

# Incorporating Texture Information into Dimensionality Reduction for High-Dimensional Images

A. Vieth\*  
TU Delft  
The Netherlands

A. Vilanova†  
TU Eindhoven  
The Netherlands

B. Lelieveldt‡  
Leiden University Medical Center  
The Netherlands

E. Eisemann\*  
TU Delft  
The Netherlands

T. Höllt\*  
TU Delft  
The Netherlands

## ABSTRACT

High-dimensional imaging is becoming increasingly relevant in many fields from astronomy and cultural heritage to systems biology. Visual exploration of such high-dimensional data is commonly facilitated by dimensionality reduction. However, common dimensionality reduction methods do not include spatial information present in images, such as local texture features, into the construction of low-dimensional embeddings. Consequently, exploration of such data is typically split into a step focusing on the attribute space followed by a step focusing on spatial information, or vice versa. In this paper, we present a method for incorporating spatial neighborhood information into distance-based dimensionality reduction methods, such as t-Distributed Stochastic Neighbor Embedding (t-SNE). We achieve this by modifying the distance measure between high-dimensional attribute vectors associated with each pixel such that it takes the pixel's spatial neighborhood into account. Based on a classification of different methods for comparing image patches, we explore a number of different approaches. We compare these approaches from a theoretical and experimental point of view. Finally, we illustrate the value of the proposed methods by qualitative and quantitative evaluation on synthetic data and two real-world use cases.

**Index Terms:** Mathematics of computing—Dimensionality reduction; Human-centered computing—Visualization techniques; Human-centered computing—Visual analytics;

## 1 INTRODUCTION

High-dimensional data is commonly acquired and analyzed in various application domains, from systems biology [26] to insurance fraud detection [37]. Typically, high-dimensional data are tabular data with many columns (or attributes), corresponding to the dimensionality per item but there are no connections between items. Dimensionality reduction techniques like t-distributed Stochastic Neighbor Embedding (t-SNE) [58] or Uniform Manifold Approximation and Projection (UMAP) [48] are well-established tools used for exploratory visual analysis of such high-dimensional data [53]. Advances in imaging techniques have introduced an increasing number of imaging data modalities producing high-dimensional images (every pixel represents a high-dimensional attribute-vector). Current state-of-the-art dimensionality reduction methods are commonly used for the explorative analysis of such imaging data, for example in cultural heritage [2], biology [9], or geospatial applications [24]. However, they rely only on attribute data of pixels and do not take additional spatial information, such as texture, present in such imaging data into account. Thus, in the resulting low-dimensional embeddings, the pixels are only arranged according to their individual attributes (Fig. 1b), but do not provide any insight into texture, neighborhoods or other spatial relations common in image analysis.

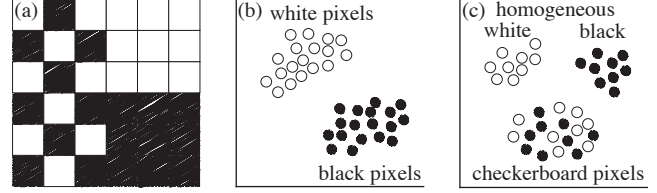


Figure 1: **Texture-aware dimensionality reduction.** An image (a) with black and white pixels forms multiple textures. Standard distance-based dimensionality reduction produces one cluster of black and one cluster of white pixels (b), a texture-aware version should create clusters for the different textures (c).

The spatial configuration is, however, commonly of interest when analyzing high-dimensional image data. For example, taking spatial neighborhood information into account, in addition to high-dimensional attributes, has led to new discoveries in single-cell biology [33]. Typical approaches to combine high-dimensional attributes and spatial information, however, rely on a two-stage process: first, high-dimensional attributes are aggregated, for example to classify pixels, then standard image analysis is performed on the aggregate images. For example, Abdelmoula et al. [1] aggregate the original attribute space of high-dimensional imaging data to a three-dimensional space using t-SNE. They use the embedding as a colormap and perform segmentation on the re-colored image. Decoupling the high-dimensional and spatial analysis in such a way has several downsides: Most importantly, boundaries between clusters in an embedding are often not well defined, and as such classification is ambiguous and has a level of arbitrariness. Issues with inaccurate classification might appear undetected and lead to wrong conclusions. Furthermore, if problems with the classification become apparent in the spatial analysis, one has to go back to the high-dimensional analysis and potentially loses all progress in the spatial analysis. Moreover, the necessary aggregation in the first step limits what is discoverable in the spatial analysis step. Therefore, we deem the integration of spatial information directly into the dimensionality reduction desirable for exploratory analysis.

We present an approach to integrate spatial information directly into the dimensionality reduction process with the goal to combine attribute and spatial information in a single embedding (Fig. 1c). Specifically, we propose to adapt the similarity computation, used in distance-based dimensionality reduction, such as t-SNE, UMAP, or Multi-dimensional scaling (MDS), to incorporate different spatial neighborhood features. We exemplarily present different similarity computation methods for such neighborhood comparisons by extending an existing classification [61] to high-dimensional images.

The main contributions of this paper are:

- incorporating texture information into distance-based dimensionality reduction for exploratory analysis of high-dimensional images through distance measures including image neighborhoods.
- the exemplary extension of t-SNE using different classes of neighborhood distance measures and their analysis.

\*e-mail: {A.Vieth | E.Eisemann | T.Hollt-1} @tudelft.nl

†e-mail: A.Vilanova@tue.nl

‡e-mail: B.P.F.Lelieveldt@lumc.nl

## 2 RELATED WORK

Dimensionality reduction and high-dimensional images are well-researched topics with a large body of work. In the following, we aim to report the work most relevant to our contribution.

### 2.1 Explorative analysis of high-dimensional images

There is a multitude of approaches for visual exploration of high-dimensional data [40]. A challenge when facing data that is additionally spatially resolved, is to effectively visualize spatial and attribute characteristics in an integrated fashion. MulteeSum [49] compares spatio-temporal gene expression data from fruit fly embryos by segmenting cells in the image and providing multiple attribute summaries per cell. Another approach is to characterize the data attributes in terms of specific features and represent them as glyphs at their respective regions in space [11]. For multivariate volume data, high-dimensional transfer functions have been employed in combination with standard volume rendering techniques [60].

Often, high-dimensional image data is visualized indirectly by first extracting features that capture interesting data characteristics and then displaying those [18, 32]. One such feature is texture. We follow Haidekker's definition of texture as "any systematic local variation of the image values" [44] since it emphasises that texture encodes spatially local relationships of pixel values. Texture feature extraction is a broad, well-established field [29] but the extension of single-channel texture features to multi-channel images is not trivial. Typically, single-channel texture features are extracted for each channel and concatenated to a feature vector. Multi-channel texture features such as color co-occurrence matrices proposed by Palm [50] are less common and engineering such features is an ongoing process in the image processing community [54].

Dimensionality reduction has been used to *create* texture features [36] and *explore* texture databases [43] but such approaches are generally out of the scope of this work.

### 2.2 Dimensionality reduction for high-dimensional images

Non-linear dimensionality reduction techniques such as t-SNE [58] and UMAP [48] have become popular techniques to visualize and explore high-dimensional data. These techniques have been applied to high-dimensional imaging data without considering texture information. Abdelmoula et al. [1] use t-SNE in a segmentation pipeline for high-dimensional images. They embed pixels according to their high-dimensional attribute values to a three-dimensional space, followed by coloring the pixels by using the 3D embedding coordinates as coordinates in the  $L^*a^*b$  color space. The resulting color images are then used to aid the segmentation with the goal of identifying tissue segments with similar properties, according to the original attribute space. Recently, Evers et al. [14] followed a similar approach to identify regional correlations in spatio-temporal weather ensemble simulations with the main difference of using MDS instead of t-SNE. Others combine dimensionality reduction with segmented image data. Facetto [35] combines un- and semi-supervised learning to aid in the visual analysis of high-dimensional imaging data in the field of structural biology. After segmenting cells and aggregating their corresponding attributes to features, they use UMAP to display the cells according to their similarities. ImaCytE [55] is a visual analysis tool for similar data that focuses on the analysis of cell neighborhoods. Again, cells are segmented and the attributes of pixels within the cells aggregated. Cells are laid out according to their similarity in the attribute space using t-SNE and the resulting information is used to analyze spatial neighborhoods. All of these applications and tools make use of standard dimensionality reduction methods that do not incorporate spatial information and instead follow a two-step approach using the results of either dimensionality reduction or spatial analysis as input to the other. Instead, we directly include spatial information in the dimensionality reduction process to reduce the number of steps and potential points of failure in interactive analysis.

In the analysis of hyperspectral images, dimensionality reduction is an important step for pixel classification. A common approach to include spatial information relies on computing the first couple of principal components of the high-dimensional data and then continuing with classic image processing methods that work on scalar or color data to extract spatial neighborhood information [16, 20]. For example, morphological image processing techniques are used to capture spatial structure in high-dimensional images [8]. Spatial-spectral local discriminant projection [28] takes a more direct way of combining spatial and spectral information into dimensionality reduction by incorporating a weighting factor into the neighborhood preserving embedding that represents the spectral similarity between spatially neighboring pixels. However, this and similar hyperspectral image analysis methods [41] rely on training with ground truth data, which is typically not available in exploratory data analysis. Recently, Halladin-Dąbrowska et al. [24] proposed a workflow using t-SNE for cleaning ground truth data, however, they do not include spatial neighborhood information in their embeddings.

A straightforward way to inform dimensionality reduction techniques of images about their spatial domain is to consider each data point's spatial location in the point similarity measure used during the embedding. Spherical SNE [42] devises a similarity function between data points in the style of bilateral filtering that weights attribute distance with pixel location distance. This approach, however, does not capture the similarity of the local structure around the compared points, which we aim for.

### 2.3 Multivariate graph visualization and embedding

Graph-based techniques are commonly applied to pattern recognition and computer vision problems on imaging data [7]. For that, images are interpreted as graphs, where each pixel is interpreted as a node and neighboring pixels are connected by a link. Several techniques for graph drawing aim to incorporate *network structure* and *node attributes* exist [34] that might be helpful for our goals.

GraphTPP [21] focuses on a visual combination of node attributes and connections in a 2D graph layout. First, principal component analysis (PCA) is applied to the data using only the attributes. Then links between nodes are overlaid on the resulting scatterplot. The user can then manually reposition points according to their interpretation, and compute a new linear projection that best fits the modified layout. GraphTSNE [39] aims to preserve graph connectivity and node attribute similarity. It does so by training a graph convolutional network on a modified t-SNE loss that combines the squared euclidean distance between node attributes and the shortest-path distance between the nodes on the graph. Their design seeks to position two points close in the embedding either when their attributes are similar or they are connected by an edge. Similarly, MVN-Reduce [47] defines a distance measure between two nodes as the sum of a node's attribute distance and their weighted shortest path distance on the graph. The resulting distances are used as input to distance-based dimensionality reduction methods like MDS. The Heterogeneous Network Embedding (HNE) framework [5] aims to create embeddings that position data points with links closer and those without further away from each other. Therein, a neural network is trained with a loss function that builds on a similarity term between point attributes that is weighted depending on their respective node linkage.

Applied to an image, transformed into a graph, all the aforementioned approaches essentially combine the pixel location distance (geodesic distances on the graph) with the attribute distance, not dissimilar to what is described for the Spherical SNE [42] (Sect. 2.2). In contrast, the goal of our proposed approach is to compare local texture structures rather than absolute distances. Two pixels are compared by taking into account the structure of the high-dimensional values in the spatial neighborhoods of the two pixels.

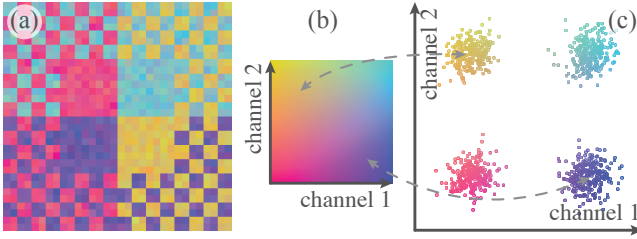


Figure 2: **Synthetic test image dataset** with  $X \times Y = 32 \times 32$  and  $C = 2$ . The distribution in attribute space (c) reveals four groups. We use a 2D colormap (b) to color pixels in image space (a) according to their attribute information.

### 3 BACKGROUND

We will first define high-dimensional image data and spatial neighborhoods more rigorously in Sections 3.1 and 3.2 respectively. In Sect. 3.3, we briefly introduce t-SNE as an example of distance-based dimensionality reduction methods.

#### 3.1 High-dimensional image data

While, in general, dimensionality reduction methods can be applied on other high-dimensional data sets as well, we focus on high-dimensional, digital 2D images. Each pixel is an image element at a unique location with an associated attribute vector. We can formalize such an image as a discrete function  $f : X \times Y \rightarrow \mathbb{R}^C$  from the spatial domain  $X \times Y \subset \mathbb{N}^2$  to the attribute range  $\mathbb{R}^C$ .  $X$  and  $Y$  are the sets of pixel coordinates that span the image domain along its two dimensions while  $C$  is the number of attributes, or image channels. A pixel is indexed with  $\mathbf{i} = (x_i, y_i)$  where  $1 \leq i \leq n$  and  $n = |X \times Y|$ . Now,  $f(\mathbf{i}) = \mathbf{g}_i$  yields the pixel’s high-dimensional attribute vector at location  $\mathbf{i}$  with  $\mathbf{g}_i = [g_{i1}, \dots, g_{iC}]$ .

