Analysis of Crash Simulation Data using Spectral Embedding with Histogram Distances

Anna-Luisa Schwartz

Institut für Numerische Simulation
Universität Bonn
Wegelerstr. 6
53115 Bonn
luisa.schwartz@uni-bonn.de

Abstract: Finite Element simulation of crash tests in the car industry generates huge amounts of high-dimensional numerical data. Methods from Machine Learning, especially from Dimensionality Reduction, can assist in analyzing and evaluating this data efficiently. Here we present a method that performs a two step dimensionality reduction in a novel manner: First the simulation data is represented as (normalized) histograms, then embedded into a low dimensional space using histogram distances and the nonlinear method of Spectral Embedding/Diffusion Maps, thus enabling a much easier data analysis. In particular, this method solves the problem of comparing simulation data with small changes in the Finite Element grids due to variations of geometry or unequally fine grid structures.

1 Introduction

Numerical simulation of car crash tests has evolved since the 1980's to be an integral part of the development process for new vehicles. Due to the rapid growth in computational capacity, enabling the generation of more and much more detailed simulations, the size of the emerging simulation data increased enormously. Today, for one simulation run data is saved for more than a million finite element nodes at several hundred time steps; tests to decide on small parameter variations (plate thickness, positions for drillings, material variations) require bundles of several thousand simulation runs.

Conventionally, evaluation of this data is performed by an engineer examining several 3D-visualisations of the simulation runs simultaneously and grouping them according to observed effects. Accelerating and simplifying this difficult and time-consuming analysis through the use of Machine Learning methods is a subject of recent research [Boh13].

As the data is highly complex with dimensionality of an order of 10^8, in particular exceeding the number of data points by a factor of hundred thousand, methods of dimensionality reduction need to be applied: Since all simulation runs in a bundle concern the same car model at the same stage of development, the data exhibits many redundancies and it is to be expected that a lower-dimensional embedding will still be able to parameterize the data

according to its so-called intrinsic geometry. A 2- or 3-dimensional embedding can then present a visual overview over the relations between simulation runs in a bundle and thus be an important tool for the development engineer.

In this paper we focus on a selected method of nonlinear dimensionality reduction: Spectral Embedding, in particular Diffusion Maps, presented 2006 by Coifman and Lafon [Coi06] and already applied successfully for simulation data analysis [Erb07, Boh13]. This method is based on the use of a kernel function describing the similarities of different data points. Commonly the similarity function is derived from distance functions on the data set. So far, when applying Diffusion Maps on car data, one regards the Euclidean distance between so-called displacement vectors, containing the absolute values of the displacements at every finite element node [Boh13, Iza14]. This approach requires identical dimensions of all displacement vectors considered, meaning identical numbers of FE nodes for every simulation run. In practice, this is not always the case. Often simulations are ran to compare the effect of varying certain structural parts, thus leading to simulation runs with different geometries and consequently different finite element meshes. To compare these simulation runs, projections on auxiliary meshes can be employed, which are however computationally costly and might not be very accurate. An alternative choice of distance functions seems promising to overcome this problem.

Here, we will for the first time investigate the approach to calculate distances between simulation runs via a usage of histograms of the displacements. Representing the high-dimensional data as histograms with fewer bins than the original dimensions means a dimensionality reduction already in a preprocessing step, which will then be further refined by the application of Diffusion Maps, using an appropriate histogram distance.

The theoretical foundations of Diffusion Maps will be presented in Section 2 of this paper, as well as a short review of histogram distances. In Section 3 we introduce our proposed method and the algorithm, followed by the results obtained by applying this approach to several sets of industrial data and a discussion in Section 4.

2 Fundamental Concepts

In the following, we will refer to the data set as $X = \{x_1, x_2, \ldots, x_m\} \subset \mathbb{R}^n$, where each of the m data points represents a full simulation run. Dimensionality reduction means the search for a map (the so-called embedding) $f : X \longrightarrow \mathbb{R}^p$ with $p \ll n$, preserving intrinsic information on the data (mostly geometric information, such as certain distances). If we assume that the data set X lies on a nonlinear manifold, we also need to apply nonlinear methods to obtain a suitable embedding. Spectral Embedding methods like Diffusion Maps find such an embedding using a kernel matrix describing local similarities between data points.

