DOI: 10.1111/cgf.12878
Eurographics Conference on Visualization (EuroVis) 2016
K-L. Ma, G. Santucci, and J. J. van Wijk
(Guest Editors)

Volume 35 (2016), Number 3

Hierarchical Stochastic Neighbor Embedding

N. Pezzotti[1], T. Höllt[1], B. Lelieveldt[1,2], E. Eisemann[1], and A. Vilanova[1]

[1]TU Delft, The Netherlands
[2]Leiden University Medical Center, Leiden, The Netherlands

Abstract

In recent years, dimensionality-reduction techniques have been developed and are widely used for hypothesis generation in Exploratory Data Analysis. However, these techniques are confronted with overcoming the trade-off between computation time and the quality of the provided dimensionality reduction. In this work, we address this limitation, by introducing Hierarchical Stochastic Neighbor Embedding (Hierarchical-SNE). Using a hierarchical representation of the data, we incorporate the well-known mantra of Overview-First, Details-On-Demand in non-linear dimensionality reduction. First, the analysis shows an embedding, that reveals only the dominant structures in the data (Overview). Then, by selecting structures that are visible in the overview, the user can filter the data and drill down in the hierarchy. While the user descends into the hierarchy, detailed visualizations of the high-dimensional structures will lead to new insights. In this paper, we explain how Hierarchical-SNE scales to the analysis of big datasets. In addition, we show its application potential in the visualization of Deep-Learning architectures and the analysis of hyperspectral images.

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: General

1. Introduction

In *Exploratory Data Analysis*, a number of visualization techniques are used to support the hypothesis-generation process. Among its goals are the extraction of important variables, the detection of outliers or the identification of underlying non-convex structures [Tuk62]. *Non-linear dimensionality reduction* techniques play a key role in the understanding of high-dimensional data [BSIM14, SMT13]. A simple example is presented in Figure 1a, where a non-convex 1D manifold structure is defined in a 2D space. Non-linear dimensionality reduction is used to generate a 1D embedding (Figure 1b). Note that a linear transformation cannot project the manifold on such a 1D space. In the last decade, the application of non-linear dimensionality reduction techniques on real-world data led to new findings as complex real-world phenomena lead to non-convex structures that resides in a high-dimensional space [ADT*13, BSC*14]. Algorithms such as Sammon Mapping [Sam69], LLE [RS00], ISOMAP [TDSL00] or tSNE [vdMH08] help during Exploratory Data Analysis by giving a meaningful representation of these high-dimensional spaces. Broadly, two different approaches have been developed by the Machine-Learning and the Visualization community. The Machine-Learning approach tends to focus on accurate but computationally-expensive techniques, whereas the Visualization approach often trades accuracy and non-convex structure preservation for interactivity. Consequently, the first type is often too slow for interactive interfaces, limiting the ability to support the hypothesis-generation process. The second type is less accurate and can generate non-existing structures. For example, *hybrid* approaches use a set of *landmarks*, also called *pivots* or *control points*, which are embedded using non-linear dimensionality-reduction techniques. The remaining points are placed by interpolating their positions. Due to the sparse amount of landmarks, this process may not reflect the underlying manifold. An example is given in Figure 1c. The landmarks are placed in the wrong order according to the manifold, if the rest of the data is not taken into account. This problem can be partly remedied by letting the user manipulate the landmark positions in the embedding. However, this interaction cannot avoid the creation of non-existing structures and requires prior knowledge of the user about the data, which is usually not available.

Our Hierarchical Stochastic Neighbor Embedding algorithm (HSNE) is a non-linear dimensionality reduction technique that aims at bridging the gap between accuracy and interactivity. It is motivated by the good results that SNE techniques show in user studies [SMT13] and is as fast as the state-of-the-art *hybrid* techniques. While our approach also involves landmarks, it differs significantly from previous work. Our landmarks are enriched by a smooth and non-convex *area of influence* on the data and the landmarks are chosen meaningfully by analyzing the data points and their k-nearest neighbor graph, while avoiding outliers. Overlaps in the *areas of influence* are used to encode similarities between landmarks. Our process is hierarchical and landmarks at a higher scale are always a subset of the previous scale. This hierarchy allows us to keep the memory footprint small, while enabling a new way of analyzing the data. We follow the *Overview-first*, *Details-*

Figure 1: **Dimensionality reduction with landmarks**. In non-linear embedding techniques the underlying manifold (a) is respected (b). In *hybrid* approaches, landmarks are placed without considering the underlying manifold (c) and data points are placed by interpolating the landmark positions (grey line in c). The layout quality thus relates to the used number of landmarks.

on-Demand paradigm [Shn96] for the analysis of non-linear embeddings. Dominant structures that appear in the *Overview* can be analyzed by generating an embedding of the related landmarks in the subsequent lower scale. In this way, the user can drill down in the data and search for structures at finer scales. It is an approach that scales very well to big datasets and we illustrate its application potential in two different use cases.

The remainder of the paper is structured as follows. After an overview of the related work, Section 3 presents the HSNE algorithm with a focus on the construction of the hierarchy, while the hierarchical analysis is presented in Section 4. Finally, Section 5 contains two use cases showing the potential of our method, while experiments on well known datasets are presented in Section 6.

2. Related Work

Linear dimensionality-reduction techniques try to preserve global distances between data points in the embedding as in the high-dimensional space. Hierarchical implementations of these techniques have been developed to reduce calculations. Notable examples are Glimmer [IMO09], Steerable MDS [WM04] and HiPP [PM08] that linearly separate the space with a top-down approach. Differently from linear algorithms, non-linear dimensionality reduction techniques try to preserve geodesic distances on manifolds between data points. However, a simple case as in Figure 1a is rarely met in practice, and the definition of geodesic distances is a challenging task. In real-world data, data points form manifolds defined by sets of points varying in size, density, shape and intrinsic dimensionality. A class of techniques known as Stochastic Neighbor Embedding (SNE) [HR02] is accepted as the state of the art for non-linear dimensionality reduction for the exploratory analysis of high-dimensional data. Intuitively, SNE techniques encode small-neighborhood relationships in the high-dimensional space and in the embedding as probability distributions. These techniques aim at preserving neighborhoods of small size for each data point. The embeddings are defined via an iterative minimization of the loss of information when placing the point in the embedding. Besides the discoveries made using algorithms like tSNE [ADT*13, BSC*14], the ability to reveal interesting structures is demonstrated by extensive user studies on real-world and synthetic data [SMT13]. Unfortunately, the application of SNE techniques to large datasets is problematic, as the computational complexity is usually $O(n^2)$. Using approximations it can be reduced to $O(n \log(n))$ [PL*15, VDM14]. Furthermore, small-neighborhood preservation might miss structures at different sizes. Our HSNE is an SNE technique, which overcomes the computational complexity and shows structures at different scales by creating a hierarchical representation of the dataset. Differently from other hierarchical techniques [IMO09, WM04, PM08], we use a bottom-up approach in the creation of the hierarchy. Our key insight is to use landmarks that represent increasingly large portion of the data.