We can also interpret such a high-dimensional image as a combination of several scalar images, each representing one channel of the high-dimensional image. Therefore, we refer to the  $c$ -th channel of the image with  $f_c$ , where  $1 \leq c \leq C$ , such that  $f_c = [g_{1c}, \dots, g_{nc}]$  denotes the values of the  $c$ -th channel for all pixels in the image.

Fig. 2 shows a synthetic toy-example of a ‘high-dimensional’ image with two attribute channels (i.e.,  $C = 2$ ). The spatial layout is displayed in Fig. 2a with each pixel color coded according its two attribute values using a 2D colormap (Fig. 2b). Fig. 2c shows a scatterplot of all attribute values. Four groups are clearly distinguishable based on the attributes. In image space the four groups form eight visually distinct regions, four group-homogeneous and four consisting of checkered patches. In Sect. 5, we use this image to showcase the characteristics of our proposed approaches and compare it to a standard t-SNE embedding.

#### 3.2 Neighborhood definition

The notion of neighborhood is twofold for high-dimensional image data: one can distinguish between neighbors in the spatial and the attribute domain. We refer to the *attribute neighborhood*  $\mathcal{N}_i^{C,k}$  of pixel  $\mathbf{i}$  as the set of indices of the  $k$  pixels with the smallest distances  $d$  to the attributes of pixel  $\mathbf{i}$ :

$$\mathcal{N}_i^{C,k} = \underset{\mathbf{j}=(x_j,y_j), 1 \leq j \leq n}{k\text{-argmin}} d(\mathbf{g}_i, \mathbf{g}_j), \quad (1)$$

where  $k\text{-argmin}$  performs a selection of the  $k$  arguments in  $\{\mathbf{j} = (x_j, y_j) : 1 \leq j \leq n\}$  that yield the  $k$  smallest distances. Typically, the squared Euclidean distance is chosen as the distance measure  $d(\mathbf{g}_i, \mathbf{g}_j) = \|\mathbf{g}_i - \mathbf{g}_j\|_2^2$ , but other distances like cosine or Hamming distance are popular choices as well.

Next, we define the *spatial neighborhood*  $\mathcal{N}_i^{S,\eta}$ . We consider a square spatial neighborhood of  $M = (2 \cdot \eta + 1)^2$  pixels centered

at  $\mathbf{i}$ . With  $\mathbf{i} = (x_i, y_i)$ , this results in the set of spatially-neighboring indices

$$\mathcal{N}_i^{S,\eta} = \{(x_i + r, y_i + s) : -\eta \leq r, s \leq \eta\}. \quad (2)$$

We focus on two-dimensional, rectilinear spatial layouts for the sake of simplicity, but in principle our method is trivially extendable for data resolved in three spatial dimensions, i.e., multivariate volumetric data:  $\mathcal{N}_i^{S,\eta} = \{(x_i + r, y_i + s, z_i + t) : -\eta \leq r, s, t \leq \eta\}$ .

#### 3.3 t-SNE

Distance-based dimensionality reduction methods like LargeVis [56], UMAP [48] and t-SNE [57, 58] all share a similar basic structure. First, based on a distance measure in the attribute space, they construct a neighborhood graph that captures local neighborhoods. Then, a low dimensional layout is produced, with the aim to represent these neighborhoods as faithfully as possible; this process is guided by optimizing a specific cost function. We will discuss our framework using the example of t-SNE. While the same concepts are applicable to all distance-based dimensionality reduction methods, we deem it easiest to follow one specific example.

To create a low dimensional embedding as described above, t-SNE uses a symmetric joint probability distribution  $P$  to describe similarities between high-dimensional points. Likewise, a joint probability distribution  $Q$  encodes the similarity of the corresponding low-dimensional points in the embedding space. Starting with a random initialization of  $Q$ , the embedding points are iteratively moved such that the distribution  $Q$  matches  $P$  well. This optimization process is guided by the Kullback-Leibler ( $KL$ ) divergence that measures the divergence of  $P$  and  $Q$  as cost function  $Cost(P, Q)$

$$Cost(P, Q) = KL(P, Q) = \sum_i \sum_{j \neq i} p_{ij} \ln \left( \frac{p_{ij}}{q_{ij}} \right), \quad (3)$$

where the probability  $p_{ij}$  represents the similarity of two high-dimensional data points  $\mathbf{g}_i$  and  $\mathbf{g}_j$ , and  $q_{ij}$  represent the similarity of the two corresponding low-dimensional data points in the embedding.  $p_{ij}$  is symmetric and computed as

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}, \quad (4)$$

where  $p_{j|i}$  can be interpreted as the probability that the point  $\mathbf{g}_j$  is in the neighborhood of the point  $\mathbf{g}_i$  in the attribute space.  $p_{j|i}$  is calculated using the distance measure  $d(\mathbf{g}_i, \mathbf{g}_j)$  between the high-dimensional points:

$$p_{j|i} = \begin{cases} \frac{\exp(-d(\mathbf{g}_i, \mathbf{g}_j)/(2\sigma_i^2))}{\sum_{\mathbf{k} \in \mathcal{N}_i^{C,k}} \exp(-d(\mathbf{g}_i, \mathbf{g}_k)/(2\sigma_i^2))} & \text{if } \mathbf{j} \in \mathcal{N}_i^{C,k} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The number of nearest neighbors  $|\mathcal{N}_i^{C,k}| = 3\phi$  can be steered with a user-defined perplexity  $\phi$ . The bandwidth  $\sigma_i$ , in turn, is determined based on the given perplexity value such that

$$\phi = 2^{-\sum_{\mathbf{j} \in \mathcal{N}_i^{C,k}} p_{ji} \log_2 p_{ji}}. \quad (6)$$

#### 4 TEXTURE-AWARE DIMENSIONALITY REDUCTION

To incorporate spatial neighborhood information into low-dimensional embeddings, we propose a set of distance measures that take the spatial neighborhood  $\mathcal{N}_i^{S,\eta}$  of pixels in high-dimensional images into account. The distance between the attributes of two attribute vectors  $d(\mathbf{g}_i, \mathbf{g}_j)$  is thus replaced by a new texture-informed distance  $d_s(\mathbf{i}, \mathbf{j}, f, \eta)$  and the  $k$  nearest neighbors in Equation 5 will be based on this new measure as well. Since we aim to compare the spatial neighborhood of two pixel, it does not suffice to include

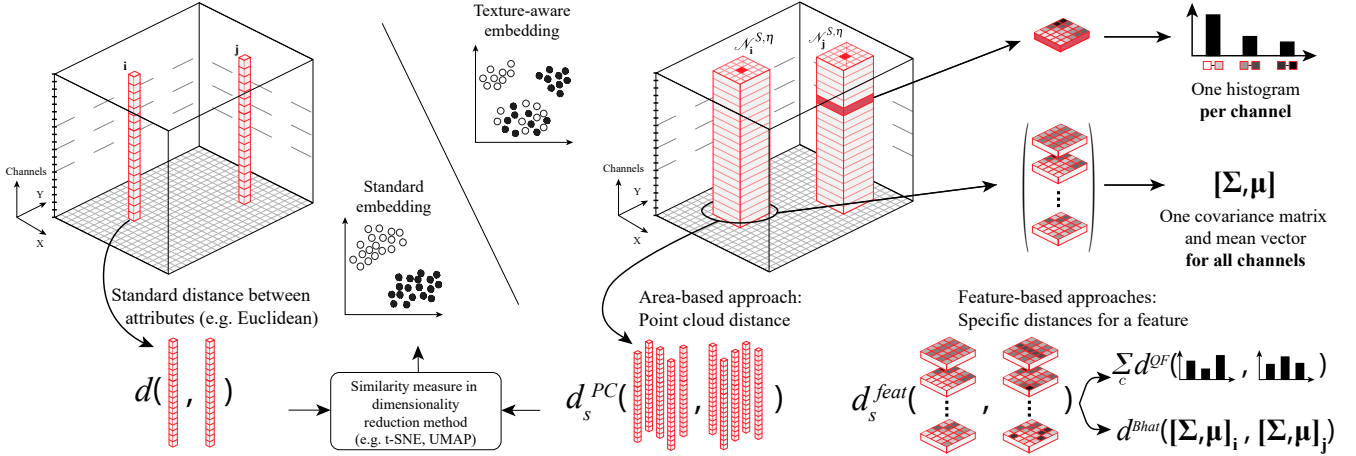


Figure 3: **Incorporating image texture information into dimensionality reduction** by adapting the distance measure that defines pixel similarity. High-dimensional image data is depicted in a data cube. Standard t-SNE compares pixels based on their attribute vectors only, e.g., using Euclidean distance (left). We propose to also consider the spatial neighborhood  $\mathcal{N}_i^{S, \eta}$  of the pixels (right) and present different approaches. Feature-based methods (FBM) derive texture features per channel (e.g., local histogram) or across channels (e.g., covariance matrix) and compare those, while area-based method (ABM) (e.g., point cloud distance) compare sets of original attribute vectors directly.

their image coordinates or spatial distance; rather, it is necessary to involve each pixel’s spatial local neighborhood. Essentially, we are comparing image patches instead of single pixel values. All other steps of the embedding process remain as they were.

#### 4.1 Comparing image patches

In the following, we will present a number of texture-aware distance measures  $d_s$ . As the space of potential measures is vast, we will focus on a few exemplary measures, following the classification of distance measures for image patches introduced by Zitová and Flusser [61]. In particular, they distinguish image patch comparison into area-based methods (ABM) and feature-based methods (FBM).

ABM and FBM for image patch comparison differ in their approach to compute similarity scores. ABM directly work with the pixel’s attribute values of the two image patches to compare. They are sometimes called intensity-based instead, which might reflect the immediate usage of the attribute values more aptly. In contrast, FBM follow a two-step approach of first computing features for each patch and then comparing those. ABM can be further categorized into correlation-like methods (for example point cloud distances), Fourier methods, mutual information methods, optimization methods [61]. We refer to Goshtasby [23] for an extensive overview and discussion of area-based similarity and dissimilarity measures. Likewise, there exists a rich body of literature discussing image patch descriptors and appropriate feature matching methods used in FBM [31, 38]. To note, most FBM extract various salient features per image with the goal of matching the whole images. Such methods do not necessarily produce one feature value per pixel which is desired for computing the pairwise distance between pixels as in our case. More extensive and general discussions of ABM and FBM can be found in [45, 61]. Here, we will present how to structurally extend image patch comparison to high-dimensional images, and will showcase t-SNE with exemplary methods for each category.

#### 4.2 Application to high-dimensional images

Classically, both FBM and ABM are applied to grayscale or color images but we work with high-dimensional images instead. Typical ABM involve direct comparison of individual points and thus can be applied to multi-channel data, by directly comparing the multi-dimensional points with any applicable metric. For FBM, a direct extension of a single-channel feature to multi-channel data does not always exist or a straightforward extension to multiple dimensions suffers from problems. Consider, for example, local histograms. They work well to summarize single-channel data, but as the number of dimensions of the histogram reflects the number of channels in the

data, a local histogram for high-dimensional data would typically be sparse due to the curse of dimensionality [3]. Therefore, in addition to creating multi-dimensional features, we also consider computing traditional one-dimensional features per channel and then compute the distance between the resulting feature vectors.

In the following, we will present a point cloud distance, namely the Chamfer distance, to showcase an area-based method. We use channel-wise histograms as single-channel features and the more general covariance matrix of the neighborhood values as a multi-channel feature to provide examples for FBM. Fig. 3 illustrates the concept behind the three approaches. An example of these methods on synthetic data will be discussed in more detail in Sect. 5.1.

#### 4.3 Feature-based methods

A wide range of image features exist and an adequate choice depends on the application and the goal of the analysis [29]. It is out of the scope of this paper to cover all possibilities. We focus on spatial heterogeneity which has been successfully applied, for example, in biomedical tumor analysis [10] or geo-spatial data analysis [17].

We investigate two texture features that capture local heterogeneity in scalar images. As a single-channel feature example, we capture heterogeneity with local histograms per pixel and channel. Histograms have been successfully used for texture synthesis [25] and lend them-self well as texture features. But since local histograms do not adapt well to high-dimensional data, we use the covariance matrix  $\Sigma$  and channel-wise means  $\mu$ , roughly generalizing the histogram measure of dispersion, as a multi-channel feature. Covariance information has as well been shown to be useful texture information for texture synthesis in generative adversarial networks [19]. For FBM, the neighborhood distance becomes  $d_s(i, j, f, \eta) = d_s^{feat}(\mathcal{T}_\eta(f, i), \mathcal{T}_\eta(f, j))$  with  $\mathcal{T}_\eta$  being the chosen feature extraction operator that will depend on the use case and neighborhood size parameter  $\eta$ .

It is worth noting that the approach of computing the features separately per channel assumes independence between all channels  $f_c$ , which is typically not the case. This means that in some cases certain combinations of attribute values and texture features cannot be distinguished. The covariance matrix feature (Sect. 4.3.2), and point cloud distance-based, (Sect. 4.4), approaches do not have this limitation since they use the full attribute space to measure the distances.