2.1 Review of Diffusion Maps

Diffusion Maps [Coi06] are based on the notion of a so-called diffusion distance, motivated as follows: Imagine the data set X to be the node set of a complete weighted graph (G, ω) with $G = (X, X \times X)$. The edge weights are assigned depending on the local similarity of the adjoint nodes:

$$\omega : X \times X \to \mathbb{R}, \omega((x_i, x_j)) = k(x_i, x_j),$$

with k being a kernel function. Here one normally uses the Gaussian kernel, $k_\varepsilon(x_i, x_j) := \exp\left(\frac{d(x_i,x_j)^2}{\varepsilon}\right)$ with a scaling parameter $\varepsilon > 0$ and d the Euclidean distance; for our application however we will apply the Gaussian kernel with an application-specific distance function $d : X \times X \to \mathbb{R}$. By normalizing the edge weights with the degrees of the correspondent nodes we obtain a Random Walk on X with transition probabilities $p(x_i, x_j) = \omega((x_i, x_j))/\sum_{y \in X} \omega((x_i, y))$.
If the points are similar according to the chosen distance function, the respective transition probabilities will be high, for dissimilar points they will be low. Running the Random Walk forward in time, we obtain information about regions of high affinity within the data set, leading naturally to a definition of the diffusion distance, depending on a time parameter $t \in \mathbb{R}$:

$$D_t(x_i, x_j)^2 := \sum_{y \in X} (p_t(x_i, y) - p_t(y, x_j))^2 \frac{1}{\pi(y)}$$

with π denoting the stationary distribution of the Markov Chain and $p_t(x_i, x_j)$ the probability of transition from x_i to x_j in t steps [Coi06]. Thus, the diffusion distance sums over all possible connections between two points in t time steps, intuitively leading to a high robustness to noise.

To approximate this distance in practice, we consider the spectral decomposition of the operator p_t or, equivalently, of the t-th power of the transition matrix $\mathbf{P} = (p(x_i, x_j))_{ij}$. As $\mathbf{P}$ is not symmetric, we conjugate it by the diagonal degree matrix $\mathbf{D}$ with $\mathbf{D}_{ii} := \sum_j k(x_i, x_j)$.

The new matrix $\mathbf{A} := \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}$ is symmetric and shares the spectrum with $\mathbf{P}$. We know from the Perron-Frobenius theorem that all eigenvalues λ_i are real, and $1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n$.

We denote the eigenvectors of $\mathbf{A}$ by $\{v_j\}$. Right and left eigenvectors $\{\phi_j\}$ and $\{\psi_j\}$ of $\mathbf{P}$ are obtained by conjugation:

$$\phi_j = \mathbf{v}_j \mathbf{D}^{\frac{1}{2}}, \qquad \psi_j = \mathbf{v}_j \mathbf{D}^{-\frac{1}{2}}.$$

Hence we can express the diffusion distance at time t by the spectral decomposition of $\mathbf{P}$ [Coi06]:

$$D_t(x_i, x_j)^2 = \sum_{l=2}^{n} \lambda_l^{2t} (\psi_l(x_i) - \psi_l(x_j))^2.$$[1]

[1] Remark: $\psi(x)$ denotes for an $x \in X$ the i-th entry of $\psi \in \mathbb{R}^m$, with i the index of $x = x_i$ in the indexed set X

The term for $l = 1$ is left out, as the eigenvector ψ_1 belonging to the eigenvalue $\lambda_1 = 1$ is constant.

Due to the non-increasing property of the spectrum we can approximate the diffusion distance by using only the first $p < n$ terms:

$$D_{t,p}(x,y)^2 := \sum_{l=2}^{p+1} \lambda_l^{2t}(\psi_l(x) - \psi_l(y))^2$$

Thus we can achieve a dimensionality reduction by embedding the data set into $\mathbb{R}^p$ by the diffusion maps

$$\Psi_{t,p} : \mathcal{Y} \longrightarrow \mathbb{R}^p, \qquad x \mapsto (\lambda_2^t\psi_2(x), \lambda_3^t\psi_3(x), \ldots, \lambda_{p+1}^t\psi_{p+1}(x))^T, \tag{1}$$

preserving the local diffusion distances in the originating space X up to an error depending on the ratio $\frac{\lambda_{p+1}^t}{\lambda_2^t}$.