The usage of landmarks is not new and can be separated in two categories, which we refer to as the *non-linear* and *hybrid* landmark techniques (see Figure 1). Both select a set of landmarks from the original dataset. *Non-linear* landmark techniques embed them using metrics that estimate geodesic distances between points [ST02, vdMH08]. Figure 1b shows a simple example, where the neigborhood relationship are extracted using the geodesic distances on the manifold. For example, Landmark-tSNE creates the K-Nearest Neighbor (KNN) Graph between the original data point and computes for each landmark the probability of reaching other landmarks with a random-walk on the KNN-Graph [vdMH08]. Non-linear landmark techniques can discover non-convex structures, but their scale is directly related to the number of selected landmarks. Further, the user is limited to the visualization of landmarks and not the complete dataset, limiting the insights that can be extracted from the data. *Hybrid* landmark techniques embed landmarks with non-linear dimensionality reduction techniques based on high-dimensional descriptors of the landmarks derived from the original data. The complete dataset is then embedded using different interpolation schemes [FFDP15, JPC*11, PNML08, PSN10, PdRDK99, dST04, PEP*11]. This approach is widely used by the visualization community due to its fast computation, making it ideal for interactive systems. However, non-convex structures are not preserved (unless the sampling is dense enough) because the underlying manifold is ignored. Figure 1c illustrates the problem: the selected landmarks are seen as a straight line even by a non-linear technique.

HSNE is a *non-linear* landmark technique, but supports the exploration of non-convex structures at different scales, while sharing the performance of *hybrid* techniques and supporting interaction to gain insights into the data. In particular, our novel hierarchical approach using an *Overview-first*, *Details-on-Demand* paradigm helps in this context.

3. Hierarchical Stochastic Neighbor Embedding

Here, we present our HSNE technique with a focus on the creation of the hierarchical data representation. An overview is given in Figure 2. Throughout the paper, calligraphic notations indicate sets, for example, $\mathcal{D}$ is the set of high-dimensional data points. Our representation is composed of different scales, or levels, organized hierarchically. We use superscripts to indicate this scale. Elements in sets are identified using subscripts. We denote $\mathcal{L}^s$ the set of landmarks representing the dataset at scale s. $\mathcal{L}^1$ represents the first scale, which is the input dataset $\mathcal{D}$. Higher scales are always subsets of previous scales ($\mathcal{L}^s \subset \mathcal{L}^{s-1}$).

Our algorithm works as follows. Starting with $\mathcal{L}^1$, we build a Finite Markov Chain (FMC) from a k-nearest-neighbor graph to encode similarities between landmarks (Section 3.1). It is used to guide the selection process of a landmark subset for the next scale

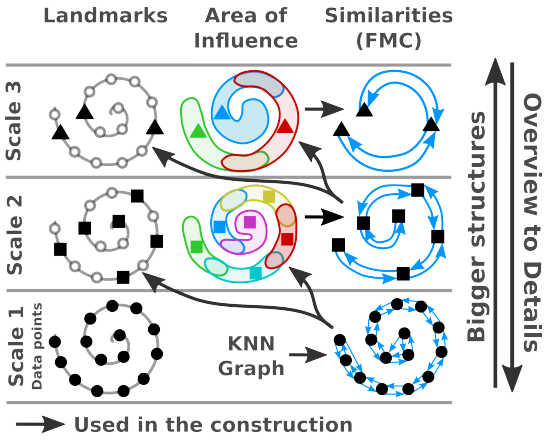

Figure 2: **Overview of the hierarchy construction**. A Finite Markov Chain (FMC) is built from the k-nearest neighbor graph. The FMC encodes the similarities between landmarks and it is used for selecting landmarks in the next scale. The FMC is also used to compute the *area of influence* of the selected landmarks on the landmarks in the lower scale. The overlap between the *areas of influence* is used to build a new FMC that encodes similarities in the new scale.

(Section 3.2) and, then, to compute an *area of influence* for each selected landmark (Section 3.3). The overlap between these areas indicates similarity and forms the basis for a new FMC encoding (Section 3.4), which is then used to compute the next scale. After preprocessing the different scales, we can perform a multi-scale analysis by computing an embedding of landmarks using their scale-dependent information (Section 3.5).

3.1. From data points to a Finite Markov Chain

A Finite Markov Chain is a random process that undergoes transitions from one state to another in a state space. Our Finite Markov Chain is used to model the random movement of a hypothetical particle on the manifold, and the states are given by the landmarks in $\mathcal{L}^s$. The transitions are encoded in a square *transition matrix* T^s of size $|\mathcal{L}^s| \times |\mathcal{L}^s|$. $T^s(i,j)$ represents the probability that the landmark $\mathcal{L}^s_j$ belongs to the neighborhood of the landmark $\mathcal{L}^s_i$ in the scale s. It is important to note that HSNE aims at encoding small neighborhoods of fixed size for every landmark. Therefore T^s is sparse by construction, and its memory complexity grows linearly with the size of the dataset.

For the Finite Markov Chain described by the transition matrix T^1, each data point $\mathcal{D}_i$ is only allowed to transition to a data point $\mathcal{D}_j$, if $\mathcal{D}_j$ belongs to the k-nearest-neighborhood $\mathcal{N}(i)$ of $\mathcal{D}_i$. The probability assigned to the transition is given by the following equation:

$$T^1(i,j) = \frac{\exp(d(i,j)^2/\sigma_i)}{\sum_k \exp(d(i,k)^2/\sigma_i)} \text{ with } j,k \in \mathcal{N}(i), \qquad (1)$$

where $d(i,j)$ are the Euclidean distances between data points, and σ_i is chosen such that $T^1(i,-)$ has perplexity of $|\mathcal{N}(i)|/3$ [VDM14]. The exponential falloff is used to reduce the problem caused by the presence of outliers, that act as *shortcuts* across manifolds. SNE techniques focus on the preservation of small neighborhoods for each data point. Thus, a small value of K is usually selected, where 100 is a common choice [vdMH08, VDM14]. To improve performance, we adopt the approximated algorithm for the computation of the k-nearest-neighborhoods proposed by Pezzotti et al. [PL*15]. Experimentally, we see that such an algorithm does not compromise the quality of the embeddings generated by HSNE while improving the computation time by two orders of magnitude. We refer the interested reader to the work of Pezzotti et al. [PL*15] for details. The computational complexity of this first step is $O(|\mathcal{D}|\log(|\mathcal{D}|))$