#### 4.3.1 Local histograms features

Local histograms are a common way to characterize texture in scalar image processing. We compute one feature, i.e., the normalized local histogram, per pixel and channel (confer the right side of Fig. 3). The histogram of attribute values of channel  $c$  in the spatial neighborhood  $\mathcal{N}_i^{S,\eta}$  is referred to as the vector  $\mathbf{h}_{ic} = [h_{ic1}, \dots, h_{icB}]$ , where  $B$  is the total number of bins. All entries are normalized by the total amount of pixels in the neighborhood. As the histogram is represented as a vector, rather than a single scalar, this yields a feature matrix per pixel:

$$\mathcal{T}_{hist}(f, \mathbf{i}) = [\mathbf{h}_{i1}, \dots, \mathbf{h}_{iC}]. \quad (7)$$

This means, to compute the distance between two pixels, we now need to compare two vectors of histograms. We can interpret a histogram as an estimate of a probability density function. As such, we can choose one of the many distance functions defined between probability distributions. One such distance is the quadratic form (QF) distance [12], defined as

$$d^{QF}(\mathbf{h}_{ic}, \mathbf{h}_{jc}) = (\mathbf{h}_{ic} - \mathbf{h}_{jc})^T \mathbf{A} (\mathbf{h}_{ic} - \mathbf{h}_{jc}) \quad (8)$$

$$= \sum_{b,k=1}^B a_{bk} (h_{icb} - h_{jcb})(h_{ick} - h_{jck}). \quad (9)$$

Here,  $\mathbf{A} = \{a_{bk}\}$  with  $0 \leq a_{bk} \leq 1$  and  $a_{bb} = 1$ , enables attributing a weight between bin indices, e.g., to take distance into account. We use  $a_{bk} = 1 - (|b - k|)/B$  as proposed by Equitz et al. [12].

With the distance per channel in place, we can define a distance  $d_s^{feat}$  for all channels as the sum of all channel-wise feature distances:

$$d_s^{hist}(\mathcal{T}_{hist}(f, \mathbf{i}), \mathcal{T}_{hist}(f, \mathbf{j})) = \sum_{c=1}^C d^{QF}(\mathbf{h}_{ic}, \mathbf{h}_{jc}). \quad (10)$$

#### 4.3.2 Covariance matrix and means

A more general dispersion feature for multi-channel data are covariance matrices. We will use these in combination with channel-wise mean values as an example for multi-channel features  $\mathcal{T}_{cov}(f, \mathbf{i}) = [\Sigma_i, \mu_i]$ . Each entry in  $\Sigma_i = \{\sigma_{ab}\}$  represents the variance between the attribute values within the spatial neighborhood  $\mathcal{N}_i^{S,\eta}$  of the channels  $f_a$  and  $f_b$ ; the vector  $\mu_i$  holds the mean values within the same neighborhood per channel.

One measure that is suited for comparing our covariance matrix feature is the Bhattacharyya distance [6]:

$$d_s^{Bhat}(\mathcal{T}_{cov}(f, \mathbf{i}), \mathcal{T}_{cov}(f, \mathbf{j})) = \frac{1}{8} (\mu_i - \mu_j)^T \Sigma^{-1} (\mu_i - \mu_j) + \frac{1}{2} \ln \left( \frac{\det \Sigma}{\sqrt{\det \Sigma_i \det \Sigma_j}} \right) \quad (11)$$

with  $\Sigma = \frac{\Sigma_i + \Sigma_j}{2}$  and  $\det(\Sigma)$  denoting the determinant of a matrix  $\Sigma$ .

#### 4.4 Area-based methods

A straightforward example ABM is to interpret the attribute vectors of pixels in the spatial neighborhood  $\mathcal{N}_i^{S,\eta}$  as a high-dimensional point cloud, see Fig. 3. Instead of comparing explicit texture features, we are now computing distances in the high-dimensional space defined by the data attributes directly. Point cloud distances have been used to compare single-channel images [30] and many naturally extend to higher dimensions since they are based around norms of differences between attribute vectors. To stay consistent with the previously-established notation, one can think of it as simply defining the feature as the data values in the spatial neighborhood without any transformation,  $\mathcal{T}_{PC}(f, \mathbf{i}) = \mathbf{T}_i = \{\mathbf{g}_j : \mathbf{j} \in \mathcal{N}_i^{S,\eta}\}$ , leading to a point cloud distance  $d^{PC}(\mathbf{T}_i, \mathbf{T}_j)$ .

Multiple distance measures between point clouds exist [11]. The choice, which one to use, depends on the application that needs to be addressed. A commonly used point cloud distance is the Chamfer (pseudo-) distance [15]. Conceptually, to calculate this point cloud distance  $d^{PC}$  between the spatial neighborhoods of two pixels  $\mathbf{i}$  and  $\mathbf{j}$ , for each point in the spatial neighborhood  $\mathcal{N}_i^{S,\eta}$  we find the closest point in the other neighborhood  $\mathcal{N}_j^{S,\eta}$  with respect to a metric (e.g., squared Euclidean distance in our implementation) and average these closest point distances. This yields:

$$d_s^{PC}(\mathbf{T}_i, \mathbf{T}_j) = \frac{1}{M} \sum_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \min_{\mathbf{p} \in \mathcal{N}_j^{S,\eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 + \frac{1}{M} \sum_{\mathbf{p} \in \mathcal{N}_j^{S,\eta}} \min_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 \quad (12)$$

In comparison with other point cloud distances, like the closely related Hausdorff distance, the Chamfer distance is more robust against outliers in the neighborhoods. Unlike the max-min Hausdorff distance, here, we take the average of all point-wise minima instead of their maximum. Supplemental Material S3 compares several Hausdorff family distances for the previously introduced toy-example image discussed in Sect. 5.

#### 4.5 Computational complexity

The computational complexity for the presented approaches can be split into two parts: first the feature extraction and second the actual distance functions.

The histogram feature computation in our implementation scales linearly with the number of spatial neighbors. For a single pixel and channel, this yields the complexity for the **local histogram** feature extraction:  $\mathcal{O}(M)$ , scaling linearly, only with the number of pixels in the neighborhood  $M = (2 \cdot \eta + 1)^2$ . The **covariance matrix** feature calculation is dominated by the computational complexity of a matrix multiplication between two matrices of size  $M \times C$ , namely  $\mathcal{O}(CM^2)$ .

The distance calculation is the more time-consuming step for the presented methods. For the **local histogram** feature approach the QF distance computation scales quadratically with the number of bins  $B$ , which dominates its complexity in  $\mathcal{O}(CB^2)$ . **Covariance matrices and mean** comparison with the Bhattacharyya distance is more expensive. Its computation involves matrix-vector multiplication and determinant calculation. Using LU decomposition for the latter yields  $\mathcal{O}(C^3)$ .

Finally, the **point cloud distance** requires the computation of all pairwise distances between the two neighborhoods. As a result, its complexity scales quadratically with the neighborhood size  $\mathcal{O}(CM^2)$ . Note, that neighborhood-based dimensionality reduction methods, like t-SNE and UMAP, use the point distances to construct k-nearest neighbor (knn) graph, which requires the computation of all pairwise distances for the whole dataset. Naively, the knn graph construction would scale quadratically with the number of pixels rather than the neighborhood. However, most modern implementations of t-SNE, and UMAP avoid quadratic complexity by using approximated knn algorithms [48, 51], as do we, see Sect. 6.

We have also experimentally verified this analysis, showing that the local histogram approach is the fastest and the Bhattacharyya distance the slowest. The full data can be found in the Supplemental Material S5.

#### 4.6 Spatial weighting

So far, we treated all pixels in the spatial neighborhood uniformly. In order to define specific patterns of interest within the neighborhood, for example by assigning pixels further away from the center a lower importance, we introduce a spatial weighting  $\mathbf{w}$ . Weights are

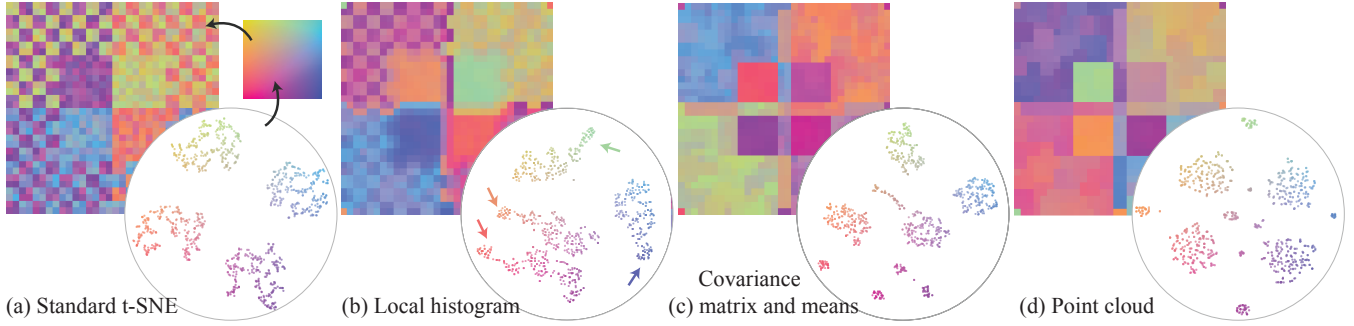


Figure 4: **Comparison of different embeddings** of the synthetic image data shown in Fig. 2 using standard t-SNE (a) and the three presented texture-aware approaches (b-d). We use the re-coloring approach introduced in Fig. 2 to indicate embedding structure in image space. Here, we use the 2D embedding coordinates to index the colormap shown in (a). As a result, pixels that are close in the embedding space have similar colors in image space.

consistent with respect to the center  $\mathbf{i}$  of a neighborhood, which implies that a pixel position  $\mathbf{p}$  receives the weight:  $\mathbf{w}(\mathbf{i} - \mathbf{p})$ .

For the histogram features, spatial weights with  $\sum \mathbf{w} = 1$  can be included in the histogram construction by scaling pixel attributes by the weight. Weights can be introduced into the covariance matrix and mean computation as detailed in the Supplemental Material S1, where we also cover the integration into the Chamfer point cloud distance. A two-dimensional Gaussian kernel is a natural weighting choice as it assigns smoothly decreasing importance to pixels further away from the neighborhood center.

## 5 APPLICATION ON SYNTHETIC DATA

To illustrate some of the properties of the different approaches, presented in Sect. 4.3 and Sect. 4.4, we created a simple synthetic image data set, shown in Fig. 2. The image consists of  $32 \times 32$  pixels, with two attribute channels, separating the pixels into four groups (Fig. 2c). As seen in Fig. 2a the spatial layout includes four homogeneous regions in the center and checkerboard patches around them, each constructed by alternating  $2 \times 2$  pixel blocks of two different classes.

Fig. 4 shows a standard t-SNE embedding of the synthetic data set as well as embeddings using the three described methods. All four embeddings were computed using a perplexity of 20 and 1,000 gradient descent iterations. For the three texture-aware approaches we considered a uniformly weighted  $3 \times 3$  neighborhood. To indicate structure derived from the t-SNE embedding in image-space without clustering, we use a simple re-coloring previously shown by Höllt et al. [27]. In short, 2D coordinates derived from the embedding are added to each pixel. The pixel is then assigned a color by using these coordinates as a lookup into a 2D colormap. As a result, pixels that are close in the embedding space, and thus are similar according to the used distance metric, will have a similar color in the image representation. We use t-SNE as an example throughout the paper, similar embeddings using UMAP and MDS can be found in Supplemental Material S2.

**Standard t-SNE** (Fig. 4a) separates the pixels into four groups, one per class, with some small scale structure within each class, introduced by noise in the data. However, the embedding does not give any insight into the spatial layout of the four classes. In particular, the pixels of each class positioned in the checkerboard pattern cannot be distinguished from pixels of the same class in the central homogeneous regions.

Fig. 4b, shows the embedding and re-coloring using the **Local histograms and QF distance**. We defined the number of histogram bins using the Rice rule:  $B = \lceil 2\sqrt[3]{M} \rceil = 5$ , with the neighborhood size  $M = 3 \times 3 = 9$ . The resulting embedding is somewhat less clear than the standard t-SNE one, consisting of only three major clusters, however, with more structure within those clusters. The four homogeneous areas in the center show up in separated areas in the embedding (arrows), indicated by their individual colors. These

regions are loosely connected to larger regions in the embedding containing the pixels from the checkerboard regions. Notably, the individual two classes forming a checkerboard region do form separate regions within the larger clusters to some degree. However, pixels of the same class in two different checkerboard regions, for example, pixels of with small values in both channels (Fig. 2), which are present in both checkerboards on the left image half, are separated, as indicated by the blue-ish and orange-ish colors.

The result of our approach using the **covariance matrix and means feature** can be seen in Fig. 4c. We can clearly identify nine separate clusters in the embedding. The four homogeneous regions correspond to the four small clusters on the bottom of the embedding, while the four larger clusters represent the four checkerboard regions, as indicated by the recolored image. The ninth cluster corresponds to the boundaries between the checkerboard regions. Again, the homogeneous areas are separated from the checkered but with much sharper boundaries. Different from the previously described approaches, however, each checkerboard area is recognized as a single cluster in the embedding, meaning that the checkerboard pattern is not visible anymore in the re-colored image. In an exploratory visual analysis setting, this would facilitate the selection and further analysis of regions with specific spatial neighbourhood characteristics.