To take out statistically unsuitable side effects, we can replace isotropic kernel functions such as k_ε by anisotropic ones [Nad08]. Here, instead of k_ε we applied the anisotropic Gaussian kernel, $\tilde{k}_\varepsilon(x,y) := k_\varepsilon / \sum_{z\in X} k_\varepsilon(x,z) \sum_{z\in X} k_\varepsilon(y,z)$, where the terms in the denominator are used as an approximation of the probability density by which the dataset X was drawn from the underlying manifold.

2.2 Review of Histogram Distances

For a consistent notation, let us formalize the notion of a (multidimensional) histogram:

Definition 1 (Histogram)**.** *For a finite set $\mathcal{I} \subset \mathbb{N}^d$ and a map $h : \mathcal{I} \longrightarrow \mathbb{R}_{\geq 0}$ we call the set $H = (h(\mathbf{i}))_{\mathbf{i}\in\mathcal{I}}$ a* histogram*. The elements of $\mathcal{I}$ represent* bins*, the assigned values $h(\mathbf{i})$ are the respective* weights*.*

To compare histograms, one can distinguish between two main classes of distances: *bin-to-bin-distances* and *cross-bin-distances*. The former only take into account the weights of matching bins, meaning they can be written as:

$$d(H_1, H_2) = \sum_{\mathbf{i}\in\mathcal{I}} f(h_1(\mathbf{i}), h_2(\mathbf{i})) \text{ for histograms } H_1 = (h_1(\mathbf{i}))_{\mathbf{i}\in\mathcal{I}} \text{ and } H_2 = (h_2(\mathbf{i}))_{\mathbf{i}\in\mathcal{I}}$$

with a suitable function $f : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}_{\geq 0}$. In particular this means that the histogram topology is completely disregarded.

Cross-bin-distances on the other hand take into account neighboring or even all bins as well. A discrete *ground distance* $d_{\mathbf{ij}}$ is utilized to define distances between the bins represented by $\mathbf{i}$ and $\mathbf{j}$. This facilitates the formulation of distances that correspond well with human perception and at the same time are more robust with respect to the choice of bin ranges. However, cross-bin-distances are obviously more computationally expensive.

Popular examples for histogram distances include the L_1- and L_2-distances of the histograms' weight vectors (bin-to-bin-distances). Note that there also exists a cross-bin histogram distance based on diffusion distances [Lin06], which undertakes the diffusion approach from a different angle than Diffusion Maps and is not further considered here. In this paper, we focus on the following distances, both widely used in image processing:

χ^2-distance The χ^2-distance is a bin-to-bin-distance, following the intuition that differences between 'large' bins are less significant than those between 'small' bins [Pel10]. It is based on the χ^2-test used in statics and is defined as:

$$\chi^2(H_1, H_2) = \frac{1}{2} \sum_{\mathbf{i} \in \mathcal{I}} \frac{(h_1(\mathbf{i}) - h_2(\mathbf{i}))^2}{h_1(\mathbf{i}) + h_2(\mathbf{i})}$$

Earth-Mover's-Distance (EMD) EMD [Rub00] is a cross-bin-distance, its name originating from the analogy to imagine one histogram as a set of earth heaps and the other as a set of holes, with a ground distance assigned to every pair of bins. The EMD then is the minimum cost of moving the earth into the holes and is obtained by solving a discrete minimum cost flow problem.
For the rather technical precise definition and set-up, we refer to [Rub00].

3 Method

We propose the following analysis procedure to compare the simulation data (cf. [Iza14]):

1. Extraction of displacement data of charasteric integral parts from the simulation data.
2. Preprocessing of data. *Here*: representation of the n_i-dimensional displacement vectors $\mathbf{v}_i$ as one-dimensional histograms $\left(h^i(j)\right)_{j=1}^{n_{\text{bins}}}$.
3. Creation of a kernel k_ε, utilizing a suitable histogram distance function as shortly discussed in Section 2.2.
4. Dimensionality reduction by Spectral Embedding. *Here*: Use of Diffusion Maps (as in Equation (1)).
5. Postprocessing and analysis of embedded data (e.g. by application of clustering algorithms). *Here*: 2D-visualisation of embedding space.

The histogram representation in step 2 allows us to apply Diffusion Maps also to compare simulation runs with varying geometries. It needs to be noted that this step represents a first layer of dimensionality reduction, where some information is lost – in our use case, this especially means that information about the positions of the displacements is discarded and only the extent of the deformation is stored.