3.2. Landmark selection and outliers identification

We use the transition matrix to carefully select meaningful landmarks in order to reduce the size of the dataset. This step is of crucial importance, e.g., in order to avoid choosing outliers as landmarks. So far, we have only given the definition of the transition matrix for the lowest scale. We define it for other scales in Section 3.4. Nonetheless, the process described here is valid at all scales, which is why we use the superscript s to indicate its generality. Before we explain our sampling solution, we introduce the concept of *equilibrium distribution* of a Finite Markov Chain. A vector π is called *equilibrium distribution* of the Finite Markov Chain, described by the transition matrix T^s, if it represents a probability distribution that is not changed by a transition in the state space:

$$\pi = \pi T^s \text{ and } \sum_i \pi_i = 1 \qquad (2)$$

Intuitively, the *equilibrium distribution* π represents the probability of being in a state after an infinite number of transitions in the state space. These transitions are often called *random walks* in the state space. Given the transition probabilities defined by Equation 1, the *equilibrium distribution* of our Finite Markov Chain assigns higher probability to data points that reside in high-density regions in the original space. Figure 3 shows an example, where the landmarks $\mathcal{L}^s$ are color coded according to the *equilibrium distribution* of the Finite Markov Chain that encodes their similarities. Landmarks in dense regions of the space, have high value of π and are selected to be in $\mathcal{L}^{s+1}$ (green circles in Figure 3). Landmarks with a low value of π are considered outliers in scale $s+1$ (blue circles in Figure 3).

Landmarks in $\mathcal{L}^{s+1}$ are selected by sampling the *equilibrium distribution* π, that is computed using a simple Markov Chain Monte Carlo technique [Gey11]. For each landmark in $\mathcal{L}^s$, we start β random walks of fixed length θ. Every landmark that is the endpoint of at least $\beta_{\text{treshold}} * \beta$ random walks is selected as a landmark in $\mathcal{L}^{s+1}$, if no random walks reach a given landmark, it is detected as outlier. We experimented with different values of β and θ, finding that $\beta = 100$ and $\theta = 50$ is a good compromise between speed and accuracy for the data we have been analyzing. Notice that the computation of random walks is not costly, and thousands can be performed every millisecond on a state-of-the-art desktop computer. We provide a default value of $\beta_{\text{treshold}} = 1.5$, that we found is conservative enough to create a hierarchical representation for all the dataset that we tested. The computation complexity of this step is $O(|\mathcal{L}^s|)$.

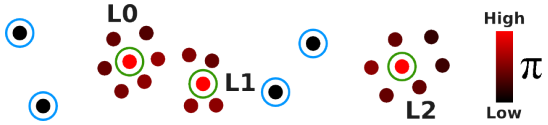

Figure 3: **Selection of landmarks and outliers** using the *equilibrium distribution* π of the Finite Markov Chain (see Equation 2). Points are color coded from black to red according to their π-value. Selected landmarks are circled in green, while potential outliers are circled in blue.

3.3. Area of influence

The process of choosing landmarks cannot be simply relaunched, as we would then loose important information from previous scales. In consequence, we will extend the definition of the transition matrix to all scales beyond the first. To this extent, we introduce the *area of influence* for each landmark, which keeps track of a landmark's impact on previous scales. The influence exercised by landmarks in $\mathcal{L}^s$ on those in $\mathcal{L}^{s-1}$, is encoded in an *influence matrix* I^s. Matrix I^s has size $|\mathcal{L}^{s-1}| \times |\mathcal{L}^s|$, where $I^s(i,j)$ is the probability that the landmark $\mathcal{L}_i^{s-1}$ in the previous scale is well represented by $\mathcal{L}_j^s$. Specifically, each row i is a probability distribution that denotes the probability that the landmark $\mathcal{L}_i^{s-1}$ is in the area of influence of landmarks in $\mathcal{L}^s$. Consequently, the influence of a scale s on scale r is defined by a chain of sparse matrix multiplications:

$$I^{r \leftarrow s} = \left[\prod_{i=s}^{r} \left(I^i \right)^t \right]^t \quad \text{with } r < s \tag{3}$$

It is important to note that the area of influence is localized, implying that I^s is sparse. Therefore, the memory complexity grows only linearly with the set of landmarks. To compute I^s, we start a number of random walks in the Finite Markov Chain described by T^{s-1} for each landmark $\mathcal{L}^{s-1}$, leading to a computational complexity of $O(|\mathcal{L}^{s-1}|)$. The random walk stops when a landmark in $\mathcal{L}^s$ is reached. The percentage of random walks reaching every landmarks in $\mathcal{L}^s$ is then used as a row for $I^s(i,-)$. Figure 4 shows the area of influence of the selected landmarks in Figure 3 as a flow, converging in landmarks of the higher scale. Depending on the data distribution in the high-dimensional space, landmarks can exercise influence on regions of different size. We define the *weight* of a landmark as the size of the region that it represents. The vector W^s encodes the weights of the landmarks at scale s, and it is defined by the following equation:

$$W^s = W^{s-1} * I^s \text{ with } W^1 = \mathbf{1} \tag{4}$$

The width of the landmarks in Figure 4 represents these weights W^s.

3.4. From areas of influence to Finite Markov Chains

Similarities between landmarks in scale s are computed using the overlaps in their *areas of influence* on scale $s-1$. Intuitively, if the areas of influence of two landmarks overlap, it means that they are close on the manifold, therefore their similarity is high. We use the influence matrix I^s to create the FMC, encoding similarities between landmarks in $\mathcal{L}^s$. The transition matrix T^s is given by the

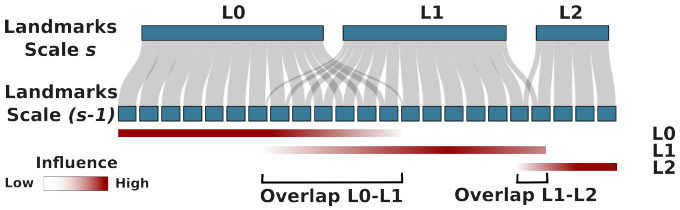

Figure 4: **The area of influence** can be seen as flow converging in landmarks of the higher scale. The area of influence of the landmarks selected in Figure 3 is shown here. The overlap in the area of influence is used to compute similarities between landmarks (see Equation 5).

following equation:

$$T^s(i,j) = \frac{\sum_{k=1}^{|\mathcal{L}^{s-1}|} I^s(k,i) I^s(k,j) W^{s-1}(k)}{\sum_{k=1}^{|\mathcal{L}^{s-1}|} \sum_{l=1}^{|\mathcal{L}^s|} I^s(k,i) I^s(k,l) W^{s-1}(k)} \tag{5}$$

where $I^s(k,i) I^s(k,j) W^{s-1}(k)$ is the overlap of the area of influence of $\mathcal{L}_j^s$ and $\mathcal{L}_i^s$ on landmark $\mathcal{L}_k^{s-1}$. Figure 4 depicts overlaps between the *areas of interest* of the landmarks selected in Figure 3. The overlap between L0 and L1 is higher than the overlap between L1 and L2, as expected because L1 is more similar to L0 than L2.

