* view, groups of features are represented as horizontal boxes. This layout allows for simple exploration of high-level features (groups) via vertical scrolling and investigation of subfeatures within a group via horizontal scrolling. Each horizontal box—each representing a group—then contains a set of *feature icons*. The leftmost *feature icon* shows a summary of the group. The remaining plots show the individual features sorted by time. The feature with the highest density R is suggested as a reference for the group and is highlighted via a red frame. To indicate the relative importance of the group, the boxes are colored according to the density value (R) of the reference feature (red = high, green = medium, blue = low). The effectiveness of this ranking of groups is also visible in Fig. 6, with the main beam highlighted in red and the beams in the second and third wake-period shown in green and blue respectively.

In particular in the context of simulations that exhibit very complex acceleration behaviors, the feature detection may return a larger number of features, many of which are often of low quality. For many analysis tasks, only the best particle beams are of interest. Feature queries based on feature characteristics facilitate the exploration process and enable fast detection of the main features of interest. In practice, physicists commonly perform feature selection based on minimum thresholds in: i) the reference p_x level $PX = px(rp(f_k, t_{ref}))$, ii) the reference density $R = |D(f_k)|$, and iii) duration of the acceleration timeframe A, of the features. In the case that selection based on these feature characteristics is not sufficient, optional advanced user controls are made available, enabling the refinement of feature queries based on: iv) the minimum support $S = |f_k|$, v) maximum momentum spread at the reference timestep CM, vi) maximum spatial spread at the reference timestep CS, vii) minimum and maximum relative x location at the reference timestep $(x_{rel}(rp(f_k, t_{ref})))$, and viii) the minimum and maximum reference timepoint t_{ref} of the features. Feature selection can be performed in both the *bunch overview* and *bunch matrix* view. In the *bunch overview*, center traces of deselected features are shown in gray or hidden. Similarly, *feature icons* of deselected features are disabled (grayed-out) or hidden in the *bunch matrix* view.

6.4 Particle Selection

Once a user has identified a feature of interest, the next step is to extract the associated particles. In the simplest case, the relevant particles are defined by the feature f_k or group $\hat{G_k} = \{p_i \mid p_i \in f_k,\ k \in G_k\}$. In practice, the set of particles relevant for subsequent analysis is often highly dependent on the specific use-case. In these cases, the physicist may manually select the particles of interest at the automatically identified reference timestep (t_{ref}) of the feature. The temporal distance fields in physical space $d_s(p_i, f_k)$ (Eq. 7) and momentum space $d_m(p_i, f_k)$ (Eq. 8) allow complex particle selections to be defined simply via two maximum thresholds. Because d_s and d_m aggregate information from multiple timesteps, particles that are only temporarily close to the feature f_k at t_{ref} are easily excluded from the selection. In the case that selection based on d_m and d_s is not sufficient, optional advanced user controls are made available to further refine a selection via range queries based on the original particle coordinates x y, z, p_x, p_y, p_z or derived quantities, such as the per-timestep distance fields $d_s(p_i, rp(f_k, t_{ref}))$ and $d_m(p_i, rp(f_k, t_{ref}))$.

Fig. 8 shows an example plot used to validate particle selections. The plot is updated continuously during the selection process, enabling interactive modification and validation of particle selections. The plot consists of a 2D scatter-plot of all particles $P(t)$ with the selected particles shown in color. The user may map any available original and derived particle quantity to the coordinate axes and particle color. A cross-hair icon shows the reference location $rp(f_k, t_{ref}))$ of the feature. In addition, the (x, p_x) traces of the selected particles are displayed as an inset plot (Fig. 8, bottom right) and statistics of the feature f_k and the current particle selection are shown in the plot's legend (Fig. 8, top right). As described earlier, particle selections can also be highlighted and traced over time in the validation plots (Fig. 7).

7 Implementation

The feature detection (Sec. 5) and exploration (Sec. 6) are implemented in C++ using Qt and OpenGL for rendering of plots and the GUI. The design of our system allows us to built the feature detection and data exploration as a single, fully integrated analysis framework (Fig. 3, green box) or as separate tools (Fig. 3, red and blue box). Being able to separate the feature detection and data exploration allows us to execute the feature detection once as an offline pre-processing step directly at the supercomputing center—here at the National Energy Research Scientific Computing Center (NERSC)—where the original data is generated and stored. The feature detection generates in this case an analysis restart file which stores all main data structures of the analysis in a portable ASCII format generated using Boost [25]. The feature exploration can then be performed interactively using a local computer based on the analysis restart files. Restart files are generally small, and require even for the 3D dataset F ($\approx 7.2TB$) only $515MB$. Using restart files avoids unnecessary data movement and re-execution of the feature detection, improves reproducibility of the analysis and facilitates sharing of analysis results.