Input: List *filelist* of m simulation data sets $\{x_1, \ldots, x_m\}$, parameter n_{bins}, time step t
Output: Histograms $\{(h^1(j))_{j=1}^{n_{\text{bins}}}, \ldots, (h^m(j))_{j=1}^{n_{\text{bins}}}\}$, stored as a matrix $\mathbf{W} \in \mathbb{R}^{n_{\text{bins}} \times m}$
where $w_{kl} = (h^l(k))$
Initialize displacement matrix $\mathbf{V} \in \mathbb{R}^{n \times m}$ and histogram matrix $\mathbf{W} \in \mathbb{R}^{n_{\text{bins}} \times m}$;
$i = 1$;
for *file* $\in$ *filelist* **do**
  Read and store displacement vectors in x, y, and z-direction at time step t
  $\mathbf{v}^x, \mathbf{v}^y, \mathbf{v}^z \in \mathbb{R}^{n_i}$ from *file* (with n_i number of finite element nodes in model);
  Calculate absolute displacement vector $\mathbf{v}^{\text{abs}}$: $\forall k : v_k^{\text{abs}} := \sqrt{(v_k^x)^2 + (v_k^y)^2 + (v_k^z)^2}$;
  $\mathbf{V}[:, i] := \mathbf{v}^{\text{abs}}$;
  $i{+}{+}$;
end
Determine minimum $\min_{\text{glob}}$ and maximum $\max_{\text{glob}}$ of entries in displacement matrix $\mathbf{V}$;
Define n_{bins} identically large classes for histograms (according to $\min_{\text{glob}}$ and $\max_{\text{glob}}$);
for $k \in \{1, 2, \ldots, n_{bins}\}$ **do**
  Generate a histogram $(h(j))_{j=1}^{n_{\text{bins}}}$ of displacement vector $\mathbf{V}[:, k]$ according to the defined class partitioning;
  Normalize histogram to total weight of 1;
  Store histogram weight vector in $\mathbf{W}[:, k]$;
end

Algorithm 1: Extraction and Preprocessing

Input: Matrix $\mathbf{W} \in \mathbb{R}^{n_{\text{bins}} \times m}$ of histogram weight vectors, originating from Algorithm 1;
(histogram-) distance[a] d; scaling parameter ε for Gaussian kernel, target dimension p
Output: Diffusion coordinates of data for the embedding into $\mathbb{R}^p$
Initialize distance matrix $\mathbf{D} \in \mathbb{R}^{m \times m}$;
for $i \in \{1, 2, \ldots, m\}$ **do**
  for $j \in \{1, 2, \ldots, m\}$ **do**
    $\mathbf{D}[i, j] := d_{ij} := d(\mathbf{W}[:, i], \mathbf{W}[:, j])$;
  end
end
Generate kernel matrix $\mathbf{K}^{(1)}$ with entries $k_{ij}^{(1)} = \exp\left(-\frac{d_{ij}^2}{\varepsilon}\right)$;
$\mathbf{p} := \mathbf{K}^{(1)} \cdot \mathbf{1}$, where $\mathbf{1} = (1 \ldots 1)^T$;
Generate[b] matrix $\mathbf{K}^{(2)} := \mathbf{K}^{(1)}./(\mathbf{p} \cdot \mathbf{p}^T)$ (approximation of anisotropic Gaussian kernel) ;
$\mathbf{v} :=$ sqrt$(\mathbf{K}^{(2)} \cdot \mathbf{1})$;
Generate symmetric matrix $\mathbf{K} := \mathbf{K}^{(2)}./(\mathbf{v} \cdot \mathbf{v}^T)$ conjugated to $\mathbf{K}^{(2)}$;
Perform spectral decomposition of $\mathbf{K}$ to obtain eigenvalues and eigenvectors $\{\lambda_i\}_{i=1}^{n_{\text{bins}}}, \{\psi_i\}_{i=1}^{n_{\text{bins}}}$;
Return $\mathbf{M}^T$ with $\mathbf{M} := [\lambda_i \psi_i]_{i=2}^{p+1}$;

Algorithm 2: Diffusion Maps with histograms for crash simulation data

[a] For cross-bin-distances the input d comprises in particular the respective ground distance
[b] The operation ./ denotes elementwise division