3.5. Generation of the embedding

SNE methods rely on a probability distribution P, that encodes neighborhood relationships. In practice, we rely on tSNE because of its ability to overcome the so called *crowding problem* [vdMH08]. tSNE interprets similarities between data points as a symmetric joint-probability distribution P. Likewise a joint-probability distribution Q is computed, that describes the similarities in the low-dimensional space. The goal is that the position of the data points in the embedding faithfully represent the similarities in the original space. The iterative minimization of the Kullback-Leibler divergence is used to reduce the information loss when Q is used to represent P. An in depth explanation on how tSNE computes Q and minimizes the divergence function is out of the scope of this work. We refer to van der Maaten et al. [vdMH08] for further details. In our HSNE, P is computed from the transition matrix T^s:

$$P(i,j) = \frac{T^s(i,j) + T^s(j,i)}{2|\mathcal{L}^s|} \text{ where } \sum_{i,j} P(i,j) = 1 \tag{6}$$

With this definition, an embedding can be computed even for a subset of the landmarks, the only requirement is that their similarities are encoded in a Finite Markov Chain. This observation is important as it enables the *Overview-First, Details-on-Demand* analysis presented in the next section. However, if the user is interested in generating a complete embedding (as in *hybrid* techniques), it can be achieved by interpolating the position of the landmarks in the top scale o:

$$\mathcal{Y}_i^1 = \sum_j^{\mathcal{L}^o} \mathcal{Y}_j^o I^{1 \leftarrow o}(i,j) \tag{7}$$

where $I^{1 \leftarrow o}(i,j)$ is the influence exercised on the data points, as shown in Equation 3.

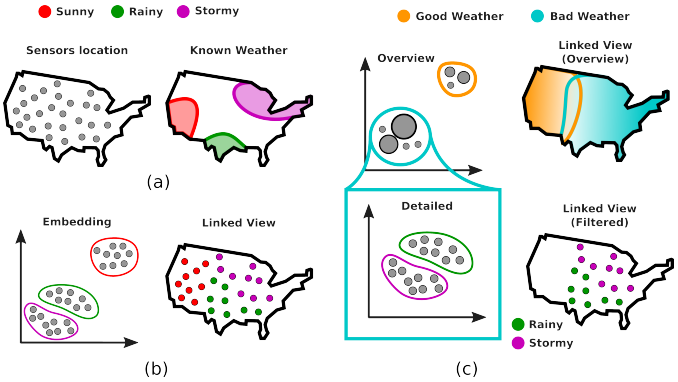

Figure 5: **Traditional vs Hierarchical analysis**. (a) High-dimensional readings from sensors located on a map and prior knowledge on the phenomenon of interest are available to the user. In the traditional analysis (b) a single embedding is generated and analyzed. In our hierarchical analysis (c), an overview shows dominant structures in the dataset. Detailed embedding of the structures are created by filtering and drilling into the data.

4. Hierarchical Analysis

In this section, we describe how the hierarchical analysis is performed by presenting how the detailed embeddings are generated by *filtering* and *drilling-down* in the data. Before addressing the algorithmic solution, we will motivate the usefulness of such a tool with an example.

4.1. Example of a hierarchical analysis

Standard dimensionality reduction techniques are often used to enable a user to visually identify groups of similar data points. This possibility is useful, as it enables tasks, such as *verification*, *naming* or *matching* [BSIM14]. Figure 5a, shows a simple *naming task* using readings from atmospheric sensors as high-dimensional data. Figure 5b shows another example, in the context of traditional analysis. In a naming task, the analysis of the given data might lead to a set of different clusters. A user could inspect these clusters by selecting one and seeing the corresponding region highlighted on the map. Using prior knowledge for a few locations, it becomes possible to attribute conditions, such as sunny, cloudy and rainy weather, on the entire map. Nonetheless, such an analysis assumes that the scale of the clustering was sufficiently precise and not overly precise.

The hierarchical nature of our approach enables a new multi-scale analysis based on the *Overview-First*, *Details-on-Demand* mantra [Shn96]. An example is given in Figure 5c. Instead of showing an embedding of all data points, the analysis starts with the highest-scale landmarks. The resulting clusters will represent very coarse dominant structures, for example, good and bad weather zones. Additionally, the area of influence encoded in the size of the embedded points gives feedback regarding the complexity of the original underlying data. If a user now wishes to explore more detailed information, a cluster can be selected and a lower scale embedding is produced. The heterogeneous data on the lower level then becomes visible, for example, bad weather transforms into cloudy and rainy regions. Our approach is particularly suited for heterogeneity at different scales, which is common in large datasets.

4.2. Filtering and drill down

To enable the investigation of details, we start from a selected subset $\mathcal{O}$ of landmarks at scale s: $\mathcal{O} \subset \mathcal{L}^s$. We drill in the data by selecting a subset $\mathcal{G}$ of landmarks at scale $s-1$: $\mathcal{G} \subset \mathcal{L}^{s-1}$, using the influence matrix I^s to connect the two scales. As explained in Section 3.3, a row i in I^s represents for $\mathcal{L}_i^{s-1}$ the influence of the landmarks $\mathcal{L}^s$ at scale s. We define F_i as the probability that landmark $\mathcal{L}_i^{s-1}$ is in the area of influence of the landmarks in $\mathcal{O}$:

$$F_i = \sum_{\mathcal{L}_j^s \in \mathcal{O}} I^s(i,j) \tag{8}$$

If all the landmarks influencing $\mathcal{L}_i^{s-1}$ are in $\mathcal{O}$, then $F_i = 1$. If no influence from $\mathcal{O}$ is exercised on $\mathcal{L}_i^{s-1}$ then $F_i = 0$. A landmark $\mathcal{L}_i^{s-1}$ is selected to be in $\mathcal{G}$ if $F_i > \gamma$, where γ is a user-defined threshold. However, it should be noted that a low value of γ is not desirable, as it will add landmarks, which are only slightly influenced by $\mathcal{O}$. A high value of γ is also not desirable, leading to the exclusion of regions that are highly influenced by $\mathcal{O}$. While it remains a parameter, we found experimentally that $\gamma = 0.5$ allows for effective drilling in the data. The transition matrix $T_{\mathcal{G}}^{s-1}$, representing the similarities in $\mathcal{G}$, is derived from T^{s-1} by removing the rows and columns of landmarks in L^{s-1}, which are absent from $\mathcal{G}$. Given the transition matrix, the embedding is computed as before (Section 3.5).

5. Use cases

Here, we show examples for our hierarchical analysis on real-world data, to illustrate our contributions and potential application areas of our HSNE. Besides high-resolution hyperspectral images of the Sun and remote-sensing data, we visualize the training set of a well-known Deep Learning model, showing how it interprets the input images. We demonstrate the HSNE's ability to show dominant structures in the *Overview* and to explore them in detailed embeddings to reveal finer-grained structures. We test our C++ implementation of HSNE on a DELL Precision T3600 workstation with a 6-core Intel Xeon E5 1650 CPU @ 3.2GHz and 32GB RAM.