The **Point cloud distance** approach, here specifically using the Chamfer distance, yields a similarly straightforward partitioned embedding in Fig. 4d. Again, all four homogeneous image patches are separately clustered as well as the checkerboard regions. The borders between the different regions now also created individual clusters in the middle of the embedding.

To quantitatively analyse the approaches, we compute the  $k$ -nearest neighbor hit, as described by Espadoto et al [13]. The average neighbor hits for 63-nearest neighbors in embedding space are 77.9% (point cloud distance), 79.4% (Bhattacharyya distance) and 80.4% (QF distance) whereas the standard t-SNE embedding yields 35.1%. While the point cloud and Bhattacharyya distance result in a higher neighbor hit for small neighbor numbers, their quality decreases slightly faster for larger numbers than the QF distances hit. See Supplemental Material S0 and Fig. 8b in Supplemental Material S2 for full details.

## 5.1 Discussion

All three presented texture-aware approaches are able to distinguish between several spatial arrangements of the high-dimensional image.

A drawback of local histogram features is the number of bins as an additional hyperparameter. Since there is no obvious choice for a good setting, the user has to fall back to heuristically setting this value. Further, in the scope of this work, we only discuss the QF distance for comparing histograms. Other distance measures are available and would likely produce different results. As a per-channel feature-based approach the local histograms implicitly assume channel independence. Thus, they cannot capture multi-

dimensional texture patterns. Using multidimensional histograms instead of a 1D histogram per channel might be able to capture such patterns. However, such an approach would drastically increase computational complexity. The histogram size grows exponentially with the number of dimensions, quickly making storing histograms and computing distances infeasible. Further, such histograms are in danger of quickly becoming very sparse and as such would not provide a useful basis for comparison anymore.

The covariance matrix feature can capture multiple attribute dimensions without requiring channel independence with the same goal of comparing the distribution of values within the defined neighborhood. However, instead of comparing all individual values, they are represented in an approximate way based on the assumption that the values are Gaussian distributed. The point cloud distance does not make this assumption and compares all attribute vectors to each other. If we compare the covariance matrix feature with the point cloud distance results, the most prominent difference between the embeddings is how they treat the borders between cluster regions. When data has a bi-modal distribution in a channel, the Gaussian assumption in the covariance feature does not reveal this case, whereas the point cloud distance would.

An advantage of the FBM methods is that they produce features that can aid the interpretation of structure in the resulting embeddings. Visualizing those features in combination with the embeddings is an interesting avenue for future work.

The spatial weights as introduced in Sect. 4.6 and spatial neighborhood size  $\eta$  affect the approaches to different degrees. See the Supplemental Material S4 for a brief overview of different neighborhood sizes and spatial weights. For example, the Chamfer point cloud distance produces very similarly clustered embeddings for several neighborhood sizes and is — with respect to the synthetic data set — not much affected by radially decreasing spatial weights for this example. Meanwhile, using the histogram feature, the checkered pixels are clustered differently when weights are applied.

## 6 IMPLEMENTATION

We implemented the described distance measures as an extension for the open-source t-SNE implementation in HDI [51, 52], where we use HNSW [46] with our custom distance functions to create the approximated k-nearest neighbor graph. The framework is implemented in C++ and OpenGL for GPU-based calculations; a Python wrapper is provided as well. Our library is available as open-source on GitHub [59].

## 7 USE CASES

Here, we illustrate the application of the presented approaches for visual data exploration using two use cases. The first use case (Sect. 7.1) describes the exploration of hyperspectral images, commonly used in geospatial analysis. For the second use case (Sect. 7.2), we applied our method to imaging mass cytometry data, a method that is recently gaining attention in systems biology. We use the Chamfer point cloud distance as an example for an area-

based method and the covariance matrix feature exemplarily as a multi-channel feature-based method. The Bhattacharyya distance comes with a high computational cost for high large channel numbers, hence the usage of the point cloud distance for the hyperspectral image example and the covariance matrix feature for the systems biology use case.

### 7.1 Hyperspectral imaging

Similar to digital photography, hyperspectral imaging captures bands of the electromagnetic spectrum. Instead of only three channels for red, green, and blue in digital photography, hyperspectral imaging captures hundreds of narrow spectral bands at the same time, each corresponding to a channel in the output image. This spectral signature can be used to recognize materials or objects in the image. To fully exploit the information in hyperspectral images, spatial relations between the high-dimensional pixels need to be considered [20].

Here, we present a use case, based on the Indian Pines data set [4]. The data set is an established reference hyperspectral image obtained by airborne visible/infrared imaging spectrometry, covering 220 adjacent spectral bands, of which 200 are typically used (after discarding 20 water absorption bands that do not contain useful information). We consider a  $145 \times 145$  pixel cutout of the data set, known as *Site 3*. This subset of the data is one of three ‘intensive’ test sites and thus has been well documented. A ground truth (Fig. 5e), providing labels indicating ground usage, such as fields, grassland, or houses for all pixels is available for the data set. We use this ground truth data to verify that the structure in our embeddings is meaningful; however, it must be noted that in explorative analysis such a ground truth is not available and information would need to be derived from the embedding structure. Each pixel in the data set maps to a  $20 \times 20$  meter area on the ground. The 200 channels of this cutout cover wavelengths range from to 400-1300 nm in roughly 9-10 nm steps.

We computed embeddings using standard t-SNE (Fig. 5a) and our texture-aware t-SNE using the Chamfer point cloud distance with a  $5 \times 5$  neighborhood (Fig. 5c). To provide a better baseline than standard t-SNE, we applied a Gaussian filter to each channel of the original image data and derived a standard t-SNE embedding from the filtered image (Fig. 5b). A Gaussian filter applies smoothing to an image by convolving it with a Gaussian kernel. Thus in the resulting image every pixel is a combination of a small neighborhood, providing a straightforward way of incorporating some spatial neighborhood information. We computed all embeddings using a perplexity of 30 and 5,000 gradient descent iterations. In Fig. 5, we color-code the ground truth labels on each embedding to indicate how well the structure in the embedding corresponds to structure in the images.

The embedding based on the Chamfer point cloud distance shows more structure than the other two embeddings. Notably, the colored points, corresponding to the labels of the ground truth, form more clearly distinguished clusters (see, for example, the orange points in

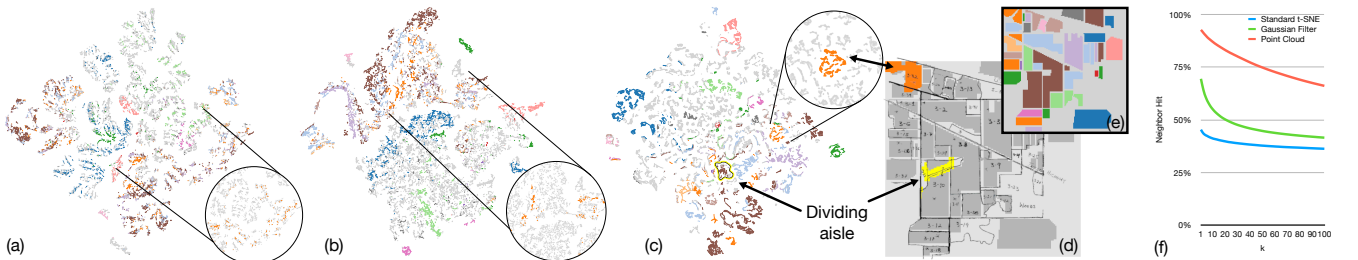


Figure 5: **Indian Pines: Comparison of texture-aware and standard t-SNE embeddings.** Embeddings using standard t-SNE (a), standard t-SNE applied after Gaussian filtering (b), and our point cloud based t-SNE (c). A manually annotated map and ground truth labels are shown in (d) and (e), respectively. Points in the embeddings are colored, according to ground truth labels for a qualitative comparison of embedding structure. Finally, we show the ration of k-nearest neighbors in embedding space with the same ground truth label for different k in (f). Note, that for an exploratory use-case no ground truth is available and we only show it here to illustrate the properties of the embeddings.

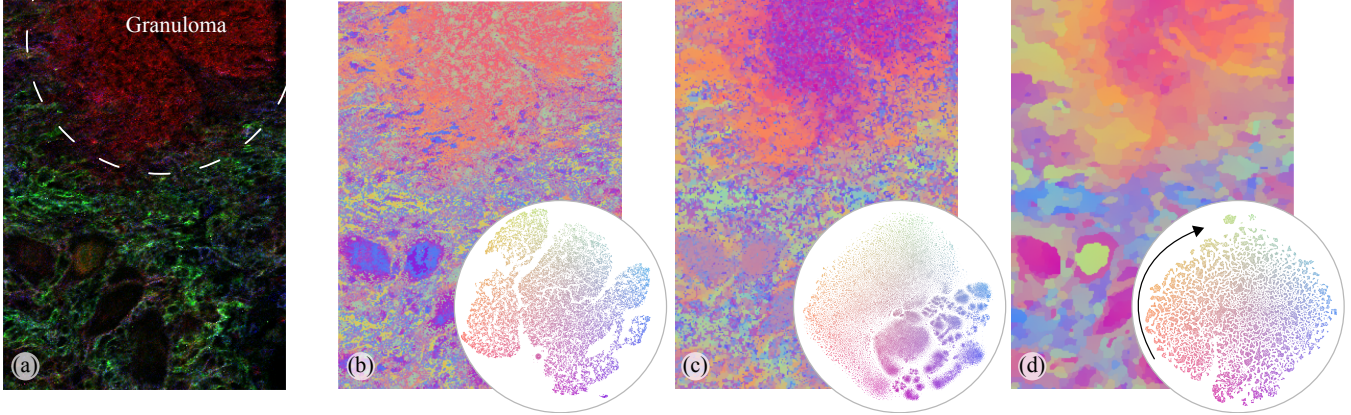


Figure 6: **Imaging mass cytometry.** A false coloring image of a lung tissue sample gives an idea of the tissue structure in (a) with the granuloma enclosed by the dashed line. Embeddings derived using standard t-SNE (b), and texture-informed embeddings using the covariance matrix and Gaussian weighting with a  $3 \times 3$  (c) and  $7 \times 7$ -sized (d) neighborhood are shown in combination with recolored images to indicate embedding structure in image space.

the insets of Fig. 5). The other two embeddings show many clusters containing points belonging to multiple regions. Most notably is the weak separation of the background, unlabeled points (light gray) in Figs. 5a and 5b, compared to the coherent, strongly separated groups in the point cloud-based embedding (Fig. 5c).

This visual impression is reinforced by a quantitative analysis using the neighborhood hit [13], the average ratio of  $k$ -nearest neighbors in embedding space with the same ground truth label. Fig. 5f shows the neighborhood hit for the first 100 nearest neighbors. The point cloud distance approach yields a significantly higher hit for all  $k$ -nearest neighbors than both standard t-SNE and the Gaussian filtering approach.

Fig. 5d shows the original hand-drawn annotations overlaid on the individual fields taken from the ground-truth data. The arrow points at an aisle dividing two parts of a field that was given in the manual annotation but was lost in the ground truth. In our point cloud-based embedding, we could identify a cluster (arrow in Fig. 5c) corresponding to this aisle and an unlabeled area next to it. The yellow area in Fig. 5d indicates the pixels corresponding to that cluster, which illustrates the ability to distinguish structure, even beyond the ground truth, in the case of the embedding using the point cloud distance. Hereby, we illustrate the usefulness of combining spatial information with the full attribute space for exploration purposes. While there exist clusters in the other two embeddings that partially correspond to the aisle, they also contain pixels from areas in different regions (see Fig. 20 in Supplemental Material S6 for an example).

In summary, our point-cloud embedding outperforms the other two with respect to the exploration of spatially continuous, meaningful regions. More examples for similar behaviour, for instance that a specific field is well captured in a single cluster of the point-cloud embedding but divided between multiple clusters in the other embeddings, are shown in Fig. 21 in Supplemental Material S6.

## 7.2 Imaging mass cytometry

Imaging mass cytometry [22] is a recent imaging modality used to study cellular biology. Imaging mass cytometry simultaneously captures the expression of up to 50 different proteins in tissue by ablating tissue sections spot-by-spot. Combining the resulting measurements in a regular grid results in a high-dimensional image, where the pixel position corresponds to the position in the tissue and the channels to the different measured proteins. Visual analytics and exploratory analysis based on dimensionality reduction is used in practice for the analysis of imaging mass cytometry data, as for example presented by Somarakis et al. [55]. They present a multi-step approach where cells are segmented, followed by aggregating high-dimensional profiles per cell. These are then used to identify cell types using dimensionality reduction and the resulting classification is the basis for exploration of local neighborhoods of cells.

For this use case, we consider a tissue sample from a mammalian lung provided by collaborating researchers. The image measures  $272 \times 374$  pixels, each pixel represents a  $1 \mu\text{m}$  area, and we consider ten attribute channels, corresponding to ten different proteins describing immune cells and structural properties. Fig. 6 shows an RGB re-coloring of the sample, mapping red, green, and blue channels to one of three structural proteins each, to give an impression of the tissue layout.