We state the combined algorithms for step 1 and 2 in Algorithm 1 and for steps 3 and 4 in Algorithm 2, adapted from [Laf04]. The matrix $\mathbf{M} \in \mathbb{R}^{m \times p}$, generated from the output of the second algorithm by $\mathbf{M}[:, i] = (\lambda_i \psi_i)^T$ for $i \in \{ 2, 3, \ldots, n_{\text{bins}} \}$, comprises in its rows the diffusion coordinates of the m data points for an embedding into $\mathbb{R}^p$. Note again that the first eigenvector to the eigenvalue $\lambda_1 = 1$ is not taken into account, as it is the constant vector and thus does not store any relevant information.

4 Results and Discussion

To analyze the discrimination ability of the algorithm, it has been implemented in Python[2] and applied to the following data sets:

Truck-Beam Structural beam of a Chevrolet C2500 Pick-Up-Truck at frontal collision, $m = 132$ simulation runs with variation of 9 parameters, each $n = 1714$ FE-nodes [Boh13].

Car-Lateral Lateral part of a medium-sized vehicle at lateral collision with varying geometry (40 to 44 integral parts, original data from German car manufacturer), $m = 143$ simulation runs with different FE-meshes, each $n_i \approx 26.100$ FE-nodes.

(a) Structural Beam (b) Bending mode 1 (c) Bending mode 2

Figure 1: *Truck-Beam* data set and different bifurcation modes

As it is difficult to find an objective measure for the algorithm's quality we chose three evaluation criteria: Simulation runs that are similar according to their visualizations should be grouped together, runs that bear no similarities to others should be recognizable as outliers and in case of bifurcations the modes should be clearly separated.
To check the consistency of the new approach we first applied the algorithm to the *Truck-Beam* data set, which can also by analyzed with the conventional method. The data set is illustrated in Figure 1. It features a clearly visible bifurcation in its bending behavior, which we wish to find in the embeddings as well. Figure 2 shows a two dimensional embedding of the *Truck-Beam* data set according to the diffusion coordinates obtained with Algorithm 2. The data points are colored corresponding to the value of the second eigenvector (resp. the first diffusion coordinate), which according to spectral clustering theory comprises the most information about the global structure of the data set [Lux07]. It can be seen, that both axes correlate with the beam's bending behavior (color-coded in the small images): the second eigenvector separates the bifurcation modes by its sign, the value of the third eigenvector is proportional to the extent of the deformation in the middle segment of the beam.

[2]We used Python 2.7.1 with an extensive use of the packages *NumPy* and *matplotlib*

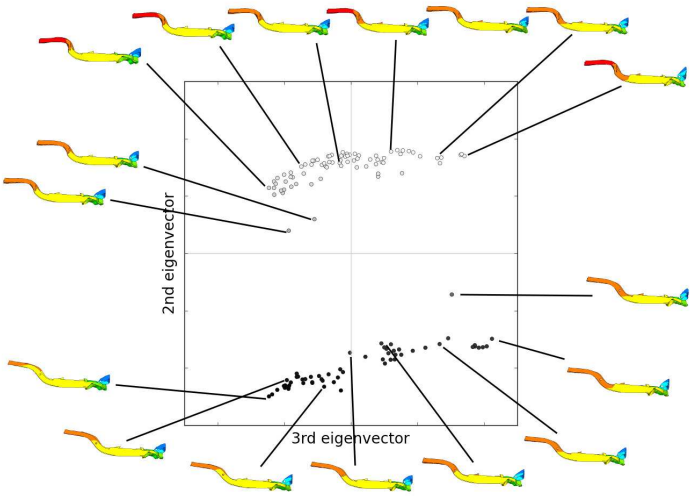
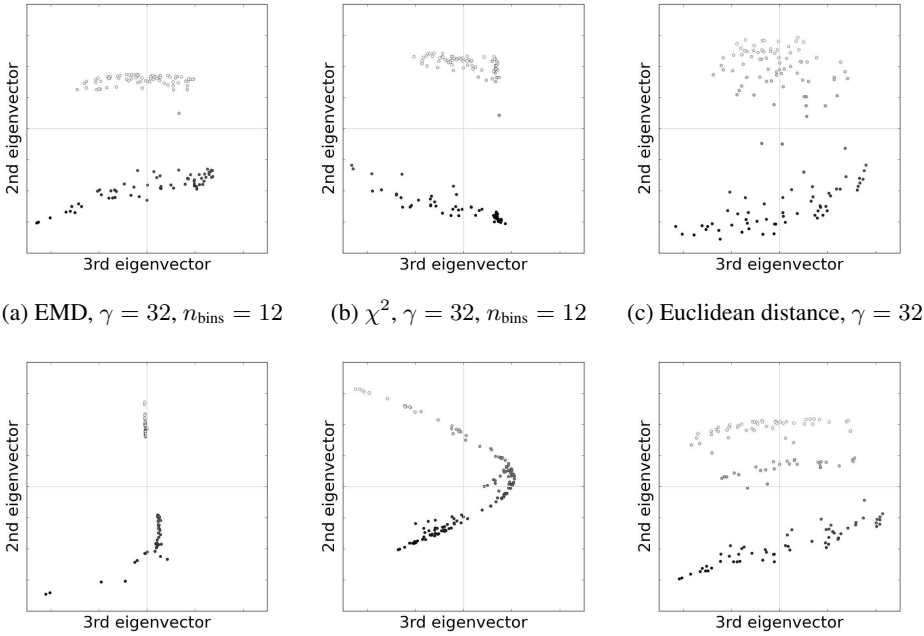


