Interactive Exploration of Large Event Datasets in High Energy Physics

Max Hermann Alexander Greß Reinhard Klein

Universität Bonn
Institut für Informatik II - Computergraphik
D-53117 Bonn, Germany
{hermann, gress, rk}@cs.uni-bonn.de

ABSTRACT

In high energy physics the structure of matter is investigated through particle accelerator experiments where particle collisions (events) occur at such high energies that new particles are produced. Providing tools for interactive visual inspection of billions of such events occurring in an experiment in an intuitive way is a challenging task. In order to solve this problem we built on previous approaches for visual browsing through image databases and extend them in several ways in order to allow efficient navigation through the collision event datasets. The key features of our novel browsing technique are its applicability to the very large event datasets, a more intuitive selection method for specifying a region of interest, and finally a clustering-based technique that further simplifies and improves the navigation process. We demonstrate the potential of our novel visual inspection system by integrating it into an event display application for the COMPASS experiment at CERN.

Keywords: Interactive browsing, similarity-based visualization, Multidimensional Scaling, Earth Mover's Distance.

1 INTRODUCTION

High energy physics (HEP) investigates the inner structure of matter by performing experiments where highly accelerated particles collide with each other or with a fixed target. Each such collision results in the birth of multiple new particles with individual characteristics (charge, momentum, etc.), which is called an *event*. A particle accelerator experiment utilizes a setup of different detectors to identify events and to be able to reconstruct the trajectories of the new particles produced in an event (called *tracks*) and thus their respective physical characteristics. A number of applications, typically called *event displays*, have been developed for the purpose of visualizing the reconstruction of an event and its tracks. However, current state-of-the-art event displays (e.g. [18, 24, 16, 2, 7, 12, 17]) focus primarily on visualizing single events in various ways [6].

With the ever growing size and energy of modern particle accelerators, the number of events produced in an experiment and used in later analysis constantly increased over the last years. A typical event dataset encountered in analysis consists of millions of events, and hundreds of such datasets are produced in the course of a year. In the COMPASS experiment at CERN [1], about 350 TB raw data per year are produced. Through preprocessing and filtering this raw data is reduced by factor 100, and the results are stored in several 2 GB files containing roughly half a million events each.

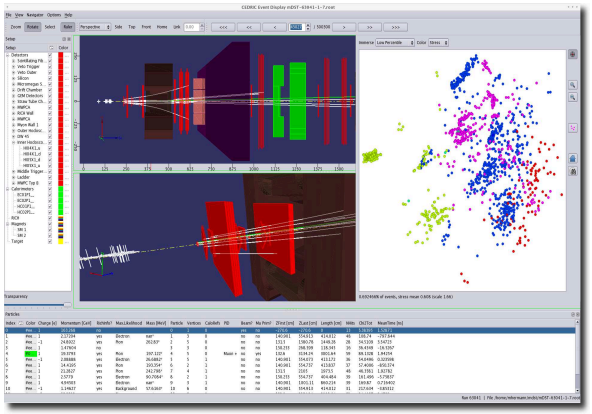


Figure 1: The interactive browsing system (visible to the right) in use in the COMPASS event display.

In event displays, the visualization typically starts by specifying an event dataset, from which events to be visualized can be chosen using very simple techniques, for instance by specifying the identification number of an event or by moving the focus to the preceding or following event in the time line. The CMS event display further has an option to automatically display a random event from the data source every 3 seconds [4].

These present tools for event navigation are neither suited nor designed for purposeful navigation. We therefore believe that a more sophisticated event navigation tool, which permits interactive browsing of the event dataset in an intuitive way will greatly simplify the interactive analysis of physical event data. The in-

teractive event browser described in this paper exactly fulfills this purpose.

Its workflow was inspired by Rubner et al. [22], who proposed a similar navigation for image databases. The basic idea is to represent events on a two-dimensional map where similar events are located close to each other. To produce such two-dimensional maps, we follow Rubner's approach in that we use multidimensional scaling and define a similarity measure for the events based on the so-called Earth Mover's Distance.

To make the approach scalable to very large numbers of events, in terms of usability as well as computation time, only subsets of events are shown on the map at a time, but the map can iteratively be refined by the user through selecting regions of interest.