5.1. Hyperspectral images

The visible light spectrum is only a tiny part of the electromagnetic spectrum and some phenomena can only be understood by considering the complete spectrum. Figure 6a, shows hyperspectral images of the sun. Different wavelengths of the electromagnetic spectrum reveal different observations, such as solar flares or the corona. The image resolution is 1024×1024, leading to a dataset composed of $\approx 1M$ data points (pixels). Each pixel is described by 12 dimensions corresponding to the intensity readings. We downloaded the data from the Solar Dynamics Observatory† on November 13th 2015. In an Exploratory Data Analysis the user needs to

† http://sdo.gsfc.nasa.gov/

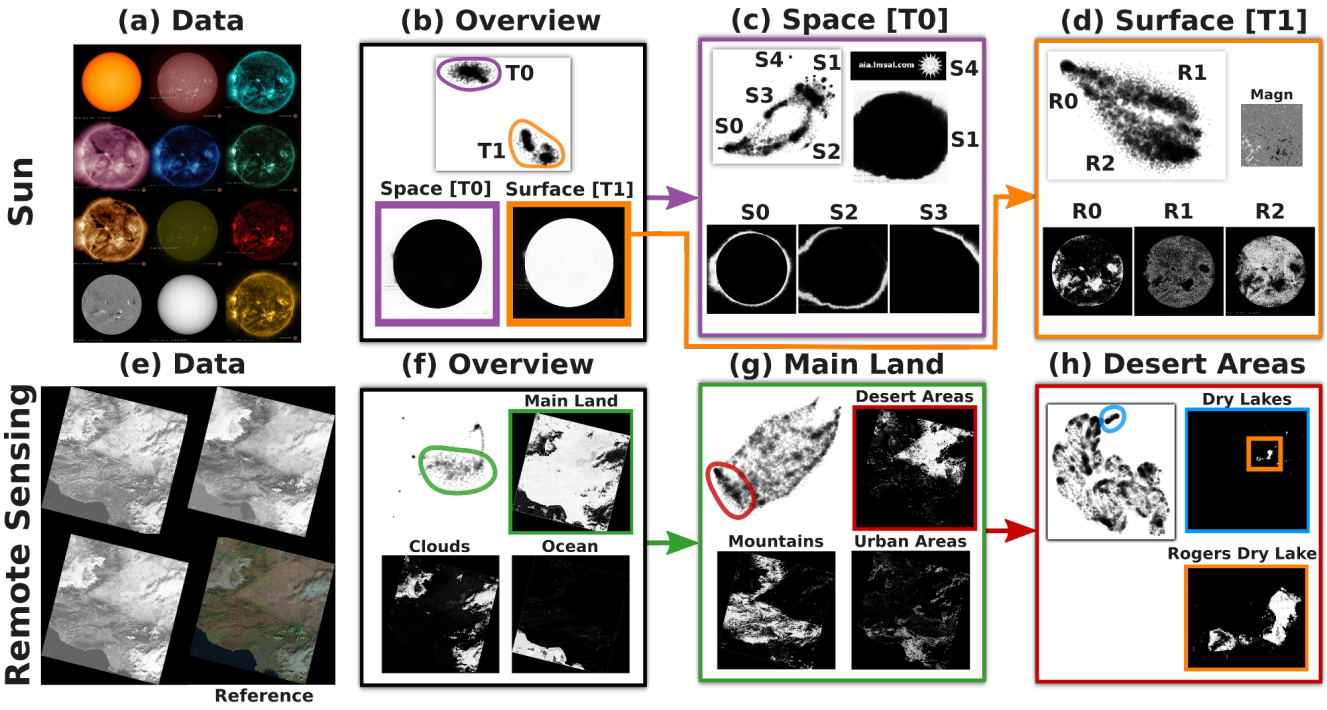

Figure 6: **Hierarchical analysis of hyperspectral images**. Hyperspectral images of the Sun and the area surronding the city of Los Angeles are analyzed using HSNE. Dominant structures are revealed at different scales and can further inspected by creating detailed embeddings.

analyze all pictures of all wavelengths in parallel. However, with an increasing number of images, the data complexity complicates the generation of a hypothesis or the differentiation of different regions. Here, we show how HSNE supports such analysis.

The hierarchical representation of the data is precomputed in 2'13" minutes and only needs to be processed once. From this representation the overview and detailed embeddings require only a few seconds and can be visualized using a Progressive Visual Analytics approach [PL*15]. Figure 6b shows the *Overview* generated by HSNE. The *Overview* is composed of 352 landmarks in two clusters (T0 and T1). Every landmark is drawn using semi-transparent circles, while the size of a landmark encodes its weight as defined in Equation 4. The clusters correspond to two dominant structures in the data, the Sun surface (T1) and the Space in the background (T0). Their *areas of influence* is visualized in the linked view. Here, an image of size 1024×1024, where a greyscale colormap is used to represent the probability of a pixel to belong to the *area of influence* of the selection. The user drills in the data by requesting detailed visualizations of the two dominant structures. A detailed embedding of T0 (Figure 6c) describes different regions of the Corona. S0 represents the area close to the surface, while S1 represents the background. S2 and S3 represent the external area of the Corona, where S3 is an area with low readings in the AIA 211 channel (pink in Figure 6a). S4 is an interesting cluster, representing the overlayed logo, present in all images. S4 is considered an outlier in the overview and, therefore, was not represented as a separate cluster. However, upon refinement, this cluster would appear, as it will be a dominant structure at this scale. A detailed embedding of T1 leads to three clusters (Figure 6c). Although not as well separated, they clearly represent different regions on the Sun surface. R0 are hotter regions, or where solar flares are visible, while R1 and R2 represent colder regions separated in one of the input images, namely the *Magn* image (Figure 6d).

We performed a similar analysis on hyperspectral images for remote sensing. These data are captured by the LandSat satellites [‡], and we present an example of the area surrounding the city of Los Angeles. The data are composed of 11 images, representing different bands of the electromagnetic spectrum. Figure 6e shows three of such images, and a reference image. Similarly to the previous example, we analyzed the images at a resolution of 1024×1024. Figure 6f shows the dominant structures in the highest scale, namely *ocean*, *clouds* and the *main land*, that are identified by the user by looking at the reference image and using its prior knowledge on the phenomenon. A detailed embedding representing the *main land* is shown in Figure 6g. It is possible to identify different parts of the detailed embeddings related to *mountains*, *urban* and *desert areas*. Drilling in, detailed embeddings are generated, such as the one representing *desert areas*, depicted in Figure 6h. More heterogeneity is revealed at this scale. For instance dry lakes, such as the *Rogers Dry Lake*, are located in the cluster of desert areas.