Figure 2: Embedding of the *Truck-Beam* dataset as in Figure 3a (parameters: $\gamma = 32, n_{\text{bins}} = 10$), with selected visualizations of deformations.

The first row of subfigures in Figure 3 shows the results obtained with different distance functions: 3a with histograms and the EMD histogram distance, 3b with histograms and the χ^2-distance function and 3c with the standard Euclidean distance of the displacement vectors without any histograms (which is applicable here as the FE grids for the simulation runs are identical). One can observe that the first two embeddings are similar concerning the structure of the clusters and the positioning of the data points. The Euclidean distance embedding is rather appropriate concerning the data as well, but the two groups are not as clearly separated as in the other embeddings, and the two central points are not placed to the correct bifurcation mode by the second eigenvector. These differences between histogram distances and displacement vector distances are confirmed for variations of the scaling parameter ε (details on the parameter choice see below) and are rather surprising, since by applying histograms in the preprocessing step and thus performing dimensionality reduction, we discarded some information. Considering the embeddings obtained, it must be assumed that a large part of this lost information was noise, which would make the technically less accurate new method more appropriate for this data set.

For the *Car-Lateral* data set with varying geometry the simulation runs are also grouped in an appropriate manner (see Figure 4). The two groups (upper and lower), as well as the two outliers, match the data well, especially along the second eigenvector. As the data are confidential industrial data provided by the manufacturer, visualizations cannot be shown here. It should be noted that this data set could not be analyzed with the conventional diffusion maps method, as the FE grids differ among the simulation runs. Therefore, our new method seems well applicable for the intended purpose in this first test.



(a) EMD, $\gamma = 32$, $n_{\text{bins}} = 12$ (b) χ^2, $\gamma = 32$, $n_{\text{bins}} = 12$ (c) Euclidean distance, $\gamma = 32$

(d) EMD, $\gamma = 1$, $n_{\text{bins}} = 12$ (e) EMD, $\gamma = 32$, $n_{\text{bins}} = 5$ (f) EMD, $\gamma = 32$, $n_{\text{bins}} = 20$

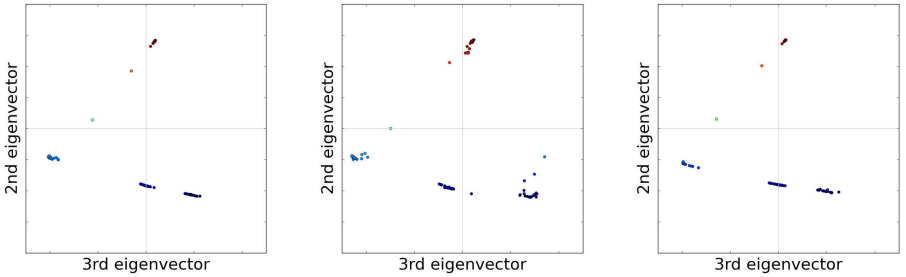
Figure 3: Different embeddings of the *Truck-Beam* dataset