Fig. 6b shows a standard t-SNE embedding and re-colored following the same re-coloring scheme as in Fig. 4. Figs. 6c and 6d show texture-aware embeddings based on the covariance matrix feature and Bhattacharyya distance. We used two different neighborhood sizes,  $3 \times 3$  in Fig. 6c and  $7 \times 7$  in Fig. 6d to show structures on different scales in the image. For both, we make use of a Gaussian spatial weighting. All embeddings were computed with a perplexity of 30 and 5,000 gradient descent iterations to ensure convergence.

Our collaborators are interested in the composition of cell structures called granuloma, indicated in Fig. 6, and their surrounding cells. A granuloma is an agglomeration of immune cells, typically to isolate irritants or foreign objects. Current analysis pipelines separate the analysis of the high-dimensional attribute data and spatial layout of the cell data [55]. Our collaborators stated that it would be useful to combine these two steps for early data exploration.

In the re-colored image in Fig. 6b we can see that the granuloma as a whole is already differentiated from the rest of the image, indicated by a bright orange area with mint green, purple, and blue inclusions. The bright orange indicates a combination of proteins, characteristic for a specific set of immune cells (macrophages) which are expected in the center of the granuloma. It is known that the area around a granuloma is made up of layers comprising of said macrophages and different combinations of other immune cells. The structure of these cell layers and interaction of cells within and between adjacent layers is subject of current research. Hints of this changing composition can be seen in Fig. 6b where the center largely consists of mint-green inclusions which slowly change to purple and blue inclusions towards the outside. A first hypothesis when analysing the given tissue was that these layers are similar all around the granuloma.

Comparing the small-scale texture-aware embedding and re-colored image in Fig. 6c, we get a similar impression with a bright-purple colored area with some inclusions defining the granuloma. Note, that the colors are not directly comparable due to the heuristic nature of t-SNE and the different structures of the embeddings. However, we already see some hints at larger scale structure. The central area of the granuloma (consisting of many blue (mint in Fig. 6b) inclusions) and the outer layers are now separately colored in a deep pink and orange, respectively, indicating separation in the embedding. Individual cells can still roughly be identified, for example the blue and purple patches within and around the structure.

Finally, using a larger neighborhood as in Fig. 6d clearly creates areas of similar color, corresponding to higher-level structures. This is expected as the neighbourhood is enlarged. Here, no individual cells are recognizable anymore. The granuloma as a whole is clearly recognizable by a pink to orange area, but in addition a clear layering structure is visible. The granuloma center is a relatively homogeneous dark pink area. Around the granuloma, we can see the layering of structures in different shades of orange to a greenish tone on the far outside, following the colormap applied to the embedding from bottom to top on the left side (arrow Fig. 6d). Upon inspection of this texture-aware embedding our collaborators were very interested in these layers surrounding the granuloma center and how clearly they were identifiable in Fig. 6d, hereby eliminating the need for a multi-step approach which was typical for their work flow. They also noted that the layer structure was more varied than they expected which they intend to study further and verify that this is indeed consistent across biological replicates.

## 8 CONCLUSION

We have presented a framework of texture-aware dimensionality reduction for visual exploration of high-dimensional images and illustrated its potential through examples based on t-SNE and three different texture-aware distance metrics. The generated embeddings combine attribute similarity with spatial context, and, thereby, support the exploration of high-dimensional images. Our method adapts the point similarity calculation of distance-based dimensionality reduction methods by taking the spatial nature of images into account. We presented two classes of approaches for comparing spatial pixel neighborhoods and extended them to high-dimensional images: Feature-based methods (FBM), extracting and comparing features of neighborhoods, and area-based methods (ABM), applying distance measures between the sets of attributes within the neighborhoods directly. We have shown strengths and weaknesses of the different approaches, illustrated them in a synthetic example and presented their applicability via two use cases.

The presented method opens several avenues for future work. We focused on images, i.e., structured, rectangular grids. Extensions to unstructured grids, common in geographic information systems, or graphs are thinkable. While we show several examples of different neighborhood sizes, we did not investigate this parameter in-depth. Using varying neighborhood sizes might reveal spatial structures that are only present at a specific scale. Another interesting avenue might be to investigate the potential of other feature extraction methods, like Markov random field texture models or neural network approaches to capture domain-specific texture characteristics when training data is available. Additionally, visualizing the extracted features alongside the embeddings is an interesting idea in itself.

In this work, we have shown that texture-aware dimensionality reduction methods can provide insights into high-dimensional images that cannot be captured with standard dimensionality reduction methods alone. Thus, we believe that they will prove to be a valuable addition to the tool box for high-dimensional data analysis.

## ACKNOWLEDGMENTS

This work received funding from the NWO TTW project 3DOMICS (NWO: 17126). We thank Paula Niewold and Simone Joosten (Leiden University Medical Center, LUMC) for providing imaging mass cytometry data and feedback. We also thank Nannan Guo, Frits Konig (LUMC) and Na Li (Jilin University) for valuable discussions.

## REFERENCES

- [1] W. M. Abdelmoula, B. Balluff, S. Englert, J. Dijkstra, M. J. Reinders, A. Walch, L. A. McDonnell, and B. P. Lelieveldt. Data-driven identification of prognostic tumor subpopulations using spatially mapped t-SNE of mass spectrometry imaging data. *Proceedings of the National Academy of Sciences of the United States of America*, 113(43):12244–12249, 2016. doi: 10.1073/pnas.1510227113
- [2] M. Alfeld, S. Pedetti, P. Martinez, and P. Walter. Joint data treatment for Vis-NIR reflectance imaging spectroscopy and XRF imaging acquired in the theban necropolis in egypt by data fusion and t-SNE. *Comptes Rendus Physique*, 19(7):625–635, 2018. doi: 10.1016/j.crhy.2018.08.004
- [3] N. Altman and M. Krzywinski. The curse(s) of dimensionality. *Nature Methods*, 15(6):399–400, 2018. doi: 10.1038/s41592-018-0019-x
- [4] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe. 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3. *Purdue University Research Repository*, 10:R7RX991C, 2015. doi: 10.4231/R7RX991C
- [5] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang. Heterogeneous Network Embedding via Deep Architectures. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015. doi: 10.1145/2783258
- [6] E. Choi and C. Lee. Feature extraction based on the bhattacharyya distance. *Pattern Recognition*, 36(8):1703–1709, 2003. doi: 10.1016/S0031-3203(03)00035-9
- [7] D. Conte, P. Foggia, C. Sansone, and M. Vento. How and why pattern recognition and computer vision applications use graphs. In A. Kandel, H. Bunke, and M. Last, eds., *Applied Graph Theory in Computer Vision and Pattern Recognition*, pp. 85–135. Springer Berlin Heidelberg, 2007. doi: 10.1007/978-3-540-68020-8\_4
- [8] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48:3747–3762, 2010. doi: 10.1109/TGRS.2010.2048116
- [9] N. L. de Vries, A. Mahfouz, F. Koning, and N. F. de Miranda. Unraveling the complexity of the cancer microenvironment with multidimensional genomic and cytometric technologiess. *Frontiers in Oncology*, 2020. doi: 10.3389/fonc.2020.01254
- [10] A. Depeursinge, O. S. Al-Kadi, and J. Ross Mitchell. *Biomedical texture analysis: Fundamentals, tools and challenges*. Elsevier, 2017. doi: 10.1016/C2016-0-01903-4
- [11] C. Eichner, A. Bittig, H. Schumann, and C. Tominski. Feature-based visual analytics for studying simulations of dynamic bi-stable spatial systems. In *EuroVis Workshop on Visual Analytics*, 2013. doi: 10.2312/PE.EuroVAST.EuroVA13.025-029
- [12] W. Equitz, H. S. Sawhney, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, 1995. doi: 10.1109/34.391417
- [13] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2153–2173, 2021. doi: 10.1109/TVCG.2019.2944182
- [14] M. Evers, K. Huesmann, and L. Linsen. Uncertainty-aware visualization of regional time series correlation in spatio-temporal ensembles. *Computer Graphics Forum*, 40(3):519–530, 2021. doi: 10.1111/CGF.14326
- [15] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2463–2471, 2017. doi: 10.1109/CVPR.2017.264
- [16] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2013. doi: 10.1109/JPROC.2012.2197589
- [17] M. M. Fischer and A. Getis, eds. *Handbook of Applied Spatial Analysis: Software Tools, Methods and Applications*. Springer Berlin Heidelberg, 1 ed., 2010. doi: 10.1007/978-3-642-03647-7
- [18] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009. doi: 10.1111/j.1467-8659.2009.01429.x
- [19] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv:1508.06576*, 2015.
- [20] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):37–78, 2017. doi: 10.1109/MGRS.2017.2762087

- [21] H. Gibson and P. Vickers. graphTPP: A multivariate based method for interactive graph layout and analysis. *arXiv: 1712.05644*, 2017.
- [22] C. Giesen, H. A. Wang, D. Schapiro, N. Zivanovic, A. Jacobs, B. Hattendorf, P. J. Schöffler, D. Grolimund, J. M. Buhmann, S. Brandt, Z. Varga, P. J. Wild, D. Günther, and B. Bodenmiller. Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry. *Nature Methods*, 11(4):417–422, 2014. doi: 10.1038/nmeth.2869
- [23] A. A. Goshtasby. Similarity and dissimilarity measures. In *Image Registration*, pp. 7–66. Springer Link, 2012. doi: 10.1007/978-1-4471-2458-0\_2
- [24] A. Halladin-Dąbrowska, A. Kania, and D. Kopeć. The t-SNE algorithm as a tool to improve the quality of reference data used in accurate mapping of heterogeneous non-forest vegetation. *Remote Sensing*, 12(1), 2020. doi: 10.3390/rs12010039
- [25] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, p. 229–238, 1995. doi: 10.1145/218380.218446
- [26] T. Höllt, N. Pezzotti, V. van Unen, N. Li, F. Koning, E. Eiseemann, B. Lelieveldt, and A. Vilanova. Cytosplere: Interactive visual single-cell profiling of the immune system. In *Proceedings of Eurographics - Short Papers and Medical Prize Awards*, 2019. doi: 10.2312/egm.20191032
- [27] T. Höllt, A. Vilanova, N. Pezzotti, B. Lelieveldt, and H. Hauser. Focus+Context exploration of hierarchical embeddings. *Computer Graphics Forum (Proceedings of EuroVis 2019)*, 38(3), 2019. doi: 10.1111/cgf.13711
- [28] H. Huang, Y. Duan, H. He, G. Shi, and F. Luo. Spatial-spectral local discriminant projection for dimensionality reduction of hyperspectral image. *ISPRS Journal of Photogrammetry and Remote Sensing*, 156:77–93, 2019. doi: 10.1016/j.isprsjprs.2019.06.018
- [29] A. Humeau-Heurtier. Texture feature extraction methods: A survey. *IEEE Access*, 7:8975–9000, 2019. doi: 10.1109/ACCESS.2018.2890743
- [30] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993. doi: 10.1109/34.232073
- [31] R. Kapoor, D. Sharma, and T. Gulati. State of the art content based image retrieval techniques using deep learning: a survey. *Multimedia Tools and Applications 2021*, pp. 1–23, 2021. doi: 10.1007/978-1-1044-021-11045-1
- [32] J. Kehrner and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, 2013. doi: 10.1109/TVCG.2012.110
- [33] B. Kenkhuis, A. Somarakis, L. de Haan, O. Dzyubachyk, M. E. Ijselsteijn, N. F. D. Miranda, B. Lelieveldt, W. M. C. van Roon-Mom, T. Höllt, and L. van der Weerd. Iron loading is a prominent feature of activated microglia in alzheimer’s disease patients. *Acta Neuropathologica Communications*, 2021. doi: 10.1186/s40478-021-01126-5
- [34] A. Kerren, H. Purchase, and M. O. Ward, eds. *Multivariate Network Visualization*. Springer, 1 ed., 2014. doi: 10.1007/978-3-319-06793-3
- [35] R. Krueger, J. Beyer, W.-D. Jang, N. W. Kim, A. Sokolov, P. K. Sorger, and H. Pfister. Facetto: Combining unsupervised and supervised learning for hierarchical phenotype analysis in multi-channel image data. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019. doi: 10.1109/tvcg.2019.2934547
- [36] S. Lefebvre and H. Hoppe. Appearance-space texture synthesis. *ACM Transactions on Graphics*, 25(3):541–548, 2006. doi: 10.1145/1141911.1141921
- [37] R. A. Leite, T. Gschwandtner, S. Miksch, S. Kriglstein, M. Pohl, E. Gstrein, and J. Kuntner. EVA: Visual analytics to identify fraudulent events. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):330–339, 2018. doi: 10.1109/TVCG.2017.2744758
- [38] C. Leng, H. Zhang, B. Li, G. Cai, Z. Pei, and L. He. Local feature descriptor for image matching: A survey. *IEEE Access*, 7:6424–6434, 2019. doi: 10.1109/ACCESS.2018.2888856
- [39] Y. Y. Leow, T. Laurent, and X. Bresson. GraphTSNE: A Visualization Technique for Graph-Structured Data. *arXiv: 1904.06915*, 2019.
- [40] S. Liu, D. Maljovec, B. Wang, P. T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, 2017. doi: 10.1109/TVCG.2016.2640960
- [41] G. Lu and B. Fei. Medical hyperspectral imaging: a review. *Journal of Biomedical Optics*, 19(1):010901, 2014. doi: 10.1117/1.jbo.19.1.010901
- [42] D. Lungu and O. Ersoy. Spherical stochastic neighbor embedding of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):857–871, 2013. doi: 10.1109/TGRS.2012.2205004
- [43] X. Luo, L. Scandolo, and E. Eiseemann. Texture browser: Feature-based texture exploration. *Computer Graphics Forum*, 40(3):99–109, 2021. doi: 10.1111/cgf.14292
- [44] H. M. Texture analysis. In *Advanced Biomedical Image Analysis*, pp. 236–275. Wiley, 2010. doi: 10.1002/9780470872093.CH8
- [45] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan. Image matching from hand-crafted to deep features: A survey. *International Journal of Computer Vision 2020 129:1*, 129(1):23–79, 2020. doi: 10.1007/S11263-020-01359-2
- [46] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473
- [47] R. M. Martins, J. F. Krüger, R. Minghim, A. C. Telea, and A. Kerren. MVN-reduce: Dimensionality reduction for the visual analysis of multivariate networks. In *EuroVis 2017 - Short Papers*, 2017. doi: 10.2312/eurovisshort.20171126
- [48] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv: 1802.03426*, 2018.
- [49] M. Meyer, T. Munzner, A. DePace, and H. Pfister. MulteeSum: A tool for comparative spatial and temporal gene expression data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):908–917, 2010. doi: 10.1109/TVCG.2010.137
- [50] C. Palm. Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5):965–976, 2004. doi: 10.1016/j.patcog.2003.09.010
- [51] N. Pezzotti, B. P. Lelieveldt, L. van der Maaten, T. Höllt, E. Eiseemann, and A. Vilanova. Approximated and user steerable tSNE for progressive visual analytics. *IEEE transactions on visualization and computer graphics*, 23(7):1739–1752, 2017. doi: 10.1109/TVCG.2016.2570755
- [52] N. Pezzotti, J. Thijssen, A. Mordvintsev, T. Höllt, B. van Lew, B. Lelieveldt, E. Eiseemann, and A. Vilanova. GPGPU linear complexity t-SNE optimization. *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE VAST 2019)*, 26(1), 2020.
- [53] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2634–2643, 2013. doi: 10.1109/TVCG.2013.153
- [54] C. Singh, E. Walia, and K. P. Kaur. Color texture description with novel local binary patterns for effective image retrieval. *Pattern Recognition*, 76:50–68, 2018. doi: 10.1016/j.patcog.2017.10.021
- [55] A. Somarakis, V. Van Unen, F. Koning, B. P. Lelieveldt, and T. Höllt. ImaCyE: visual exploration of cellular microenvironments for imaging mass cytometry data. *IEEE Transactions on Visualization and Computer Graphics*, 27:98–110, 2021. doi: 10.1109/tvcg.2019.2931299
- [56] J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th international conference on world wide web*, pp. 287–297, 2016. doi: 10.1145/2872427.2883041
- [57] L. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.
- [58] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- [59] A. Vieth. Spatial information in dimensionality reduction (spidr). <https://github.com/biovault/Spidr>, 2022. doi: 10.5281/zenodo.6120879
- [60] L. Zhou and C. Hansen. Transfer function design based on user selected samples for intuitive multivariate volume exploration. In *IEEE Pacific Visualization Symposium*, pp. 73–80, 2013. doi: 10.1109/PacificVis.2013.6596130
- [61] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003. doi: 10.1016/S0262-8856(03)00137-9