‡ http://landsat.usgs.gov/

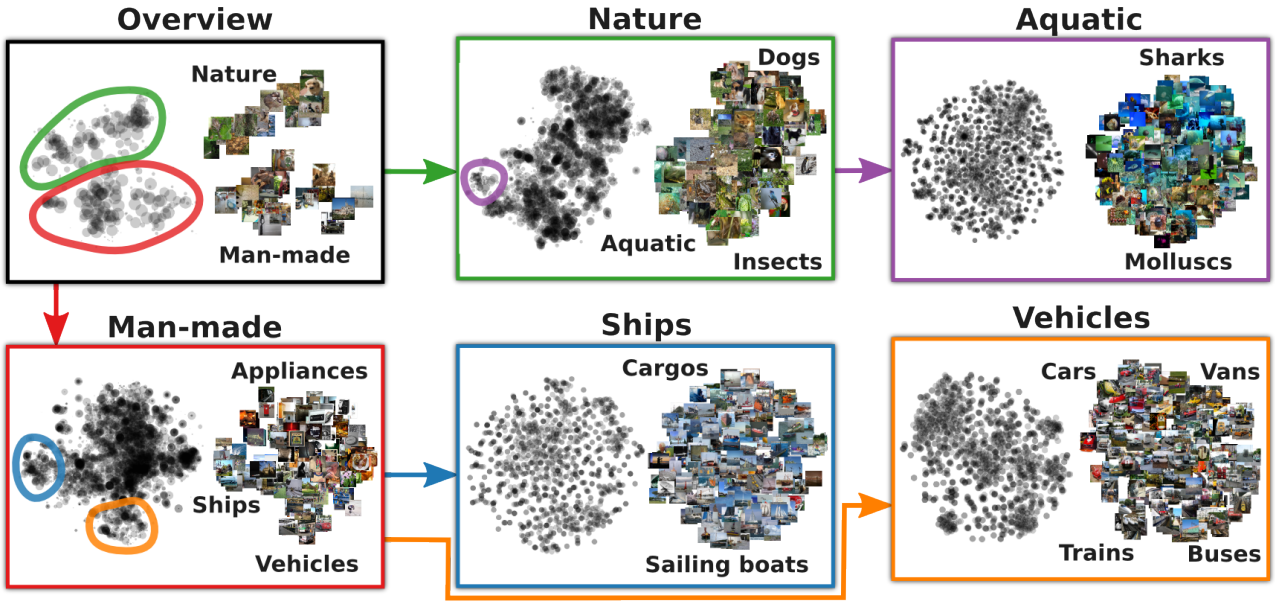

Figure 7: **Deep Learning models**. Features are extracted from 100k images using a Deep Neural Network (DNN) [KSH12] and the hierarchical analysis is performed using HSNE. Starting from the overview, dominant structures at different levels are revealed. The user can inspect the embeddings and request detailed visualization. This is achieved through filtering of the landmarks and by drilling down in the hierarchy. A high-resolution version of the figure is provided in the supplemental materials.

5.2. Visualization of Deep Learning datasets

Deep Learning builds upon neural networks composed of many (hence, the name *deep*) layers. Deep Neural Networks (DNN) achieved impressive results in image recognition, sentiment analysis and language translation. For an overview of the field, we refer to [LBH15]. However, it is difficult to visualize how a DNN works. An approach that was used recently, is to select some potential inputs that are processes by the DNN [JvdMJV15, MKS*15]. For each input, the values computed by the last layer of the network are used as high-dimensional descriptor. Once that the descriptor are assigned to each data point, they are embedded in a 2D space using non-linear dimensionality reduction techniques. tSNE is usually selected for such a task [JvdMJV15, MKS*15]. The limitation of this approach is that only small subsets can be visualized at a given time, limiting the ability to evaluate and inspect how the network is trained. We extract features from the test set of a well known DNN [KSH12], leading to a dataset consisting of 100k images and 4096 dimensions. The hierarchical representation of the data is computed in 92 seconds, while every embedding requires only few seconds to be computed. Our approach shows the hierarchical nature of the learning process, as depicted in Figure 7. In the overview two clusters are visible. We label them as *Man-made* and *Nature*, based on the inspection of the images represented by the landmarks. Detailed embeddings of the clusters are produced and confirm the previous labeling. In the *Nature* cluster new dominant structures are revealed, such as images of *Aquatic* animals, *Insects* or *Dogs*. Similarly, a detailed visualization of the landmarks labeled as *Man-made* reveal more heterogeneity in the data. The user can drill deeper in the data, for example by requesting detailed visualization of landmarks identified as *Ships*, *Vehicles* and *Acquatic* animals.

6. Evaluation

In this section we provide experimental evidence that HSNE outperforms *hybrid* and *non-linear* dimensionality reduction techniques. In our evaluation, we use the MNIST dataset [§] (60k points, 784 dimensions), the CIFAR-10 dataset [¶] (50k points, 1024 dimensions) and the TIMIT dataset [‖] (1M points, 39 dimensions). Figure 8 shows the embeddings of the MNIST dataset produced with our approach compared to those created by *non-linear* techniques (tSNE and L-SNE [vdMH08]) and *hybrid* techniques (LSP [PNML08], Piecewise-LSP [PEP*11], created by the Projection Explorer tool [POM07], and LAMP [JPC*11] created by the Projection Analyzer tool [**]). Our HSNE embedding is computed for three scales, resulting in the highest-level embedding containing 1431 landmarks. The tSNE embedding is computed using approximated computations [PL*15, VDM14] to reduce the computational complexity to $O(n \log n)$. For the L-SNE algorithm, we randomly selected 1431 landmarks and we use approximated k-nearest-neighbor computations (see Section 3.1), making it comparable to the setting for the HSNE. We were not able to generate a LSP embedding of the MNIST dataset due to its size and present

§ http://yann.lecun.com/exdb/mnist/
¶ https://www.cs.toronto.edu/ kriz/cifar.html
‖ https://catalog.ldc.upenn.edu/LDC93S1
** https://code.google.com/archive/p/projection-analyzer/