4.1 Parameter choices and variations

The most important parameters to be varied in our proposed method are the ε of the Gauss kernel for Diffusion Maps and the n_{bins} for the histogram construction. The Gauss kernel's parameter determines the size of the neighborhood around a point x where the value $k_\varepsilon(x, y)$ differs from zero significantly. While there are no objective rules about suitable values, some rules of thumb have been established. As proposed for Diffusion Maps by Lafon we use the average distance between a point and its nearest neighbor [Laf04] and additionally scale it with a parameter γ: $\varepsilon := \frac{\gamma}{m} \sum_{x \in \mathcal{Y}} \min_{y \in \mathcal{Y}} \{\, d(x, y) \mid d(x, y) \neq 0 \,\}$.

For the *Truck-Beam* data set all tested choices of γ lead to reasonable embeddings, but changes apply to the axes' functions: For $\gamma = 1$ (cf. Figure 3d) the second eigenvector in addition to the information about the bifurcation modes comprises the information about the total deformation of the beam. Third and fourth eigenvector (not depicted here) take the function of indication vectors, as they each only comply information about one of the bifurcation modes and become zero for the other. For growing γ the embedding changes, to finally stay stable from $\gamma = 16$ with the point cloud shaped as in Figure $3a$.

Using histograms, the choice of n_{bins} (or the interval length $h \sim \frac{1}{n_{\text{bins}}}$ respectively) is crucial for the expressiveness of the histogram. Too small as well as too large classes will not do



(a) EMD, $\gamma = 32$, $n_{\text{bins}} = 16$ (b) χ^2, $\gamma = 32$, $n_{\text{bins}} = 16$ (c) EMD, $\gamma = 32$, $n_{\text{bins}} = 5$

Figure 4: Different embeddings of the *Car-Lateral* dataset

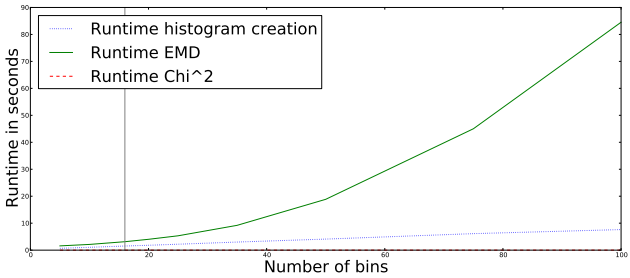


Figure 5: Practical Runtimes, evaluated for the *Car-Lateral* data set. The choice of $n_{\text{bins}} = 16$, obtained by the Sturges rule, is marked by the gray vertical line.

justice to the underlying structure: small intervals emphasize minor changes too much; large intervals obscure the structures entirely. In statistical applications, rules of thumb have been established for class intervals, depending on the number of observed data points. The displacements calculated in the car crash show structures similar to those of several normal distributions: As the nodes are all connected, a large displacement for one node influences all neighboring nodes as well. Therefore we applied one of these rules here as well: Due to its simplicity and the similar results of all rules for small data sets, we initially chose n_{bins} according to the Sturges-rule [Stu26]:

$$n_{\text{bins}} = \lceil \log_2(m) + 1 \rceil.$$

Our experiments showed, that for all data sets the Sturges rule provided very good results. Less classes obscured some of the information, as to be seen in Figure 3e with a choice of $n_{\text{bins}} = 5$ for the *Truck-Beam* data set, where the bifurcation is no longer obvious and some misplacements appear compared to the choice of $n_{\text{bins}} = 12$ proposed by the Sturges-rule. For more classes on the other hand, general structures and tendencies are to be seen for all tested bin numbers, but here as well the bifurcation becomes less obvious, cf. Figure 3f for $n_{\text{bins}} = 20$, where three agglomerations can be seen instead of two. The *Car-Lateral* data

set was even more stable for variations of n_{bins}, as to be seen in Figure 4c: even for only 5 bins the embedding looked very similar to the Sturges-rule based choice of $n_{\text{bins}} = 16$; analogously the shape of the point cloud does not change for very small bin sizes such as $n_{\text{bins}} = 100$. The negative effects of too large choices for n_{bins} are (as expected) boosted by the application of bin-to-bin-distances, such as χ^2, but still surprisingly good, especially considering the low runtime [Sch13].