# Supplemental Material:

## Incorporating Texture Information into Dimensionality Reduction for High-Dimensional Images

A. Vieth, A. Vilanova, B. Lelieveldt, E. Eisemann, and T. Höllt

### S0: COMPUTATION SETTINGS

For all t-SNE computations we set used the following HDLib parameters: `Exaggeration=250`, `exponential decay=40`, `number of trees=4`, and `number of checks=1024`. When computing approximated nearest neighbors with HNSWlib we use the default parameters `M=16` and `ef_construction=200` as well as the random seed 0.

We apply Gaussian filtering using OpenCV [1] with the function `GaussianBlur()` and the settings `sigmaX=5` and `ksize=3`.

For the quantitative analysis, we compute the  $k$ -nearest neighbor hit, as described by Espadoto et al [2]. In brief, for labelled data, for every point in the low-dimensional embedding, we compute the fraction of the  $k$  nearest neighbors in the low-dimensional embedding have the same label as the probed point. This fraction is then averaged for all points in the dataset. For the synthetic data, we define the ground truth by separating the checkered areas and homogeneous areas as shown in Fig. 8a. For the Indian Pines dataset, we use the 16-class ground truth data provided with the original data. For the synthetic data, we limit  $k$  to  $k = [1..63]$ , as the inner, homogeneous squares in the image cover 64 pixels, meaning larger values for  $k$  would include more neighbors than pixels existing for the given label. For the Indian Pines data, we compute the  $k$ -nearest neighbor hit for  $k = [1..100]$ . For all, we do not include the probed point in the  $k$ -nearest neighbors.

### S1: WEIGHTED FEATURE COMPUTATION AND WEIGHTED CHAMFER DISTANCE

With weights  $\mathbf{w}$  that sum to 1 and weighted  $\mu^* = [\mu_1^*, \dots, \mu_C^*]$ , an entry  $\sigma_{jk}$  of the covariance matrix  $\Sigma_i$  is given by:

$$\sigma_{jk} = \sum_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \mathbf{w}(\mathbf{i} - \mathbf{q})(g_{qj} - \mu_j^*)(g_{qk} - \mu_k^*)^T. \quad (13)$$

where the weighted means are  $\mu_c^* = \sum_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \mathbf{w}(\mathbf{i} - \mathbf{q})g_{qc}$ .

For weighting the covariance matrix feature, including the channel-wise means, one only needs to introduce the weights  $\mathbf{w}$  in the calculation of the expected value as probabilities.

The Chamfer point cloud distance from Equation 12 can be extended by weighting the minimal distances from each point in the first to the second neighborhood as shown in Equation:

$$d_s^{PC}(\mathbf{T}_i, \mathbf{T}_j) = \frac{1}{|\mathcal{N}_i^{S,\eta}|} \sum_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} w(\mathbf{i} - \mathbf{q}) \min_{\mathbf{p} \in \mathcal{N}_j^{S,\eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 + \frac{1}{|\mathcal{N}_j^{S,\eta}|} \sum_{\mathbf{p} \in \mathcal{N}_j^{S,\eta}} w(\mathbf{j} - \mathbf{p}) \min_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2. \quad (14)$$

### S2: TEXTURE-AWARE UMAP AND MDS EMBEDDINGS

It is possible to use the spatially informed distances between image patches of high-dimensional images in any distance-based dimensionality reduction method. Here, we show spatially informed UMAP and metric MDS embeddings for the synthetic data set from Sect. 5.

We use the `umap-learn` [3] implementation for UMAP and `scikit-learn` [4] metric MDS. Note, that in the MDS Bhattacharyya example, the central cluster is actually two: the upper part corresponds to the upper left area in the image and the lower part to the lower right. Between the two clusters are the border points between the checkered regions (and the pixels on the vertical border between the homogeneous areas).

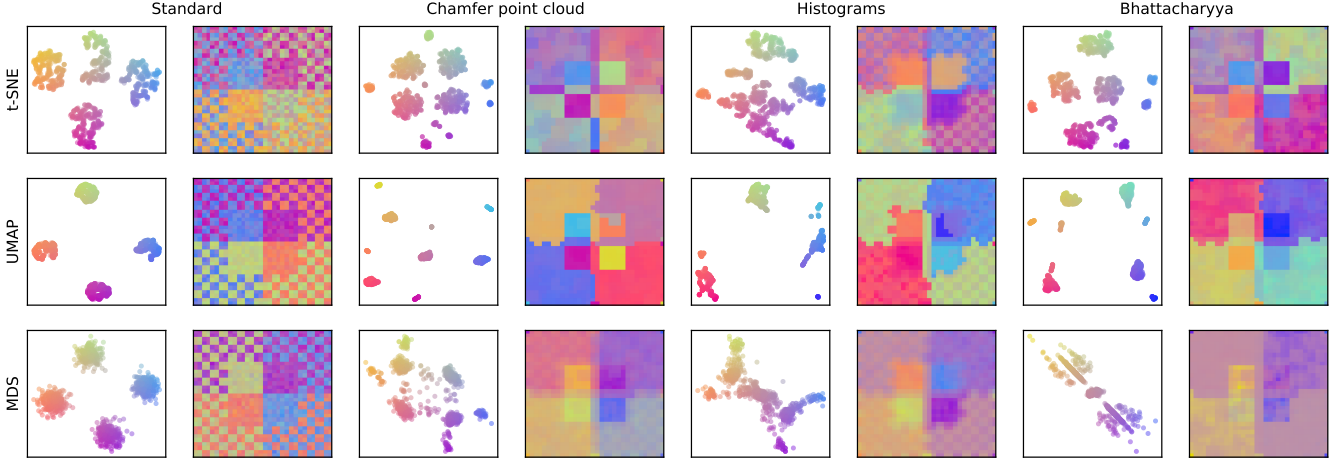


Figure 7: Spatially informed and standard embeddings with t-SNE, UMAP and MDS. Embeddings and recolored images as described in Fig. 4 in the main paper.

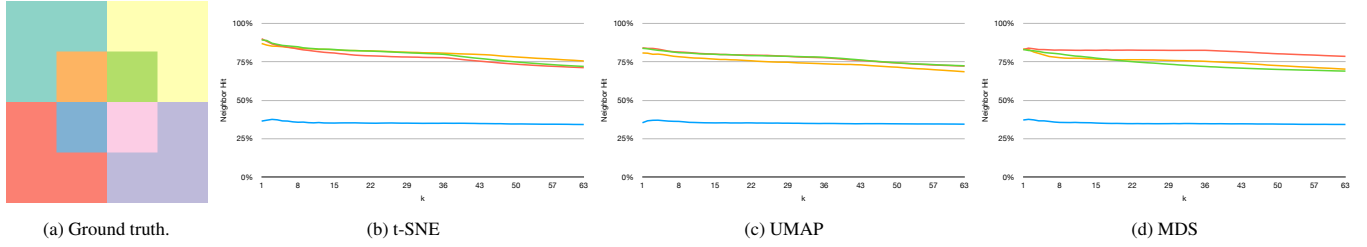


Figure 8: Ground truth for the synthetic dataset, classifying the four checkered areas on the outside and the four homogeneous regions in the center (a). Nearest neighbor hit for the t-SNE (b), UMAP (c), and MDS (d) embeddings from Fig. 7. — standard version, — covariance matrix and mean, — point cloud distance, — histogram.

### S3: OTHER POINT CLOUD DISTANCES

The Chamfer point cloud distance, see Equation 12, belongs to the broader family of distances related to the Hausdorff distance. These distances build on finding the nearest point for each point in one set to the other. Instead of averaging the minima, the Hausdorff distance takes their maximum instead:

$$d_{Haus}^{PC}(\mathbf{T}_i, \mathbf{T}_j) = \max \left\{ \max_{\mathbf{q} \in \mathcal{N}_i^{S, \eta}} \left( \min_{\mathbf{p} \in \mathcal{N}_j^{S, \eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 \right), \max_{\mathbf{q} \in \mathcal{N}_j^{S, \eta}} \left( \min_{\mathbf{p} \in \mathcal{N}_i^{S, \eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 \right) \right\} \quad (15)$$

Another variant that might be more robust against outliers in the data might take the median, instead of the average, like the Chamfer distance:

$$d_{HM}^{PC}(\mathbf{T}_i, \mathbf{T}_j) = \frac{1}{2} \left( \text{median}_{\mathbf{q} \in \mathcal{N}_i^{S, \eta}} \left( \min_{\mathbf{p} \in \mathcal{N}_j^{S, \eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 \right) + \text{median}_{\mathbf{q} \in \mathcal{N}_j^{S, \eta}} \left( \min_{\mathbf{p} \in \mathcal{N}_i^{S, \eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 \right) \right) \quad (16)$$

Sum of squared differences: When taking the average instead of the minimum of point-wise distances in the Chamfer distance, ending up an average of averages.

$$\begin{aligned} d_{SSD}^{PC}(\mathbf{T}_i, \mathbf{T}_j) &= \frac{1}{|\mathcal{N}_i^{S, \eta}|} \sum_{\mathbf{q} \in \mathcal{N}_i^{S, \eta}} \frac{1}{|\mathcal{N}_j^{S, \eta}|} \sum_{\mathbf{p} \in \mathcal{N}_j^{S, \eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 + \frac{1}{|\mathcal{N}_j^{S, \eta}|} \sum_{\mathbf{q} \in \mathcal{N}_j^{S, \eta}} \frac{1}{|\mathcal{N}_i^{S, \eta}|} \sum_{\mathbf{p} \in \mathcal{N}_i^{S, \eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 \\ &= \frac{2}{|\mathcal{N}_i^{S, \eta}| \cdot |\mathcal{N}_j^{S, \eta}|} \sum_{\mathbf{q} \in \mathcal{N}_i^{S, \eta}} \sum_{\mathbf{p} \in \mathcal{N}_j^{S, \eta}} \|\mathbf{g}_q - \mathbf{g}_p\|_2^2 \end{aligned} \quad (17)$$

Weighted versions analogous to Sect. 4.6:

$$d_{Haus}^{PC}(\mathbf{T}_i, \mathbf{T}_j) = \max \left\{ \max_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \left( w_{\mathbf{q}} \min_{\mathbf{p} \in \mathcal{N}_j^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right), \max_{\mathbf{q} \in \mathcal{N}_j^{S,\eta}} \left( w_{\mathbf{q}} \min_{\mathbf{p} \in \mathcal{N}_i^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right) \right\} \quad (18)$$

$$d_{HM}^{PC}(\mathbf{T}_i, \mathbf{T}_j) = \frac{1}{2} \left( \text{median}_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \left( w_{\mathbf{q}} \min_{\mathbf{p} \in \mathcal{N}_j^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right) + \text{median}_{\mathbf{q} \in \mathcal{N}_j^{S,\eta}} \left( w_{\mathbf{q}} \min_{\mathbf{p} \in \mathcal{N}_i^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right) \right) \quad (19)$$

$$d_{SSD}^{PC}(\mathbf{T}_i, \mathbf{T}_j) = \frac{2}{|\mathcal{N}_i^{S,\eta}| \cdot |\mathcal{N}_j^{S,\eta}|} \sum_{\mathbf{q} \in \mathcal{N}_i^{S,\eta}} \sum_{\mathbf{p} \in \mathcal{N}_j^{S,\eta}} (w_{\mathbf{q}} + w_{\mathbf{p}}) \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \quad (20)$$

See Figs. 9 and 10 for a comparison of these point cloud distances for the synthetic data set from the main paper.