To accelerate the queries required for the temporal query (P, Eq. 3) and particle tracing (TR, Eq. 5), the feature detection takes advantage of bitmap indexing using the index/query software FastBit [26], [27]. Using FastBit [2], [28], significantly improves the performance of our feature detection algorithm (see Sec. 8.3). To further improve performance, the temporal query and particle tracing are performed backwards in time, avoiding extra calculations at early timesteps that do not contain any relevant particles.

To enable our collaborators to easily define advanced, custom analyses and visualizations based on the results from our analysis, we support the export of analysis results in a variety of different formats. From the feature analysis the user can export CSV and VTK files of user-defined and automatically generated subsets of particle data and particle traces, including raw and derived data. Id-based particle selections (e.g., f_k or $D(f_k)$) can also be saved as so-called named selections, defining a list of selected particles. This strategy provides us with great flexibility and enables our scientific collaborators to easily process particle features that have been identified by our analysis using their preferred external analysis tools, e.g, Matlab, R, or VisIt. We here us the state-of-the-art visualization system VisIt [14], [29] to generate advanced 3D visualizations that combine original simulation field and particle data and information derived from our feature analysis (see, e.g., Fig. 1, 4, 9, 10, and 11). We use VisIt because it is commonly used by our collaborators at the LOASIS program [2], [22].

8 Results

In the following we discuss results and applications of our feature-based analysis to demonstrate: i) that our system fulfills critical analysis requirements of our target users (Sec. 8.1), ii) that our system enables application scientists to address important scientific questions that cannot be addressed using current methods (Sec. 8.2), and iii) to study the compute performance of our system (Sec. 8.3). Note, the feature detection has been performed in all cases using the default parameter settings described in Sec. 5.2. All analysis results have been validated by three accelerator physicists.

8.1 Evaluation

The goal of this section is to evaluate whether our analysis meets critical user analysis requirements and needs.

Analysis Requirements: In close collaboration with accelerator physicists of LBNL's LOASIS program we identified the following set of main analysis requirements. The analysis should: **(R1)** enable automatic extraction of the main particle beam, **(R2)** correctly identify all main particle features of interest, **(R3)** be applicable to a diverse set of simulation datasets of varying spatial and temporal resolution and modeling different accelerator designs, and **(R4)** enable intuitive analysis of the overall acceleration process. Here we discuss how our analysis addresses these requirements.

(R1) Extracting the main beam: Many practical use cases depend on accurate characterization of the best particle beam, i.e.: i) identification of the most condensed beam and ii) extraction of the particles that form the beam. Automating the characterization of the main beam is central to enable the analysis of large simulation ensembles. Using our feature detection, the most condensed beam is identified by the highest-ranked acceleration feature, i.e., the feature with the highest estimated density $R = |D(f_k)|$. Extraction of the particles that form the best beam can be automated via the query $d_m < 4 * 10^{10}$ && $d_s < 4 * 10^{-6}$. For the six example datasets, this design enables automatic detection and extraction of the main particle beam. In dual- and triple-pulse simulations, the particle beams are typically well-separated and the described setup for automatic extraction of the main beam works well. In the case of single-pulse simulations, the proper boundaries of a beam are often not clear and depend on user requirements (e.g., Fig. 2, $t = 35$). Thus, although the main beam is identified correctly by the highest-ranked acceleration feature—i.e., the reference timestep, location, and estimated beam statistics are known—custom extraction thresholds d_s and d_m may be needed in single-pulse simulation data to fit user requirements.

(R2) Correctly identify all main features: For the single-pulse case, Fig. 2 shows an example of a timeseries of density plots with red boxes highlighting the main particle beams manually identified by an expert user in dataset A. Our feature analysis is consistent with the manual analysis in that it: i) selects the correct beams, ii) identifies appropriate reference timesteps and iii) can furthermore help identify the temporal lack of a beam in case the timesteps of highest density and peak energy disagree (Fig. 11). For dataset A, we find three medium-to-high ranking feature groups, each indicating one of the three main particle beams. We also find two low-ranking groups that identify secondary acceleration features. For dataset B, which models a similar accelerator design to A but at higher temporal fidelity, the analysis finds three comparable medium-to-high ranking feature groups. As expected, due to the higher temporal resolution, the analysis finds more temporal subfeatures per group for dataset B than A. For the dual-pulse simulation C, we find five groups, each representing a main particle bunch associated with a different period of the plasma wave. We also find another feature group, defining the set of particles of the main beam that reaccelerate after the main beam has decelerated. Similarly, we find for the high-resolution, 3D, dual-pulse simulation dataset F, four main feature groups, each indicating a main particle bunch trapped in the first to fourth period of the plasma wave (Fig. 5(b) and 6). In addition, the analysis found three feature groups associated with sets of particles that are lost transversely during the acceleration process (see Sec. 8.2 and Fig. 9). For the two triple-pulse simulations D and E, we each find three beams uniquely identified by a single acceleration feature (Fig. 5(c), dataset D). For E, the main beam is further subdivided into three characteristic subfeatures (not shown).