While the use of two-dimensional maps has a great potential for navigation through a complex dataset, a major problem is the distortion of distances introduced by the dimensionality reduction that is needed for creating the map. This is especially relevant when the user selects a certain region of interest in the map to interactively browse through similar events or to refine the map to a certain subset of similar events. For this purpose, the selection of similar events solely based on a 2D neighborhood would certainly not be adequate, since because of the generally unavoidable distortion, some pairs of events shown close to each other may exhibit significant dissimilarity (even though the distance of points in the map is in general proportional to the dissimilarity of the respective events). Such outliers should not be included in the refined map to avoid unnecessarily high distortions, which would hinder an efficient navigation.

In this paper, we tackle this problem by defining a new criterion for transferring a selected region on the map to a selection of events in the non-Euclidean space (Section 5). This technique accounts for local distortions in the map projection and is robust to the aforementioned kind of outliers. Another important aspect of the new technique is that it can also be applied when subsampling strategies have been applied during the calculation of the maps, on which the user selects the regions of interest, i.e. when only a partial Euclidean embedding of the respective event set has been determined. This is of relevance because such subsampling strategies are inevitable to make the approach applicable also to very large datasets, as we will explain in Section 3.2.

This improved strategy for selecting a region of interest in the dataset allows for a better user control of the navigation process since it avoids refining into regions corresponding to unwanted outliers contained in a selected map region. Additionally, the control over the navigation process can be further improved by integrating a cluster selection technique which allows not only to inspect certain clusters of interest more easily, but also helps to produce less distorted maps during iterative refinement.

The paper is organized as follows: Section 2 sketches the previous work on navigation approaches and also briefly describes the basics of the fundamental algorithms for dimensionality reduction and clustering used in this paper. Section 3 describes how we define a similarity measure for HEP event datasets, which is a prerequisite for being able to determine map representations of events. Additionally, it discusses how we make this approach applicable to large-scale datasets by the use of sampling. Section 4 describes the process of interactive navigation through event datasets including cluster selection and iterative refinement. Furthermore, the applicability and limitations of recent approaches to select a region of interest for the refinement are discussed. Section 5 describes the proposed new technique for specifying the region of interest. Finally, Section 6 demonstrates the usefulness of the proposed navigation technique on examples of real event datasets, and Section 7 concludes with a summary.

2 PREVIOUS WORK

If a similarity measure in form of a metric can be supplied for a specific type of data we speak of *metric data*. We call the distance in the corresponding metric space *dissimilarity* to emphasize the fact that the metric space is in general not an Euclidean space.

2.1 Map Navigation

The general idea of a map representation of metric data is to embed it into the Euclidean $\mathbb{R}^2$ where the distance between two points in the Euclidean space approximates the dissimilarity between the corresponding objects according to the given metric.

In the context of image databases, Rubner et al. [22] describe a navigation technique based on map representations of images. To compute such map representations, a metric for images is proposed based on color distribution. Which images are shown on a map is specified by queries where a query itself is stated in terms of a color distribution. In the navigation process described in [22] the user can create a new map by selecting a point in the current map. For the selected point a query is generated and the k images, which are the most similar to the queried point according to its color distribution, are shown on the new map. The number k of visible images is decreased after each navigation step.

The semantic image browser by Yang et al. [27] also makes use of map representations of images. To select which images are shown on such a map, the user must specify a sample image and a dissimilarity threshold which can be interactively chosen through a scaling bar. Exactly those images are selected whose dissimilarities to the specified sample image are not greater than the specified threshold.

2.2 Earth Mover's Distance

For a dataset whose data items are described by distributions, a metric can be defined by a solution of the so-called mass transportation problem [9, 21]. This solution corresponds to a metric called *Monge-Kantorovich Distance*, or in context of statistics, to the *Mallows Distance* [14]. Also in context of image databases, this similarity measure has been applied in several works [19, 10, 22, 23, 27]. Here, it was given the name *Earth Mover's Distance* (EMD) [22].