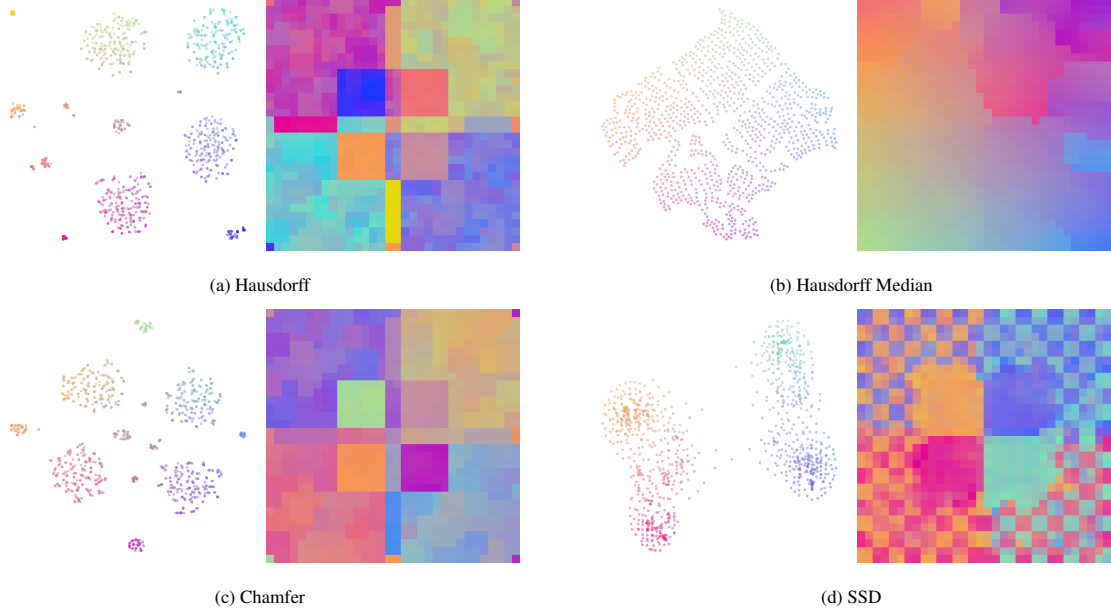


Figure 9: All spatially informed embeddings are computed with a 3x3 neighbourhood.

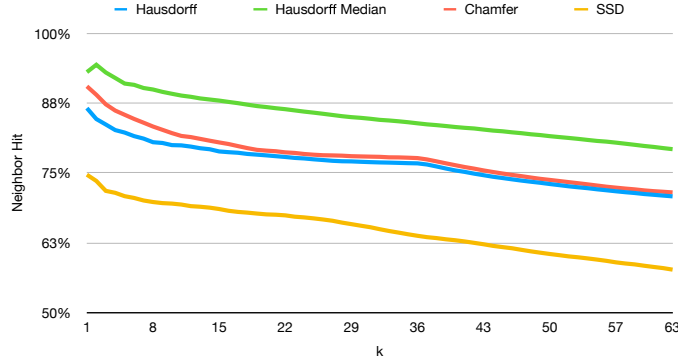


Figure 10: Nearest neighbor hit for the four embeddings in Fig. 9

#### S4: Varying neighborhood sizes and spatial weighting

See Figs. 11, 12, 13, 14 and 15 for an overview of the effect of different neighborhood sizes for the synthetic data set from the main paper.

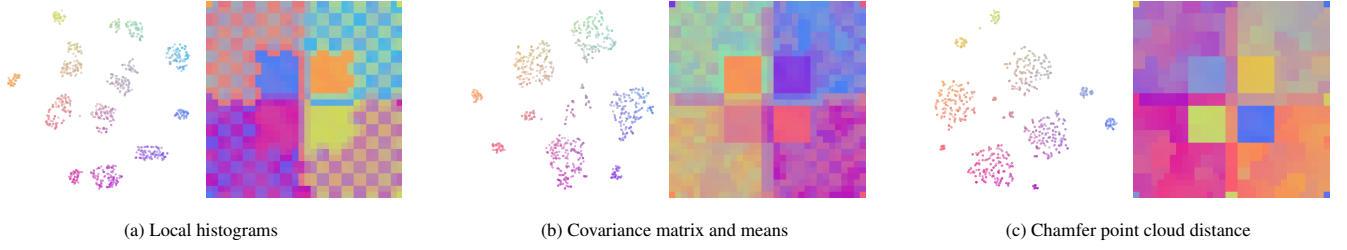


Figure 11: All spatially informed embeddings are computed with a 3x3 neighbourhood and Gaussian weighting.

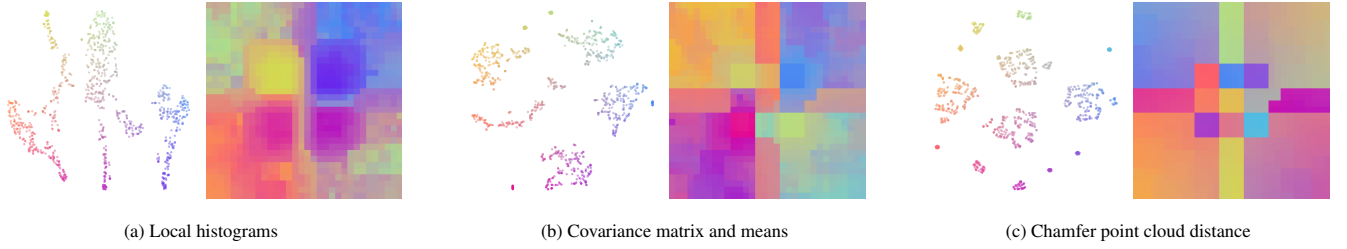


Figure 12: All spatially informed embeddings are computed with a 5x5 neighbourhood.

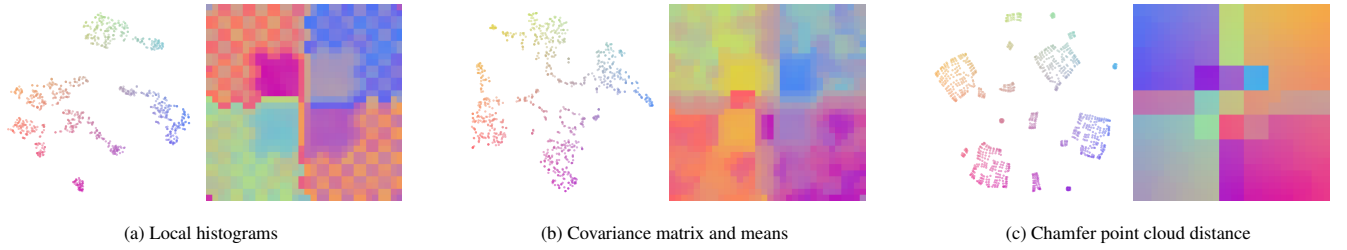


Figure 13: All spatially informed embeddings are computed with a 5x5 neighbourhood and Gaussian weighting.

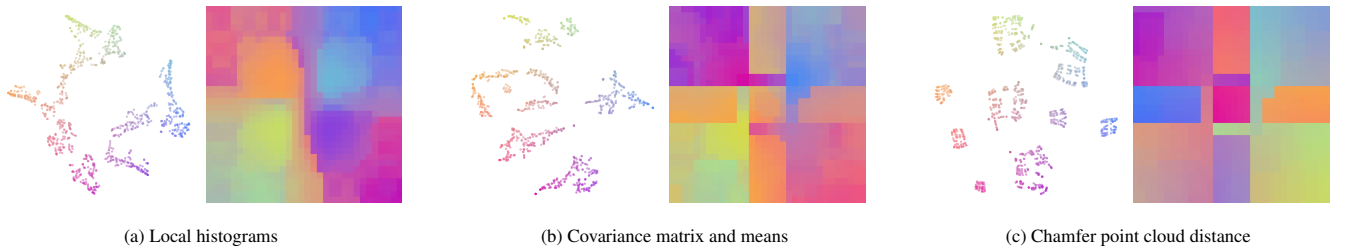


Figure 14: All spatially informed embeddings are computed with a 7x7 neighbourhood.

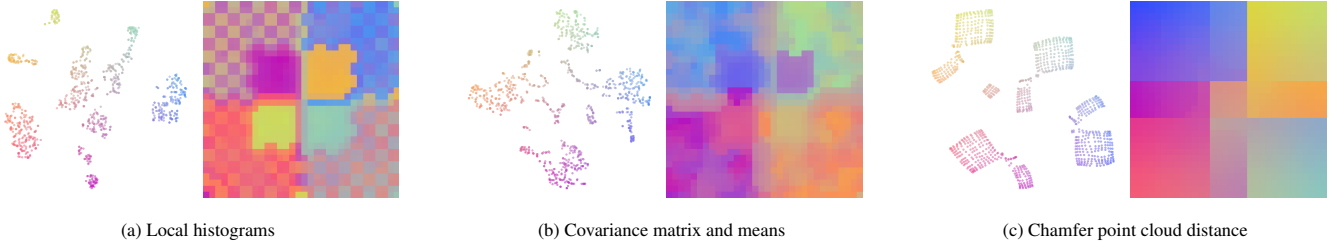


Figure 15: All spatially informed embeddings are computed with a 7x7 neighbourhood and Gaussian weighting.

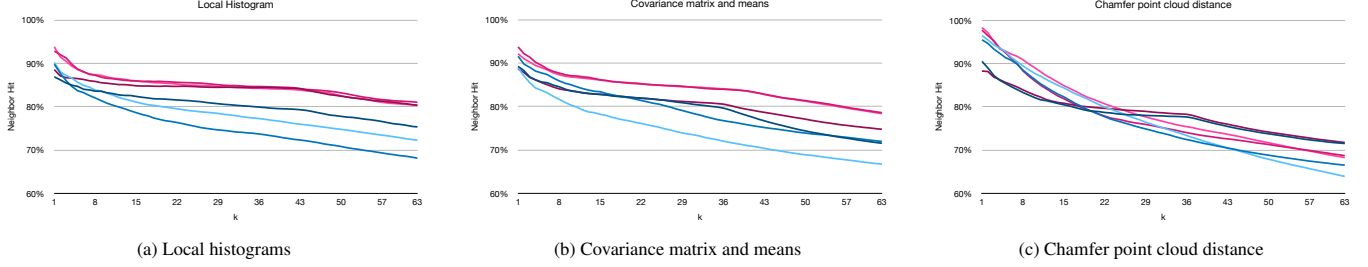


Figure 16: Neighbor hit values for the local histogram (Fig. 4b), covariance (Fig. 4c), and point cloud (Fig. 4d) -based embeddings and their different neighborhood size versions (Figs. 11, 12, 13, 14 and 15). — 3x3 neighborhood, — 5x5 neighborhood, — 7x7 neighborhood, — 3x3 neighborhood Gaussian weighted, — 5x5 neighborhood Gaussian weighted, — 7x7 neighborhood Gaussian weighted.

## S5: Computation time evaluation

Figs. 17 and 18 show the computation time for the distance computation (including feature computation) and subsequent embedding time. All measurements were conducted on a computer with an Intel i5-9600K processor and a NVIDIA GeForce RTX 2080 SUPER graphics cards. A corresponding theoretical complexity analysis of each distance is presented in Sect. 4.5. Some measurements show the influence of hardware optimizations implemented in the used libraries, which influences the computation time, see for example the time behaviour of the Bhattacharyya distance for various neighborhood sizes in Fig. 17.