(R3) Applicability to diverse datasets: These results illustrate that our feature-based analysis can be applied successfully, and without modification, to a large range of simulation datasets modeling different accelerator designs. Furthermore, besides dataset B, all datasets show a low to moderate temporal resolution and vary greatly in spatial resolution. Despite these challenges, the feature detection identified all main features of interest correctly in all cases. These results indicate that the analysis works accurately on data having low or high spatial and temporal resolution and is applicable to 2D and 3D simulation data.

(R4) Intuitively analyze the overall acceleration process: By enabling a top-down analysis workflow—starting from a feature-based view down to the level of individual particles—our analysis facilitates and accelerates the analysis of the overall acceleration process (more detailed use-cases are discussed in Sec. 8.2). Through grouping and selection of acceleration features, the user can quickly identify the main particle bunches and gain an overview of all acceleration features formed during the course of a simulation. All parameters of the analysis are physically motivated, allowing the user to quickly and intuitively explore the space of acceleration features. To achieve maximum flexibility, and due to the multivariate nature of the data, a large range of threshold parameters need to be made accessible to the user. The use of derived summary fields—e.g., the distance fields d_s and d_m (Eq.7,8)—and derived beam statistics, improve accuracy of the particle and

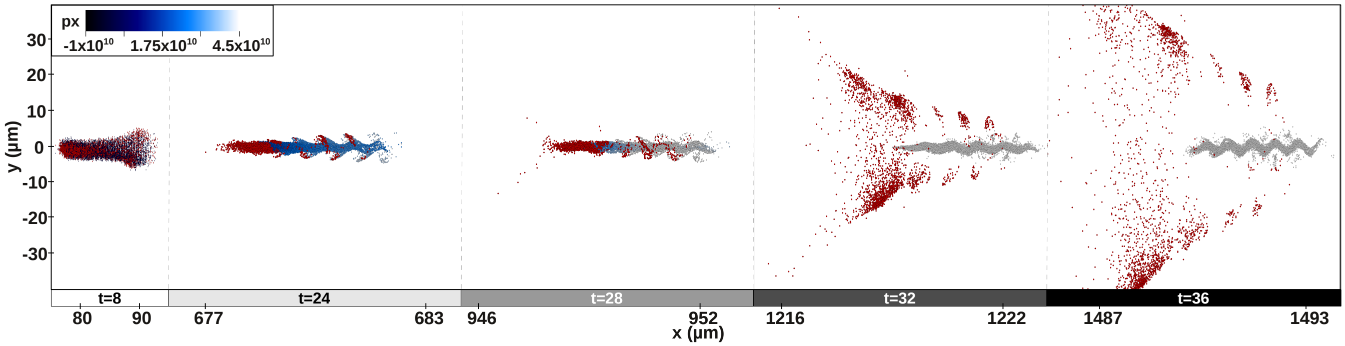


Fig. 9. Evolution of the main particle beam of dataset F. Initially, 21761 simulated particles are part of the main beam. Of these, 19599 particles remain with the main beam (colored particles) and 2162 particles (i.e., about 10%) are lost transversely (red particles). The main beam particles are colored according to their momentum p_x using the indicated color scale.