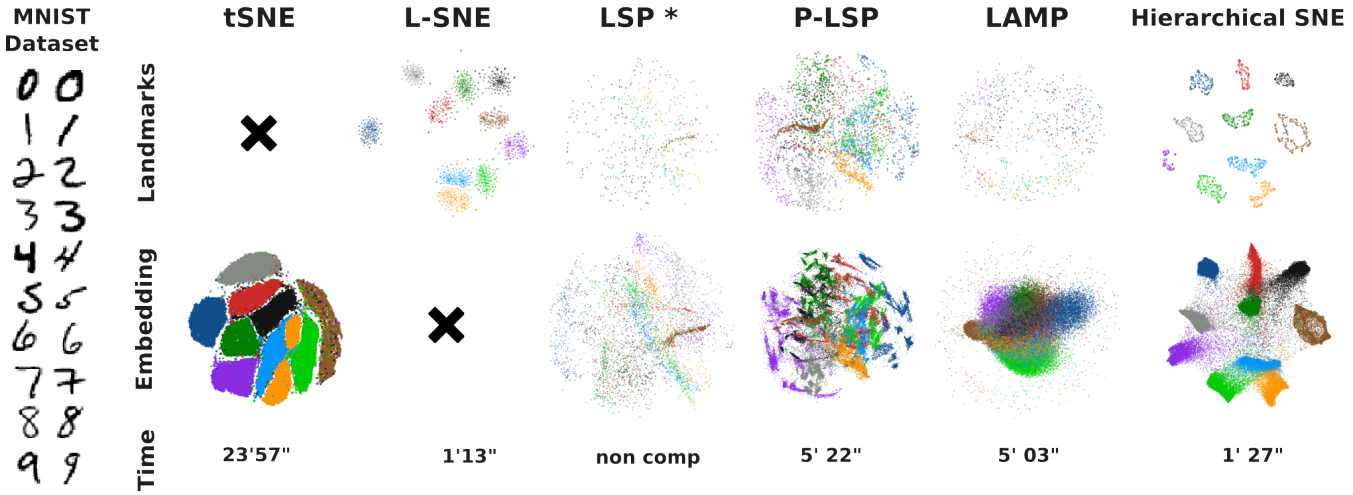

Figure 8: **Embeddings of the MNIST dataset** created by non-linear dimensionality reduction techniques (tSNE and Landmark-SNE) and by *hybrid* techniques (LSP, P-LSP and LAMP). Differently from *hybrid* techniques, HSNE preserves the manifold in the landmark embedding, creating compact clusters in the complete embedding.

an embedding of 5k randomly selected data points instead. We use the default parameters for the selection of the landmarks, leading to 500 landmarks in LSP, 3714 in P-LSP and 734 in LAMP. For each technique we present, where available, the embedding containing only the landmarks, as well as the complete embedding. Our HSNE is much faster than tSNE and comparable to *hybrid* techniques.

We base our quantitative assessement of the quality of the embedding on the *Nearest-Neighbor Preservation* metric (NNP) as proposed by Venna et al. [VPN*10] and implemented by Ingram and Munzner [IM15]. For each data point, the K-Nearest-Neighborhood (KNN) in the high-dimensional space is compared with the KNN in the embedding. Average precision/recall curves are generated by taking into account high-dimensional neighborhoods of size $K_{high} = 30$ [IM15]. The precision/recall curves are computed by selecting K_{emb}-neighborhoods in the embedding, iterating K_{emb} from 1 to K_{high} and computing the true positive TP in the K_{emb}-neighborhood. The precision is set as TP/K_{emb} and the recall as TP/K_{high}. The curve is obtained by connecting the points in the precision/recall space for each value of K_{emb} [IM15]. However, NNP fails to measure the preservation of high-level information, e.g. neighborhood preservation in a geodesic sense and, to the best of our knowledge, no such metric exists. Therefore, we assess the high-level structure preservation both by a visual inspection of the labeled data and by the evaluation of the NNP during the drill-down in the data. Intuitively, if HSNE does not have the ability to preserve high-level structures, during a drill-down part of the data will be left out, leading to gaps in the lowest-level embedding and, consequently, to bad NNP.

Even if a validation of the visual cluster cannot be performed, given its non-convex nature [Aup14], the MNIST dataset contains compact manifolds [vdMH08] that represent handwritten digits (see examples in Figure 8). Therefore, based on the visual separation of the labeled landmarks, we can conclude that HSNE preserves manifolds similar to non-linear dimensionality-reduction algorithms. Hybrid techniques are incapable of well separating the manifolds in this example. Due to the fact that the underlying manifold is not respected, the landmark positions in the embedding ignores local structures in the data, leading to problems similar to the one depicted in Figure 1c. HSNE separates the manifolds even better than tSNE, see orange cluster in the tSNE embedding compared to orange landmarks in the HSNE embedding. This result is a consequence of tSNE focusing only on the preservation of small neighborhoods. When the size of the data increases, we experimentally found that minimization performed by tSNE will often incur in local minima that disrupt the visual representation of high-level manifolds.

tSNE's ability to preserve small neighborhoods is confirmed by the NNP precision/recall curves presented in Figure 9a. For HSNE we compute a precision/recall curve for each scale by linearly interpolating the data points using landmarks in the corresponding scale, as in Equation 7. In the highest scale, HSNE outperforms the other *hybrid* techniques but it performs worse than tSNE. This is expected as the information preserved by HSNE at this scale is not measured by NNP. When the lowest scale is considered, the precision/recall curve of HSNE and tSNE are similar. However, HSNE is designed to filter the data during the hierarchical analysis. Figure 9b shows the analysis performed by selecting landmarks that belong to the digit '7' (green points in Figure 8) and computing the the precision/recall curves using the points selected to be in the lowest scale. HSNE outperforms tSNE in the lowest scale: by reducing the number of data points to embed, HSNE is less influenced by local minima during their placement, leading to a better NNP. This result also confirms that in the higher scales of the hierarchy, manifolds are consistently represented, avoiding the creation of gaps in the lowest level embedding during the analysis. We obtained similar results for different analysis performed on the three datasets.

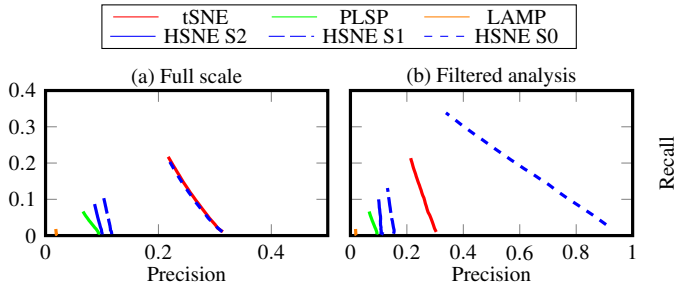

Figure 9: **Nearest Neighborhood Preservation (NNP)** on the MNIST dataset. HSNE outperforms hybrid techniques and it is comparable to tSNE on a full scale analysis. When the user filters the data during the drill-in, HSNE outperforms tSNE.

7. Conclusions

We presented Hierachical Stochastic Neighbor Embedding (HSNE). HSNE introduces the well-known mantra *Overview-First, Details-on-Demand* in non-linear dimensionality-reduction techniques. Our technique preserves non-convex structures, similarly or better than the state-of-the-art methods, but can be employed in interactive software for the Exploratory Analysis of high-dimensional data. Even though complete embeddings (similar to *hybrid* techniques) are possible, a key strength is the interactive hierarchical analysis to reveal dominant structures at different scales, which is useful in various applications, as evidenced by our use cases.