The original mass transportation problem was stated by Monge 1781, where he asks how a piece of soil can be moved from fillings to excavations with the least amount of work. Formally Monge's problem can be stated as a linear program in the following way [21]: Let position and masses of the fillings and excavations be given by the discrete distributions $a = \{(x_1, p_1), \ldots, (x_m, p_m)\}$ and $b = \{(y_1, q_1), \ldots, (y_n, q_n)\}$ where the sites x_i and y_j are typically in $\mathbb{R}^d$ and the masses are normalized, i.e. $\sum p_i = \sum q_j = 1$. The work to move a unit amount of mass from site x_i to site y_j is quantified by the real-valued function $c(x_i, y_j) \equiv c_{ij}$, the so-called *ground distance*. A solution is given by an assignment $\mu : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \mathbb{R}$ of how much mass is transported from a filling site to an excavation site. The optimal assignment then is found by solving the following linear program:

$$\min_{\mu} \sum_{i=1}^{m} \sum_{j=1}^{n} \mu(i,j) c_{ij} \tag{1}$$

$$\begin{aligned} \text{subject to} \quad & \mu(i,j) \geq 0 && i=1,\ldots,m \wedge j=1,\ldots,n \\ & \textstyle\sum_{j=1}^{n} \mu(i,j) = p_i && i=1,\ldots,m \\ & \textstyle\sum_{i=1}^{m} \mu(i,j) = q_j && j=1,\ldots,n \end{aligned}$$

As shown in [22], the solution to this linear program is truly a metric.

In summary, solving the mass transportation problem lets us define a metric between discrete distributions by the means of an adequate ground distance c_{ij}.

2.3 Multidimensional Scaling

The first *Multidimensional Scaling* (MDS) method, the so-called classical MDS, was introduced by Torgerson in 1952 [25]. For a detailed overview of this and more recent MDS methods, including its metric and non-metric variant, we refer to [3].

In general, MDS methods determine a coordinate representation of dissimilarity data in low dimensional Euclidean space such that the pairwise coordinate distances approximate the dissimilarity data. Because no specific coordinate can be deferred from the pairwise distances only, a reference coordinate system is choosen with the barycenter of the data as origin and an arbitrary rotation. Thus, the resulting coordinates are unique up to rotation and reflection.

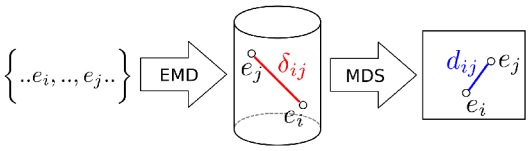


Figure 2: Creation of an event map: First the set of events to be visualized is embedded into the metric event space (depicted as cylinder) and subsequently MDS-projected onto an event map (depicted as rectangle). Two events e_i and e_j are shown with their dissimilarity δ_{ij} and distance d_{ij}.

To estimate the quality of approximation, one can relate the resulting coordinate distances d_{ij} to the original dissimilarities δ_{ij} by the use of a stress function such as *Kruskal stress* [13]:

$$s_{\text{Kruskal}} = \sqrt{\frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}} \tag{2}$$

The Kruskal stress is zero for a perfect reconstruction, while non-zero stress indicates a distortion of the data.

2.4 Clustering

An overview of the numerous work on cluster analysis of high-dimensional data is outside the scope of this paper. We refer to [26] for a recent survey.

Clustering algorithms that operate on metric data, where no representation in an Euclidean space is known, are usually called relational clustering algorithms. An established relational clustering algorithm that partitions the given metric data into a fixed number of clusters is the *Partitioning Around Medoids* (PAM) approach by Kaufman and Rousseeuw [11]. It is similar to the non-relational k-means approach, but tries to find k *representative objects* from the dataset, called *medoids*, that minimize the sum of intra-cluster dissimilarities. In our system, we will use a variation of this approach, called *Clustering Large Application* (CLARA) [11], designed to handle large datasets. It first draws a random sample of the dataset, then uses PAM to find representative objects from this sample, and finally assigns all objects from the dataset to the determined clusters. This is repeated for multiple samples of the dataset, and the best solution is returned.

3 CREATING A MAP OF EVENTS

The similarity measure that we explain in Section 3.1 gives us an embedding of the event dataset into a metric space which we call *event space*. The metric is defined by the pairwise dissimilarity between two events. For arbitrary subsets of this event space a two-dimensional representation of the events can be produced via dimensionality reduction by MDS. The structure of the events can be visualized by showing this representation on a

map, which is what we call an *event map*. The map creation process is summarized in Fig. 2.

In the following we will show how the metric embedding is solved and discuss the computational complexity of the map creation process and the thereby resulting need for subsampling of large datasets.

3.1 Embedding into metric space

An event is fully described by describing the therein produced particles with their trajectories (tracks). The number of tracks in an event is variable and the tracks have no specific order. All tracks can be characterized by the same fixed number p of real-valued physical parameters. Thus a track can be represented by a vector $t \in \mathbb{R}^p$, while an event in turn cannot be considered as a vector of tracks.