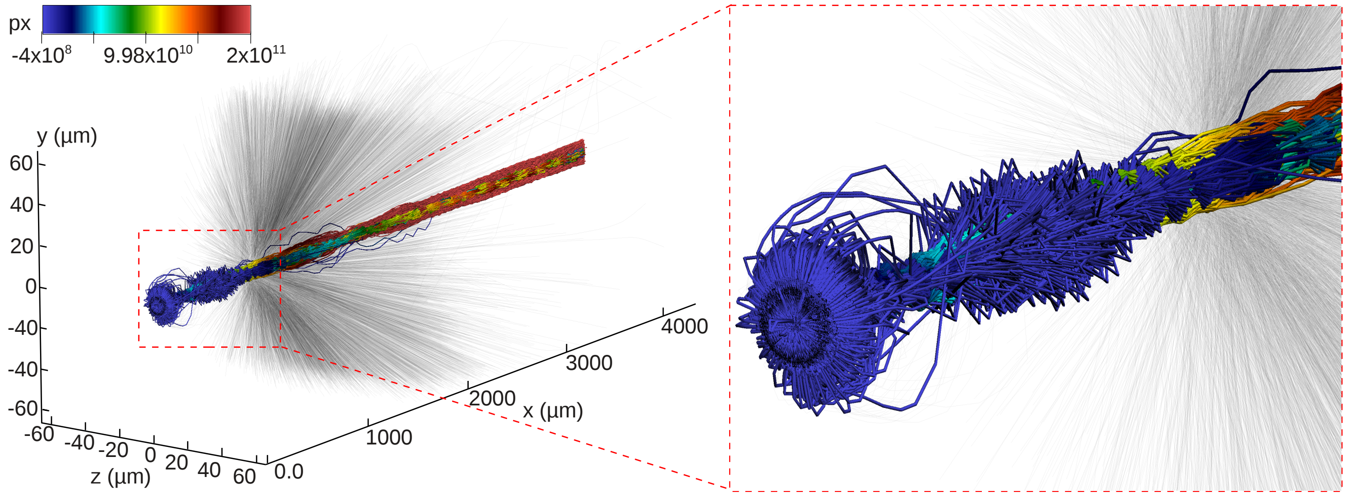


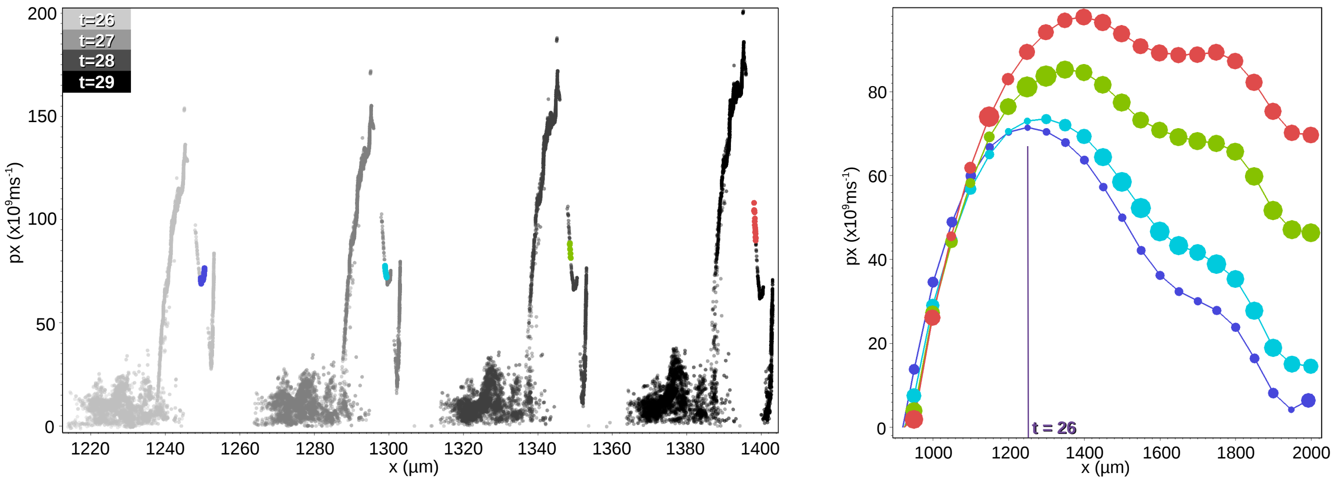
Fig. 10. Particle traces in physical space (x, y, z) for dataset F. Particles that exceed a transverse radius of $\sqrt{y^2+z^2} > 2.5 * 10^{-5}m$ are shown in gray. The particles that persist within the three main beams are colored by their momentum in acceleration direction p_x using the indicated color scale, illustrating their gain in energy over the course of the simulation. The zoom-in shows the particle traces during the injection and early acceleration phase. The particles are initially displaced transversely (plume-like shapes) and are, once trapped in the plasma wave, accelerated in the longitudinal direction x.

feature selection while reducing the amount of parameters needed in practice (see advanced user controls; Sec. 6.3 and 6.4).