The various results indicate that HSNE is a beneficial replacement for non-linear and hybrid algorithms in Visual Analytics solutions. The use of the area of influence, is an important visualization element and delivers additional information, although new strategies would have to be developed to effectively exploit it. Nonetheless, this aspect is important when considering systems to assess the quality of embeddings [MCMT14]. These mainly focus on visualizing and inspecting missing and false neighborhood relationships between data points. In the future, we want to investigate this neighborhood encoding further and explore how it can help users in assessing the quality of the embedding at different scales. We also consider applying uncertainty visualization techniques to illustrate the selected landmarks in linked views, as the area of influence is directly expressed in terms of probabilities. The multi-scale nature of many real-world phenomena leads us to the conclusion that HSNE may give new insights into a number of problem domains.

Acknowledgements. This work received funding through the STW Project 12720, VAnPIRe.

References

[ADT*13] AMIR E.-A. D., DAVIS K. L., TADMOR M. D., SIMONDS E. F., LEVINE J. H., BENDALL S. C., SHENFELD D. K., KRISHNASWAMY S., NOLAN G. P., PE'ER D.: viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature biotechnology 31*, 6 (2013), 545–552. 1, 2

[Aup14] AUPETIT M.: Sanity check for class-coloring-based evaluation of dimension reduction techniques. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (2014), ACM, pp. 134–141. 8

[BSC*14] BECHER B., SCHLITZER A., CHEN J., MAIR F., SUMATOH H. R., TENG K. W. W., LOW D., RUEDL C., RICCARDI-CASTAGNOLI P., POIDINGER M.: High-dimensional analysis of the murine myeloid cell system. *Nature immunology 15*, 12 (2014), 1181–1189. 1, 2

[BSIM14] BREHMER M., SEDLMAIR M., INGRAM S., MUNZNER T.: Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (2014), ACM, pp. 1–8. 1, 5

[dST04] DE SILVA V., TENENBAUM J. B.: *Sparse multidimensional scaling using landmark points*. Tech. rep., Stanford university, 2004. 2

[FFDP15] FADEL S. G., FATORE F. M., DUARTE F. S., PAULOVICH F. V.: Loch: A neighborhood-based multidimensional projection technique for high-dimensional sparse spaces. *Neurocomputing 150* (2015), 546–556. 2

[Gey11] GEYER C.: Introduction to markov chain monte carlo. *Handbook of Markov Chain Monte Carlo* (2011), 3–48. 3

[HR02] HINTON G. E., ROWEIS S. T.: Stochastic neighbor embedding. In *Advances in neural information processing systems* (2002), pp. 833–840. 2

[IM15] INGRAM S., MUNZNER T.: Dimensionality reduction for documents with nearest neighbor queries. *Neurocomputing 150* (2015), 557–569. 8

[IMO09] INGRAM S., MUNZNER T., OLANO M.: Glimmer: Multilevel mds on the gpu. *IEEE Transactions on Visualization and Computer Graphics 15*, 2 (2009), 249–261. 2

[JPC*11] JOIA P., PAULOVICH F., COIMBRA D., CUMINATO J., NONATO L.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2563–2571. 2, 8

[JvdMJV15] JOULIN A., VAN DER MAATEN L., JABRI A., VASILACHE N.: Learning visual features from large weakly supervised data. *Preprint arXiv:1511.02251* (2015). 7

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, Pereira F., Burges C., Bottou L., Weinberger K., (Eds.). 2012, pp. 1097–1105. 7, 8

[LBH15] LECUN Y., BENGIO Y., HINTON G.: Deep learning. *Nature 521*, 7553 (2015), 436–444. 7

[MCMT14] MARTINS R. M., COIMBRA D. B., MINGHIM R., TELEA A.: Visual analysis of dimensionality reduction quality for parameterized projections. *Computers & Graphics 41* (2014), 26–42. 9

[MKS*15] MNIH V., KAVUKCUOGLU K., SILVER D., RUSU A. A., VENESS J., BELLEMARE M. G., GRAVES A., RIEDMILLER M., FIDJELAND A. K., OSTROVSKI G., ET AL.: Human-level control through deep reinforcement learning. *Nature 518*, 7540 (2015), 529–533. 7

[PdRDK99] PEKALSKA E., DE RIDDER D., DUIN R. P., KRAAIJVELD M. A.: A new method of generalizing sammon mapping with application to algorithm speed-up. In *ASCI* (1999), vol. 99, pp. 221–228. 2

[PEP*11] PAULOVICH F. V., ELER D. M., POCO J., BOTHA C. P., MINGHIM R., NONATO L. G.: Piece wise laplacian-based projection for interactive data exploration and organization. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1091–1100. 2, 8

[PL*15] PEZZOTTI N., , LELIEVELDT B. P., VAN DER MAATEN L., HÖLLT T., EISEMANN E., VILANOVA A.: Approximated and user steerable tsne for progressive visual analytics. *Preprint arXiv:1511.02251* (2015). 2, 3, 6, 8

[PM08] PAULOVICH F. V., MINGHIM R.: Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1229–1236. 2

[PNML08] PAULOVICH F. V., NONATO L. G., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics 14*, 3 (2008), 564–575. 2, 8

[POM07] PAULOVICH F. V., OLIVEIRA M. C. F., MINGHIM R.: The projection explorer: A flexible tool for projection-based multidimensional visualization. In *Computer Graphics and Image Processing, 2007. SIBGRAPI 2007. XX Brazilian Symposium on* (2007), IEEE, pp. 27–36. 8

[PSN10] PAULOVICH F. V., SILVA C. T., NONATO L. G.: Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1281–1290. 2

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (2000), 2323–2326. 1

[Sam69] SAMMON J. W.: A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 5 (1969), 401–409. 1

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages* (1996), IEEE, pp. 336–343. 2, 5

[SMT13] SEDLMAIR M., MUNZNER T., TORY M.: Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2634–2643. 1, 2

[ST02] SILVA V. D., TENENBAUM J. B.: Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems* (2002), pp. 705–712. 2

[TDSL00] TENENBAUM J. B., DE SILVA V., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323. 1

[Tuk62] TUKEY J. W.: The future of data analysis. *The Annals of Mathematical Statistics* (1962), 1–67. 1

[VDM14] VAN DER MAATEN L.: Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research 15*, 1 (2014), 3221–3245. 2, 3, 8

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9*, 2579-2605 (2008), 85. 1, 2, 3, 4, 8

[VPN*10] VENNA J., PELTONEN J., NYBO K., AIDOS H., KASKI S.: Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *The Journal of Machine Learning Research 11* (2010), 451–490. 8

[WM04] WILLIAMS M., MUNZNER T.: Steerable, progressive multidimensional scaling. In *IEEE Symposium on Information Visualization* (2004), IEEE, pp. 57–64. 2