The number of bins yields the initial configuration for the distance computations and thus, combined with the choice of a distance function, determines the algorithm's runtime. Here lies a large advantage of the χ^2-distance, as it runs linearly in $O(n_{\text{bins}})$, whereas EMD is usually implemented with Orlin's algorithm [Kor08], resulting in a runtime of $O({n_{\text{bins}}}^3 \log n_{\text{bins}})$. A practical evaluation is plotted in Figure 5. For all class sizes considered, calculation of the χ^2-distance took less than 0.01 seconds.

5 Conclusion

Our presented approach for the analysis of car crash simulation data proved successful for a real-world example with varying geometries. In a preprocessing step we represented the displacement data as normalized histograms and then used pairwise histogram distances to calculate a lower dimensional embedding using Diffusion Maps. The industrial data was arranged in a meaningful manner with visible correlations between the coordinate axes and the deformation behavior, which should simplify further analysis by development engineers.

Applying the method to a data set with fixed grid structures we could compare it to classical diffusion maps based on the Euclidean distances of the displacement vectors. This led to the surprising result that for our (simple) data set the distinguishing quality of the embedding was even improved by the new approach. Especially the EMD histogram distance proved stable against parameter variations. The χ^2-distance shows comparably good embedding results at much better runtimes, but with a higher sensitivity to parameter changes.

For the histogram representation in contrast to a displacement vector representation all information about positions and directions of the deformations are discarded. For the data investigated here, this did not prove problematic, but might be for more complex data sets. In general though the bundles of simulation runs generated in the car development process are all considered under the same loading conditions, so that positions of deformations should not vary. Still, to overcome these potential problems for more complex simulation runs one could further investigate the use of multidimensional histograms, e.g. 3-dimensional histograms to store displacement information on all coordinate directions instead of absolute displacements.

Acknowledgements

This paper is based on the author's bachelor's thesis [Sch13] and was written at Fraunhofer Institute for Scientific Computing and Algorithms SCAI. The author wishes to thank Prof. Dr. Jochen Garcke (Fraunhofer SCAI and Institute for Numerical Simulation, University of Bonn) and Rodrigo Iza-Teran (Fraunhofer SCAI) for their support and encouragement.

References

[Boh13] Bohn, B., Garcke, J., Iza Teran, R., Paprotny, A., Peherstorfer, B., Schepsmeier, U. and Thole, C. A. Analysis of Car Crash Simulation Data with Nonlinear Machine Learning Methods. *Procedia Computer Science*, 18:621 – 630, 2013. 2013 International Conference on Computational Science.

[Coi06] Coifman, R. and Lafon, S. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[Erb07] Erban, R., Frewen, T., Wang, X., Elston, T., Coifman, R., Nadler, B. and Kevrekidis, I. Variable-free exploration of stochastic models: a gene regulatory network example. *Journal of Chemical Physics*, 126(15):155103, 2007.

[Iza14] Iza Teran, R. Enabling the Analysis of Finite Element Simulation Bundles. *International Journal for Uncertainty Quantification*, 4(2):95–110, 2014.

[Kor08] Korte, B. and Vygen, J. *Kombinatorische Optimierung - Theorie und Algorithmen*. Springer-Verlag, Berlin, 2008.

[Laf04] Lafon, S. *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, 2004.

[Lin06] Ling, H. and Okada, K. Diffusion Distance for Histogram Comparison. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 246–253, 2006.

[Lux07] Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[Nad08] Nadler, B., Lafon, S., Coifman, R. and Kevrekidis, I. Diffusion maps-a probabilistic interpretation for spectral embedding and clustering algorithms. In *Principal manifolds for data visualization and dimension reduction*, pages 238–260. Springer, 2008.

[Pel10] Pele, O. and Werman, M. The Quadratic-Chi Histogram Distance Family. In *Computer Vision â ECCV 2010*, volume 6312 of *Lecture Notes in Computer Science*, pages 749–762. Springer Berlin Heidelberg, 2010.

[Rub00] Rubner, Y., Tomasi, C. and Guibas, L.J. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000.

[Sch13] Schwartz, A.-L. Diffusion Maps und ihre Anwendung bei der Analyse von Automobildaten. Bachelor's thesis, University of Bonn, 2013.

[Stu26] Sturges, H. A. The choice of a class interval. *American Statistical Association*, 21:65–66, 1926.