8.2 Use Cases

The goal of this section is to demonstrate that our feature-based analysis enables application scientists to address important scientific questions that cannot be addressed adequately using current manual beam analysis methods. A central goal in designing a plasma-based accelerator is to achieve reproducible high-energy, high-density particle bunches with a low energy spread. This leads to a number of desired properties: i) a single condensed bunch of particles should be injected, ii) no accelerated particles should be lost, and iii) the laser pulse should remain focused. To enable optimization of accelerator designs, physicists need to be able to understand the origins, injection process and evolution of the accelerated particles and the acceleration features they define. In the following, we discuss three example use cases, illustrating how our feature-based analysis can be applied in practice to address these challenging analysis tasks.

Beam formation and evolution: Once a user has identified a feature of interest, the associated particles can be interactively traced across time, to study the origin, formation and evolution of the feature. For the single-pulse simulations (datasets A and B), we observe at least three distinct phases of particle injections. The particles that become trapped are in all cases initially located at distinct narrow bands in y (brown particles in Fig. 1(a)). For the dual- and triple-pulse case [30], we observe that all accelerated particles are initially



(a) Reference particle plot

(b) Reference trace plot

Fig. 11. (a) (x, p_x) scatter-plot showing timesteps 26 to 29 of dataset A at which the feature detection found four distinct subfeatures of the second particle beam. All particles used in the analysis are shown in gray colored by time. The reference particles of the four features are highlighted in color. (b) shows the (x, p_x) reference traces of the same four subfeatures as shown in (a) and using the same color scheme as in (a). Size of the circular icons indicates the estimated compactness in momentum space ($CM(D(rp(f_k, t)))$), i.e., the smaller the circular icon the more condensed the feature is estimated to be.

located in physical space in close proximity to the collision site of the laser pulses (Fig. 1(b) and 1(c)). In the acceleration process, these particles separate into different bunches trapped in different periods of the plasma wave. For the triple-pulse case, we observe a clear initial spatial grouping of the primary (magenta) and secondary beam (green) (Fig. 1(c)). Once we have identified the particles of interest and their origins, we can follow the path of the particles further to analyze how the particles are injected, trapped and accelerated (see, e.g., Fig. 10).

Transverse particle loss: In Fig. 7 (top right) we observe for dataset F that the number of particles with $p_x > 10^{10} ms^{-1}$ increases initially monotonically, then stagnates during timesteps 31–35, and then decreases again during timesteps 36–50. As illustrated in Fig. 9, this behavior is due to a transverse loss of particles. Transverse particle loss is critical, as it implies loss of laser energy to particles that are ultimately not part of the main beam. Understanding the causes, progression and severity of particle loss is essential to enable further optimization of accelerator designs.

Transversely lost particles reach their peak in p_x much earlier than the actual beam(s) and are detected by the feature detection as separate acceleration features. In addition to the three feature groups shown in Fig. 6, we find three additional groups of features corresponding to the transverse particle loss for each of the three main beams. Based on the results from the feature detection we can accurately classify the main beams and the associated transverse loss. This allows us to quantify and study the temporal evolution of transversely lost particles in the context of the main beams.

Fig. 9 illustrates the evolution of the main beam and associated set of particles that are lost transversely (red). Initially the red particles are co-located, trapped and accelerated along with the particles of the main beam (blue). As the beam evolves, these particles appear at the “tail” and “outer” parts of the beam and eventually separate from the main beam, exiting the simulation window transversely. Similar behavior can be observed for all main bunches of dataset F. For the main beam, $\approx 10\%$ of the particles that initially form the beam are lost. Fig. 10 shows the 3D traces of all accelerated particles (P, Eq. 3), illustrating the particle injection process and the transverse particles loss later on.

Feature comparison: Earlier approaches towards analysis of particle beams focused on particle beams as a single condensed feature. In our analysis a beam is often identified as a group of acceleration features, each representing a substructure of the beam due to differences in the temporal evolution of the particles. Analysis and comparison of these beam substructures enables a more detailed understanding of the formation and evolution of particle beams than possible when viewing beams as single entities. For example, Fig. 11 compares four automatically detected subfeatures of the second main beam of dataset A (see also Fig. 2). The four particle features reach their peak in p_x momentum at different timesteps and appear at distinct longitudinal (x) locations within the beam (Fig. 11(a)). The locations of the features within the beam and their timepoint of detection follow the same pattern, illustrating the fact that the particles dephase progressively over time. Fig. 11(b) shows that both the lilac and cyan feature are good references for the beam, as both show a low energy spread when the beam reaches its peak energy. The green and red feature appear at the tail of the beam and show much larger energy spread. While the features are similar during the acceleration phase of the beam ($x \leq 1.22 * 10^{-3}m$), the beam stretches in p_x during the deceleration phase ($x \geq 1.22 * 10^{-3}m$), causing the features to diverge in p_x. Similarly, we can compare high-level features (feature groups or beams), by comparing their reference features and associated particle selections.