In general, it is difficult to define a metric on events because there exists no precise notion of similarity for events in physics. But in contrast to that, defining a metric on tracks is easier because we can exploit the rich set of metrics in $\mathbb{R}^p$. A suitable metric on tracks must take into account (a) the inhomogeneous ranges of the different physical parameters and (b) the correlation between different parameters. Both requirements are met by the statistical Mahalanobis distance

$$d_{\text{Mahalanobis}}(f,g) = \sqrt{(f-g)^T \Sigma^{-1} (f-g)} \qquad (3)$$

where Σ is the covariance matrix for all parameters.

Events are sets of track and, since we can consider a set as a special kind of discrete distribution by assigning each item the same weight, we can use the EMD as described in Section 2.2 to define a metric on events. For this, we formalize an event e as an equally weighted discrete distribution of its n tracks (represented as vectors t_i)

$$e = \{(t_1, 1/n), (t_2, 1/n), \ldots, (t_n, 1/n)\} \qquad (4)$$

and use the metric (3) as ground distance between tracks.

3.2 Making the Approach Applicable to Large-Scale Datasets

As stated in the introduction, our aim is to provide interactive navigation for event datasets, which consist, even when restricted to relatively short time-frames, of millions of events. Similarly to the known navigation approaches for image databases (see Section 2.1), also our navigation approach, which will be described in Section 4, requires frequent recalculation of Euclidean embeddings for different subsets of the dataset. Unfortunately, it is practically infeasible to calculate two-dimensional Euclidean embeddings for millions of events in a way suitable for such an interactive application. This is detailed in the following section. To circumvent this issue we use the strategy of subsampling as described further below.

Feasibility of the map creation

Computing a map representation of a set of events involves first the computation of all pairwise dissimilarities based on the EMD, and second the calculation of the Euclidean embedding via MDS.

According to [19], calculating the EMD is, in general, in $\mathcal{O}(M^3)$, where in our case M corresponds to the average number of tracks per event. For certain specific ground distances such as the L_1-metric, there exist faster algorithms for calculating the EMD [10, 15], which utilize the special structure of the ground distance to solve the linear program more efficiently. However, for the Mahalanobis ground distance (3) used in our approach these improvements are not applicable.

Therefore, in our case the time needed for calculating the full $N \times N$ dissimilarity matrix for a set of N events is in $\mathcal{O}(N^2 \cdot M^3)$. Since also the storage space required for the full dissimilarity matrix is quadratic with respect to the number of events, computing and storing the full matrix quickly becomes infeasible with an increasing number of events. Therefore, instead of calculating the full dissimilarity matrix before constructing the Euclidean embedding via MDS, we calculate the dissimilarities on demand, i.e. at the time when they are needed for the MDS calculation. This however implies that new EMD evaluations may have to be performed whenever new Euclidean embeddings are calculated during the interactive navigation process.

In addition to the time required for calculating dissimilarities, also the time required for calculating the Euclidean embedding via MDS is of relevance in this context. In case of the classical MDS, which has in general a lower time-complexity than the metric or non-metric MDS variant, this calculation requires $\mathcal{O}(N^3)$ time for a set of N events if singular value decomposition (SVD) is used for determining the basis of the Euclidean space [20]. In our case where a dimensionality reduction to only two dimensions is desired, the practical runtime of the MDS can be largely improved by using a Lanczos iteration [5] instead of the SVD to compute only the first two eigenvalues and eigenvectors. In addition, we use the fast approximation technique for evaluating the MDS proposed by [28]. With these improvements we observe in our practical application that the time for computing the MDS projection is rather insignificant in comparison to the time required for the associated EMD evaluations. Nevertheless practical timings show the quadratic dependence of the overall runtime on the number of events.

Subsampling

To make the approach applicable to large-scale datasets despite the computational complexity discussed above, we use sampling strategies as follows. Instead of calculating the Euclidean embedding of the whole event

dataset or of the whole set of events that the viewer is currently interested in, the Euclidean embedding is constructed only for a subset of this set of events, whose size allows for a rapid evaluation and thus for a prompt feedback in the navigation process. We found that for the most real datasets, the selection of a representative subset is possible, that exhibits the same, or at least similar, characteristics as the original complete set of events. For a meaningful map representation of a set of events, first of all its overall structure, which is characterized by the formation of clusters and their positioning in relation to each other, is important for the viewer. Dominant clusters that exhibit a large number of events are retained in the map representation with high probability irrespective of the sampling strategy used. Therefore, the use of random sampling is usually sufficient.