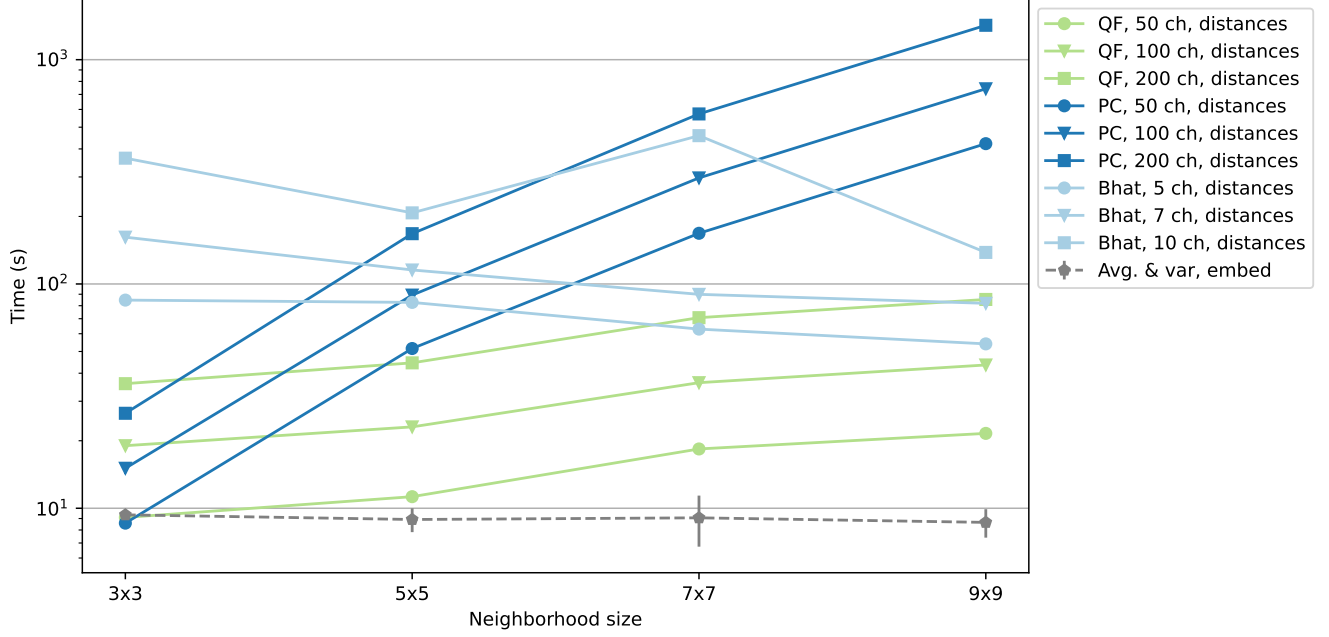


Figure 17: Computation time of the distance computation for varying neighborhood sizes: local histogram comparison with the quadratic form distance (QF), covariance matrix feature comparison with the Bhattacharyya distance (Bat) and the Chamfer point cloud distance (PC). The Indian Pines data set with 21.025 data points and 200 channels was used for computation. The same random channel subsets were used for runs with less than 200 channels. The Bhattacharyya distance is listed for fewer channels since its runtime grows impractically large for higher channel counts, as shown in Fig. 18. As mentioned in the main paper, when using the QF distance we use the Rice rule to set the number of histogram bins. This results in 5, 6, 8 and 9 bins for the various neighborhood sizes respectively. The embedding time is not influenced by the distance metric and shown as an average of all measurements with variance bars.

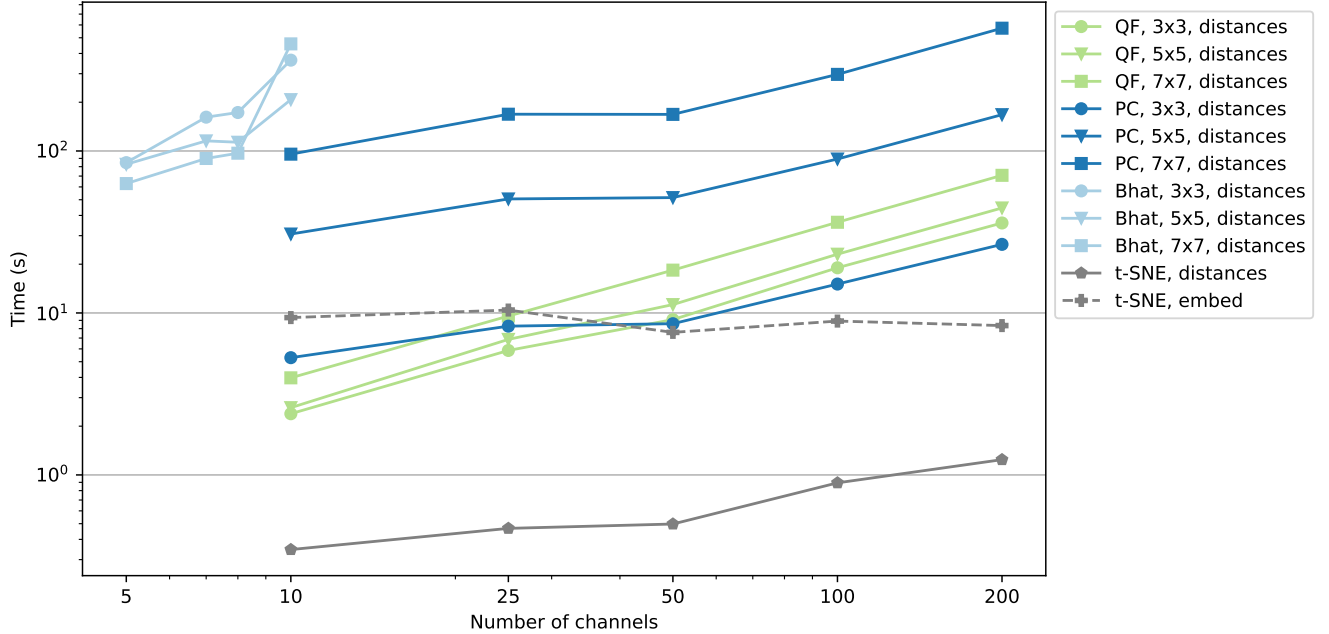


Figure 18: Computation time of the distance computation for channel numbers: local histogram comparison with the quadratic form distance (QF), covariance matrix feature comparison with the Bhattacharyya distance (Bat) and the Chamfer point cloud distance (PC). The Indian Pines data set with 21.025 data points and 200 channels was used for computation. The same random channel subsets were used for runs with less than 200 channels. The Bhattacharyya distance is listed for fewer channels since its runtime grows impractically large for higher channel counts. The histogram bin number for the QF distance is set as described in Fig. 17. The embedding time is not influenced by the distance metric and the shown embedding times for the standard t-SNE procedure is representative for the embeddings times of all runs.

## S6: Indian Pines - additional figures

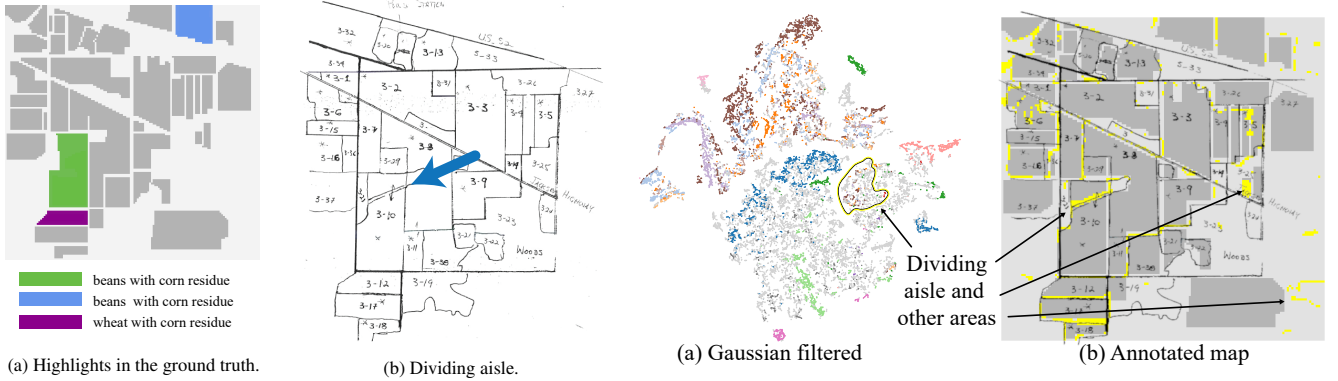


Figure 19: Ground truth of Indian Pines Site 3 with highlighted areas as further discussed in Fig. 21 and an annotated map of the Site without a color-coded background.

Figure 20: Attempted selection of the dividing aisle in the t-SNE embedding based on the Gaussian filtered data. In contrast to Fig. 5d, it is not possible to find a cluster that only corresponds to the aisle.

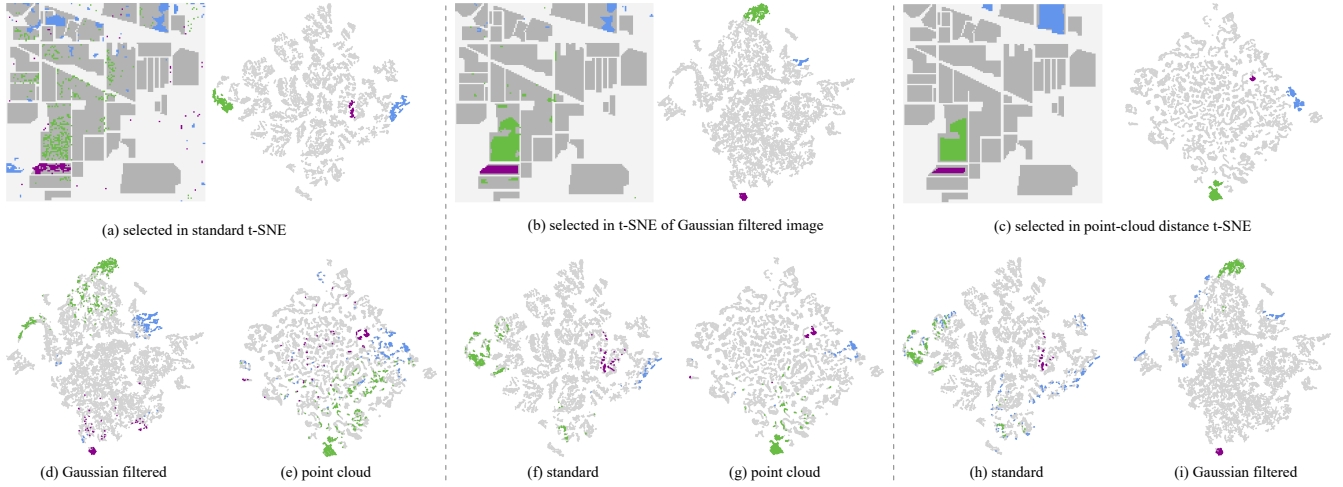


Figure 21: **Indian Pines: Comparison of spatially-aware and standard t-SNE embeddings.** The top row (a-c) shows a standard t-SNE embedding, an embedding of the Gaussian filtered data set and our spatially-aware embedding (based on the Chamfer point cloud distance). We tried to select the three highlighted regions from Fig. 19a in each embedding and show the respective pixel on the ground truth. The lower row (d-i) highlights the selections made in the above embeddings in the two other embeddings, for example highlighted points in (d) and (e) correspond to points selected in (a).

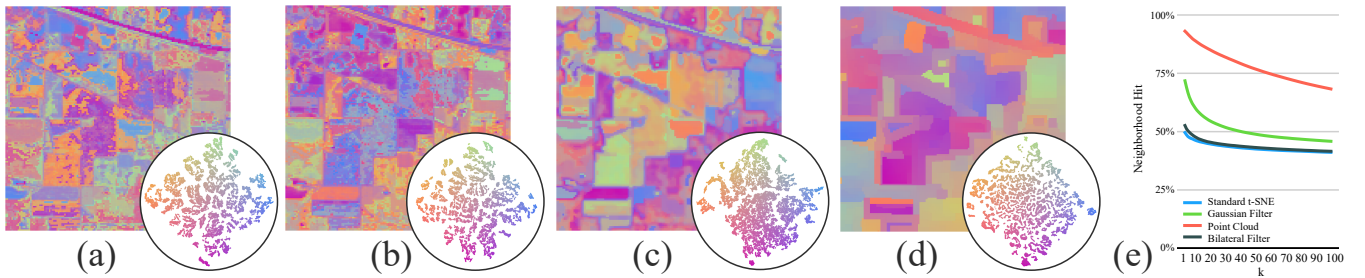


Figure 22: **Overview of the Indian Pines dataset** with different embedding methods. Coloring based on colormapping embedding coordinates, as discussed for Fig. 4. (a) standard t-SNE, (b) standard t-SNE applied to a bilaterally filtered version of the image, (c) same with a Gaussian filter, (d) our point cloud-based t-SNE, and (e) k-nearest neighbors in embedding space as in Fig. 5f with values for (b) added.

## **SUPPLEMENTAL REFERENCES**

- [1] G. Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [2] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea. Toward a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2153–2173, 2021. doi: 10.1109/TVCG.2019.2944182
- [3] L. McInnes, J. Healy, N. Saul, and L. Grossberger. UMAP: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.
- [4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.