Using current manual methods, transverse particle loss and temporal subfeatures of particle beams cannot be classified and studied quantitatively. Also, while the main particle beam(s) can often be identified manually, our analysis avoids the need for time-consuming manual investigation and supports more accurate and complete classification of acceleration features, enabling quantitative analysis of large data collections which would be impractical otherwise.

8.3 Performance

The goal of this section is to study the compute performance characteristics of our feature detection algorithm.

Serial Performance: To evaluate the serial performance, we executed the feature detection ten times for each of the six datasets (Table 1) and report the median time (see Fig. 12). We performed all tests on Freedom at NERSC consisting of 4 quad-core AMD 2.4GHz processors and 64GB of memory, while the analysis was restricted to use 1 core and 4GB of memory. For the 2D datasets, the analysis requires less than 28 seconds in all cases. For the 7TB 3D dataset F, the feature detection requires on the order of 100 minutes in serial. The feature detection is a one-time pre-processing and visual feature exploration can afterwards be performed interactively in all cases. The particle tracing and the temporal query are most expensive as they require searches of all particles and perform all raw data I/O required by the analysis. The high temporal resolution of dataset B allows the detection of many fine features, explaining the increased time for computing bunch distances.

Discussion: The good performance we observe even in serial is due to the effective use of advanced data queries for pre-classification and filtering of the data and the efficient implementation of these queries using FastBit. Using FastBit, the evaluation of queries scales linearly with respect to the number of query-hits [31]. The temporal query, particle tracing, and computation of maxima and bunches, hence, have a computational complexity of $O(n|P|)$, with n being the number of timesteps and $|P|$ being the number of particles returned by the temporal query. The computation of the bunch distance fields d_s and d_m has a complexity of $O(n|P|k)$, with k being the number of features ($k << |P|$).

Scalability and Future Work: The main challenge towards scaling of the analysis to even larger, future datasets lies in efficient parallel I/O and data query. The performance of the bunch computation and further data classification depend mainly on the size of the greatly reduced particle subset P and are, therefore, expected to scale well with increasing dataset sizes. Byna et al. [1] have shown that the combination of H5Part and FastBit used here, support high-performance, parallel I/O and query even for datasets consisting of 1 trillion particles per timestep [1]. Parallelizing the I/O and query promise to allow the feature detection to scale well to future data sizes.

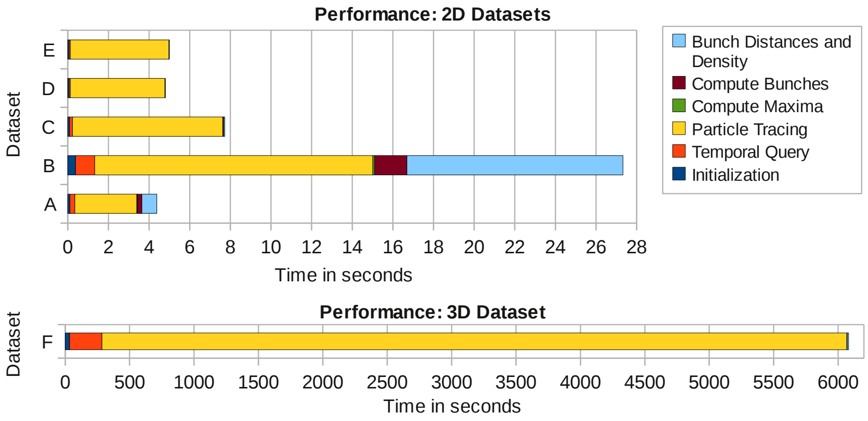


Fig. 12. Median absolute times for the serial analysis of the six datasets. The compute costs are dominated by the times for I/O and query of the temporal query and particle tracing.

9 Conclusion

We presented a novel approach towards feature-based analysis of plasma-based particle accelerator simulation data. To the best of our knowledge, this is the first automatic feature detection approach that: i) enables automatic detection of subtle temporal particle features, ii) supports analysis of transverse particle loss and comparison of beam substructures, and that iii) has been demonstrated to be effective for single, dual as well as triple colliding pulse accelerator simulations. Utilizing advanced query-driven methods for pre-classification and tracing of particles, enables us to incorporate important information about the temporal history of particles in the feature detection not accessible to traditional per-timestep feature detection methods and avoids the need for complex feature-tracking.