4 INTERACTIVE NAVIGATION

In this section we describe our approach for interactive navigation through huge event datasets by the use of event maps, which we call *event browsing*. There are two motivations for employing such a browsing through several event maps representing smaller and smaller subsets of the dataset, corresponding to successively narrower regions in event space:

- **Stress.** Due to the distortion of the MDS-scaling, an event map of all events cannot convey the fine structures and sub-structures of the dataset. In contrast, a map representation corresponding to a smaller region in event space exhibits less distortion and can thereby convey finer structures.
- **Subsampling.** In most practical cases subsampling of a huge event set is needed to create the event map. On such an event map not all events are accessible. But the subset of a small enough event space region can be shown on an event map without subsampling.

The main navigation technique in this approach is a technique we call *refinement* where the user selects a region of interest on an event map and subsequently a new map is computed based on the selection. Repeated application of this refinement yields maps of smaller and smaller subsets of the dataset until a sufficiently narrow region of interest in event space is reached. We call this interactive process *iterative refinement*.

The second technique used in our system is based on clustering the event dataset. Besides improving the visualization of the dataset structure, clustering can support the navigation process by providing an alternative selection method we call *cluster selection*.

We first describe the cluster selection in Section 4.1, while the discussion of iterative refinement is postponed to Section 4.2.

4.1 Cluster Selection

The integration of clustering techniques (cf. Section 2.4) into the interactive event browser was motivated by the following observation: When analyzing an event dataset using the proposed similarity measure, the contained events typically fall into several clusters and sub-clusters of similar events. Therefore, the detection and labeling of clusters in the event map is an important component of our system to support intelligent user navigation through the dataset.

To visualize the cluster membership of events, the events on an event map can be colored according to their cluster membership. This cluster visualization lets the user recognize regions on the map where due to the MDS projection separate clusters have been mapped on top of each other. Furthermore, the user can select certain clusters and restrict the event map to show only events from this clusters, i.e. in further iterative refinement only events originating from the selected clusters are considered. We call such a restriction of the event set to events from selected clusters *cluster selection*.

Selecting and exploring clusters provides a strategy to solve the problem of overlapping clusters in a map representation. This can be seen as clutter reduction technique [8] but additionally the cluster selection has the advantage, that for the restricted set of events a new map layout is calculated, which is usually less distorted.

4.2 Iterative Refinement

During iterative refinement, the user selects a region of interest on the map from which a new event map is computed which only consists of events inside the region of interest. The underlying idea is to enable the user to examine a subset of the event dataset (which corresponds to a narrower region in event space) in more detail. But due to distortion introduced by MDS projection, a region on the map may contain events which are highly dissimilar to most of the other events inside that specific region. Providing a selection method which is robust to such outliers is non-trivial.

Another requirement for a selection method emerges from the fact that the event set has been subsampled for the calculation of the map. By a selection, we want to determine not only events from the subsample, but rather a region in the event space. Directly mapping the selected region (containing only visible events) into the event space (containing the whole event dataset) is not possible in general since the event space is a pure metric space.

Applicability of recent selection methods

Classical 2D selection techniques like rectangular, elliptic, or freehand selection tools select only a subset of the events visible on the map. Thus they do not meet the requirements for a selection technique on subsampled event maps in the context of iterative refinement.

A selection method similar to the navigation through image databases as proposed by Rubner et al. [22]

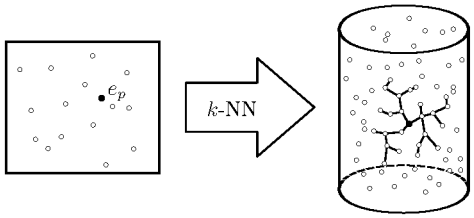


Figure 3: Scheme of previous selection methods.

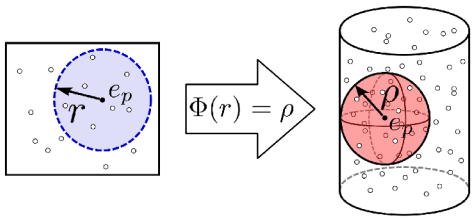


Figure 4: Iterative refinement with the proposed technique. Dependent on a radial selection of radius r around a pivot event e_p, a corresponding set of events in the event space with a maximum dissimilarity ρ to e_p is determined.