The general feature-based analysis approach described here has the potential to improve and simplify feature detection problems in a large range of applications that study time-dependent, particle-based phenomena. For example, while the halo of a particle beam in LINAC simulations is often defined instantaneously based on the radial location of particles at a given timestep, the beam halo is highly dynamic in nature. Similar to the acceleration features studied here, a halo feature could be defined more accurately as dense clusters of particles in space (*bunch detection*) that reach a minimum radial amplitude larger than the maximum desirable beam spot size (*temporal query*) and reach a peak in radial amplitude at the same time (*particle tracing and trace maxima analysis*), while the complete halo consists of a larger number of features with a similar spatio-temporal localization (*feature grouping*). While the specific classification criteria often vary for different applications— e.g., Fusion, LINACS, LPA, flow, or magnetic reconnection— the general design of the analysis and feature exploration approach promises to be broadly applicable.

In our future work we plan to parallelize the I/O and data queries (see Sec. 8.3) and to apply our methods to automatically process large collections of simulation data to study different accelerator designs and parameter sensitivities.

Acknowledgments

This work was supported by the Director, Office of Science, Offices of High Energy Physics and Advanced Scientific Computing Research of the U.S. Department of Energy under Contract No. DE-AC02-04CH11231 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET) and the COMPASS project. The colliding pulse simulations were supported by DOE, NNSA, NA-22. This research used resources of NERSC which is supported by the Office of Science of the U.S. Department of Energy under Contract NO. DE-AC02-05CH11231. We thank the FastBit, H5Part, VORPAL and VisIt team for ongoing development and maintanance efforts.

References

[1] S. Byna, J. Chou, O. Rübel, Prabhat, H. Karimabadi, W. S. Daughton, V. Roytershteyn, E. W. Bethel, M. Howison, K.-J. Hsu, K.-W. Lin, A. Shoshani, A. Uselton, and K. Wu, "Parallel I/O, Analysis, and Visualization of a Trillion Particle Simulation," in *Proceedings of SuperComputing 2012*, Nov 2012.

[2] O. Rübel, Prabhat, K. Wu, H. Childs, J. Meredith, C. G. R. Geddes, E. Cormier-Michel, S. Ahern, G. H. Weber, P. Messmer, H. Hagen, B. Hamann, and E. W. Bethel, "High Performance Multivariate Visual Data Exploration for Extemely Large Data," in *Proceedings of SuperComputing 2008*, Nov 2008.

[3] E. Esarey, C. B. Schroeder, and W. P. Leemans, "Physics of laser-driven plasma-based electron accelerators," *Reviews of Modern Physics*, vol. 81, pp. 1229–1285, 2009.

[4] C. G. R. Geddes, "Plasma Channel Guided Laser Wakefield Accelerator," Ph.D. dissertation, UC Berkeley, 2005.

[5] LOASIS, http://loasis.lbl.gov/.

[6] C. G. R. Geddes, C. Toth, J. van Tilborg, E. Esarey, C. Schroeder, D. Bruhwiler, C. Nieter, J. Cary, and W. Leemans, "High-Quality Electron Beams from a Laser Wakefield Accelerator Using Plasma-Channel Guiding," *Nature*, vol. 438, pp. 538–541, 2004.

[7] W. P. Leemans, B. Nagler, A. J. Gonsalves, C. Toth, K. Nakamura, C. G. R. Geddes, E. Esarey, C. B. Schroeder, and S. M. Hooker, "GeV electron beams from a centimetre-scale accelerator," *Nature Physics*, vol. 2, pp. 696 – 699, 2006.

[8] E. Esarey, R. F. Hubbard, W. P. Leemans, A. Ting, and P. Sprangle, "Electron Injection into plasma wake fields by colliding laser pulses," *Physical Review Letters*, vol. 79, no. 14, Oct 1997.

[9] C. Nieter and J. R. Cary, "VORPAL: A Versatile Plasma Simulation Code," *J. Comput. Phys.*, vol. 196(2), pp. 448–473, 2004.

[10] C. K. Birdsall and A. Langdon, *Plasma Physics via Computer Simulation*, 1st ed., ser. Series in Plasma Physics. Taylor & Francis, Inc., Oct 2004.

[11] D. R. Lipa, R. S. Laramee, S. J. Cox, J. C. Roberts, R. Walker, M. A. Borkin, and H. Pfister, "Visualization for the physical sciences," *Computer Graphics Forum*, vol. 31, no. 8, pp. 2317–2347.