(cf. Section 2.1) can be stated in our context of event datasets as follows: Starting from a single selected event e_p, the *pivot event*, the set of k nearest neighbors in the event space is determined, from which the new event map can be computed (see Fig. 3). To allow an *iterative* refinement by means of this method the parameter k should be decreased after every refinement step. The main drawback of this selection method is the missing visual feedback about the selected region on the map that would give the user a spatial impression about the region that will be shown on the refined map. Since this is especially important in the context of large datasets where several subsequent refinements are performed, Yang et al. [27] propose to mark the selected k nearest neighbors on the map. But due to distortion, the k neighbors in metric space will probably not be direct neighbors on the map. Furthermore, in the presence of subsampling, most of them may not lie on the map at all. Thus, because this method does not take into account the described problems due to distortion and subsampling, it is not suited for the iterative refinement.

5 NOVEL SELECTION TECHNIQUE FOR ITERATIVE REFINEMENT

Our novel selection technique incorporates the advantage of classical 2D selection techniques to provide the user a direct visual feedback about the selected region, but further fulfills the requirements described in the context of iterative refinement (see Section 4.2) by considering the distortion as well as the subsampling.

The distortion is taken into account by not considering outlying events in the selected region of interest. As an indicator for outliers we use the distortion that a respective event experiences. This is measured by the so-called local stress which will be defined in Section 5.1. Based on this, we present our novel selection technique in Section 5.2.

5.1 Local Stress as Indicator for Outliers

Even though MDS generally tries to choose the distances of the points in the map proportional to the dissimilarities of the respective events, it is not unlikely, because of the dimensionality reduction to only two dimensions, that some distances in the map differ largely from the respective dissimilarities. The stress of a map (see Section 2.3) describes the overall distortion of dissimilarities. In analogy, to measure the local distortion inside a selected region at a certain point, we define the *local stress* of an event e_i relative to a set E of neighboring events on the map as

$$s_E(e_i) = \sum_{e_j \in E} d_{ij} / \sum_{e_j \in E} \delta_{ij} \tag{5}$$

where d_{ij} is the distance after the projection between e_i and any event e_j in the selection and δ_{ij} the dissimilarity of the corresponding events. $s_E(e_i)$ measures the average distortion of the dissimilarities between event e_i and all other events inside E. A value of 1 denotes that these dissimilarities directly map to the respective map distances on average, and $s_E < 1$ indicates the amount of distortion with respect to set E.

If certain events have a significantly high distortion with respect to E, these events could be fundamentally farther away in metric space than the distance on the map suggests and can thus be considered as outliers. Therefore, the local stress can be used to characterize outliers concerning the current selection that should not be considered in the further refinement.

5.2 Proposed Technique

The idea is to define the selection in the event space as the set of all events which are similar to events contained in the radial selection on the map which are not considered as outliers.

Starting from a pivot event e_p, the user specifies interactively a radial selection region of radius r on the map, see Fig. 4. The thereby selected region defines a set $K = K(e_p, r)$ of events on the map.

To transfer the radial selection on the map to a selection in event space we estimate a dissimilarity ρ in the event space such that all events which exhibit a dissimilarity smaller than ρ towards e_p can be considered to be the region of interest in event space from which the refined map is computed. Choosing a subset in this way from the set of all events has the advantage that events in the subset are similar to most of the events in the selected radial region. Thus they are a good estimation to the region of interest the user wants to explore by the selection. We denote the dissimilarity estimation by $\rho = \Phi(e_p, r)$.

As discussed in the previous section, the distortion leads to outliers. Thus, to take the distortion in the calculation of ρ into account, we identify outliers in the selected region as follows: If the local stress of the respective event deviates more than α times the standard deviation σ_K from the local stress mean $\overline{s_K}$ of all events inside the radial region, then it is considered as outlier. Thereby we can define the set of non-outliers as

$$E_\alpha = \{e_i \in K(e_p, r) \mid s_K(e_i) < (\overline{s_K} - \alpha \cdot \sigma_K)\} \qquad (6)$$

where the parameter α specifies the "strictness" of the outlier classification. Under the assumption of normal distribution, a choice of $\alpha \geq 1$ is reasonable.

Finally, ρ is estimated by the greatest dissimilarity a non-outlying event inside the radial selection exhibits towards the pivot event:

$$\Phi_\alpha(e_p, r) = \max\{\delta_{ip} \mid e_i \in K \setminus E_\alpha\} \qquad (7)$$

However, in the case of very high stress or several overlapping clusters inside the radial selection region, the maximum dissimilarity ρ may be still overestimated. Therefore optionally a λ-quantile can be applied, which means that only the most similar λ percent of the non-outlying events in the radial selection region for estimating ρ are considered.

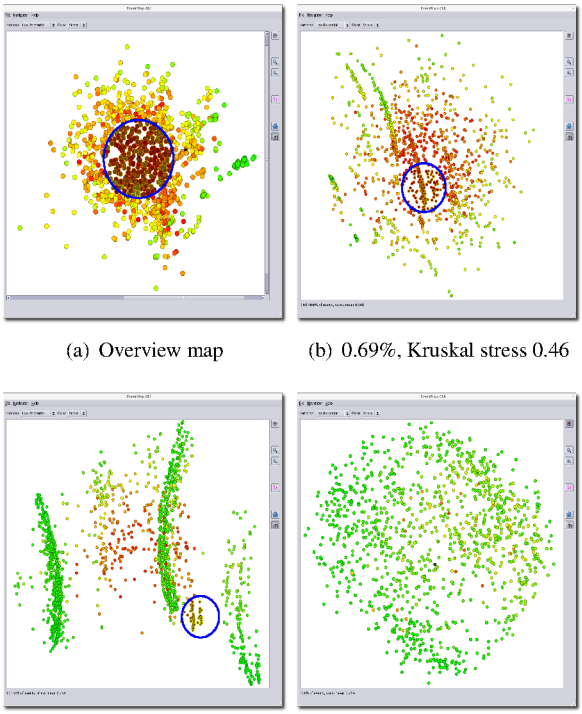
The presented technique indeed gives the user feedback about the selected region, but further is also robust against outliers. Navigation is greatly improved by the fact that in response to a selected region, a new map of events representing that region is shown. Parts of the map can now be examined on different scales by specifying repeatedly regions of differing radii. This is a real improvement in contrast to previous selection methods discussed in Section 4.2.

6 APPLICATION

This section demonstrates the potential of the event browser within a real event display application. We tightly integrated the event browser in an event display, which has been developed for the COMPASS experiment at CERN. Fig. 1 shows a screenshot.

Integration into an Event Display

We connected the event browser to the event display in such a way that every event selected inside the currently visible map is passed to the event display for visualization and further analysis. This allows for a rapid examination of the events on the map. Because nearby events on the map are in fact similar to each other (as approved by physicists), the user can restrict the examination to several events from each visible cluster to get an impression of the type of events represented by that cluster or region on the map. In case of overlapping clusters the user can perform a cluster selection and restrict the investigation to a single cluster.



(a) Overview map (b) 0.69%, Kruskal stress 0.46

(c) 3.01%, Kruskal stress 0.34 (d) 100%, Kruskal stress 0.28

Figure 5: Refinement example. (a)-(d) show successive refinements of the selected regions. Stress is color coded from red (high) to green (low). The percentage gives the fraction of the visible subsample to all events inside region of interest.

Browsing Example

An example browsing workflow is given in Fig. 5, where three successive iterative refinements for a dataset containing about half a million events are shown. Starting with an overview map, which shows a subsample of 1500 events of the complete dataset, in each step a radial region (indicated by the blue circle) is refined. The computation time of a refined map is about 30 seconds for 1500 events on our test system, a 3GHz Pentium-4, and less than 10 seconds for 1000 events. As expected, the overview map exhibits a high stress (visible from the color coding) and no recognizable structure despite the green cluster to the right. But in the course of the following refinements, substructures are revealed in successively less distorted maps (as the Kruskal stress approves). Additionally, since the investigated region in event space gets narrower and narrower, the fraction of visible events increases until potentially all events of the respective region are shown after the last refinement step.

7 CONCLUSION

Based on navigation techniques known from the context of image databases, we have developed an approach for the interactive exploration of large HEP event datasets. A central contribution of this approach is a new criterion for transferring a selected region on the map to a

selection of events in the non-Euclidean, metric space. The proposed technique takes the local stress in the map projection into account and is robust to outliers. This makes the iterative refinement process more intuitive and better controllable for the user compared to previous navigation approaches.