[12] Root is available from http://root.cern.ch/drupal/.

[13] IDL is available from http://tinyurl.com/7x8f5gf

[14] VisIt is available from https://wci.llnl.gov/codes/visit/.

[15] R. A. Fonseca, S. F. Martins, L. O. Silva, J. W. Tonge, F. S. Tsung, and W. B. Mori, "One-to-one Direct Modeling of Experiments and Astrophysical Scenarios: Pushing the Envelope on Kinetic Plasma Simulations," *Plasma Physics and Controlled Fusion*, vol. 50, 124034, Nov 2008.

[16] R. A. Fonseca, L. O. Silva, F. S. Tsung, V. K. Decyk, W. Lu, C. Ren, W. B. Mori, S. Deng, S. Lee, T. Katsouleas, and J. C. Adam, "OSIRIS: A Three-Dimensional, Fully Relativistic Particle in Cell Code for Modeling Plasma Based Accelerators," *Lecture Notes in Computer Science, Computational Science ICCS 2002*, vol. 2331/2002, pp. 342–351, 2002.

[17] S. F. Martins, R. A. Fonseca, L. O. Silva, and W. B. Mori, "On Dynamics and Acceleration in Relativistic Shocks," *Astrophysical Journal Letters (ApJ)*, vol. 695, L189-L193, Apr 2009.

[18] A. Bagherjeiran and C. Kamath, "Graph-based Methods for Orbit Classification," in *SDM*, 2006.

[19] N. S. Love and C. Kamath, "Image Analysis for the Identification of Coherent Structures in Plasma," in *Applications of Digital Image Processing. Edited by Tescher, Andrew G.. Proceedings of the SPIE*, vol. 6696, 2007.

[20] J. Hlína, V. Nwnicka, and J. Sonsky, "Identification of dynamic patterns and their velocities in thermal plasma jets," *Czechoslovak Journal of Physics*, vol. 54, no. 2, pp. 199–210, Feb 2004.

[21] D. Ushizima, O. Rübel, Prabhat, G. Weber, E. W. Bethel, C. Aragon, C. Geddes, E. Cormier-Michel, B. Hamann, P. Messmer, and H. Hagen, "Automated Analysis for Detecting Beams in Laser Wakefield Simulations," in *Proceedings of The Seventh International Conference on Machine Learning and Applications 2008 (ICMLA 08)*. Los Alamitos, CA, USA: IEEE Computer Society Press, 2008, p. 382387.

[22] O. Rübel, C. G. R. Geddes, E. Cormier-Michel, K. Wu, Prabhat, G. H. Weber, D. M. Ushizima, P. Messmer, H. Hagen, B. Hamann, and W. Bethel, "Automatic Beam Path Analysis of Laser Wakefield Particle Acceleration Data," *IOP Computational Science & Discovery*, vol. 2, no. 015005, p. 38pp, Nov 2009.

[23] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The state of the art in flow visualisation: Feature extraction and tracking," *Computer Graphics Forum*, vol. 22, pp. 775–792, 2003.

[24] M. Ferreira de Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: a survey," *IEEE TVCG*, vol. 9, no. 3, pp. 378 – 394, Jul-Sep 2003.

[25] BOOST is available from http://www.boost.org/.

[26] FastBit is available from http://tinyurl.com/73fm6dw.

[27] K. Wu, K. Stockinger, and A. Shosani, "Breaking the Curse of Cardinality on Bitmap Indexes," in *SSDBM*, 2008, pp. 348–365.

[28] K. Stockinger, J. Shalf, K. Wu, and E. W. Bethel, "Query-Driven Visualization of Large Data Sets," in *Proceedings of IEEE Visualization 2005*. IEEE Computer Society Press, Oct 2005, pp. 167–174.

[29] H. Childs, E. S. Brugger, K. S. Bonnell, J. S. Meredith, M. Miller, B. J. Whitlock, and N. Max, "A Contract-Based System for Large Data Visualization," in *Proceedings of IEEE Visualization 2005*, Oct 2005, pp. 190–198.

[30] M. Chen, C. Geddes, E. Esarey, C. B. Schroeder, W. P. Leemans, E. Cormier-Michel, and D. Bruhwiler, "Simulation studies on electron injection by colliding pulses and density modulation in laser plasma accelerators," in preparation.

[31] K. Wu, E. Otoo, and A. Shoshani, "On the Performance of Bitmap Indices for High Cardinality Attributes," in *VLDB*, 2004, pp. 24–35.