To make the interactive navigation feasible for large datasets, we subsample the event set corresponding to the region of interest in order to obtain a representative subset consisting of not more than a certain fixed number of events before calculating its Euclidean embedding. This is also taken into account when calculating a refined map in such a way that all events in the event space are considered and not just the subsample represented on the map where the region was selected by the user. In addition, we integrated a second navigation technique, namely cluster selection, into our approach to further improve the navigation process.

The practical usability of the proposed approach was verified by applying it to real large-scale datasets from the COMPASS experiment.

It seems also important to note that our improvements of the navigation process in comparison to the recent work are independent of the transferring of these techniques to the domain of HEP event datasets. Therefore, as future work we would like to evaluate these improvements also in context of other large-scale datasets.

REFERENCES

[1] P. Abbon et al. The COMPASS experiment at CERN. *Nuclear Instruments and Methods*, A577:455–518, 2007.

[2] G. Barrand. Panoramix. In *Proc. of Computing in High Energy Physics (CHEP'04)*, 2004.

[3] I. Borg and P. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 1997.

[4] CMS Collaboration. *The CMS Offline WorkBook, Chapter 4.8 The CMS Event Display*. Available online at https://twiki.cern.ch /twiki/bin/view/CMS/WorkBook.

[5] P. Deuflhard and A. Hohmann. *Numerical Analysis in Modern Scientific Computing: An Introduction*. Texts in Applied Mathematics 43. Springer, 2003.

[6] H. Drevermann, D. Kuhn, and B. Nilsson. Event display: Can we see what we want to see? In *CERN School of Computing*, 1995.

[7] J. Drohan et al. The ATLANTIS visualisation program for the ATLAS experiment. In *Proc. of Computing in High Energy Physics (CHEP'04)*, pages 361–364, 2004.

[8] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Trans. Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[9] F. L. Hitchcock. The distribution of a product from several sources to numerous localities. *J. Math. Phys.*, 20:224–230, 1941.

[10] T. Kaijser. Computing the Kantorovich distance for images. *J. Math. Imaging and Vision*, 9(2):173–191, 1998.

[11] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 1990.

[12] O. Kind, J. Rautenberg, et al. A ROOT-based client-server event display for the ZEUS experiment. In *Proc. of Computing in High Energy Physics (CHEP'03)*, 2003.

[13] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

[14] E. Levina and P. Bickel. The earth mover's distance is the Mallows distance: Some insights from statistics. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'01)*, 2001.

[15] H. Ling and K. Okada. EMD-L_1: An efficient and robust algorithm for comparing histogram-based descriptors. In *Proc. of the 9th European Conference on Computer Vision (ECCV'06)*, pages 330–343, 2006.

[16] Z. Maxa et al. Event visualization for the ATLAS experiment - the technologies involved. In *Proc. of Computing in High Energy Physics (CHEP'06)*, 2006.

[17] D. McNally. Event visualization tools at LEP. In *Proc. of the HEPVis 96 Workshop*, CERN, 1996.

[18] I. Osborne et al. CMS event display and data quality monitoring at LHC start-up. In *Proc. of Computing in High Energy Physics (CHEP'07)*, 2007.

[19] S. Peleg, M. Werman, and H. Rom. A unified approach to the change of resolution: Space and grey-level. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 11(7):739–742, 1989.

[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1997.

[21] S. T. Rachev and L. Rüschendorf. *Mass Transportation Problems*, volume 1: Theory. Springer, 1998.

[22] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proc. of the 6th International Conference on Computer Vision (ICCV'98)*, pages 59–66, 1998.

[23] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Intern. J. Comp. Vision*, 40(2):99–121, 2000.

[24] M. Tadel and A. Mrak-Tadel. AliEVE - ALICE event visualization environment. In *Proc. of Computing in High Energy Physics (CHEP'06)*, 2006.

[25] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17:401–419, 1952.

[26] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Trans. Neural Networks*, 16(3):645–678, 2005.

[27] J. Yang, J. Fan, D. Hubball, Y. Gao, H. Luo, W. Ribarsky, and M. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *Proc. of IEEE Symposium on Visual Analytics Science and Technology*, pages 191–198, 2006.

[28] T. Yang, J. Liu, L. McMillan, and W. Wang. A fast approximation to multidimensional scaling. In *Proc. of the IEEE Workshop on Computation Intensive Methods for Computer Vision*, 2006.