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 σ_{jk} of the covariance matric $\mathbf{\Sigma_i}$ is given by:

$$\sigma_{jk} = \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} \mathbf{w}(\mathbf{i}-\mathbf{q})(g_{\mathbf{q}j}-\mu_j^*)(g_{\mathbf{q}k}-\mu_k^*)^T. \tag{13}$$

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

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:

$$\begin{aligned} d_s^{PC}(\boldsymbol{T}_{\mathbf{i}}, \boldsymbol{T}_{\mathbf{j}}) = & \frac{1}{|\mathcal{N}_{\mathbf{i}}^{S,\eta}|} \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} w(\mathbf{i}-\mathbf{q}) \min_{\mathbf{p}\in\mathcal{N}_{\mathbf{j}}^{S,\eta}} \|\mathbf{g_q}-\mathbf{g_p}\|_2^2 + \\ & \frac{1}{|\mathcal{N}_{\mathbf{j}}^{S,\eta}|} \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{j}}^{S,\eta}} w(\mathbf{j}-\mathbf{p}) \min_{\mathbf{q}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} \|\mathbf{g_q}-\mathbf{g_p}\|_2^2. \end{aligned} \tag{14}$$

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).

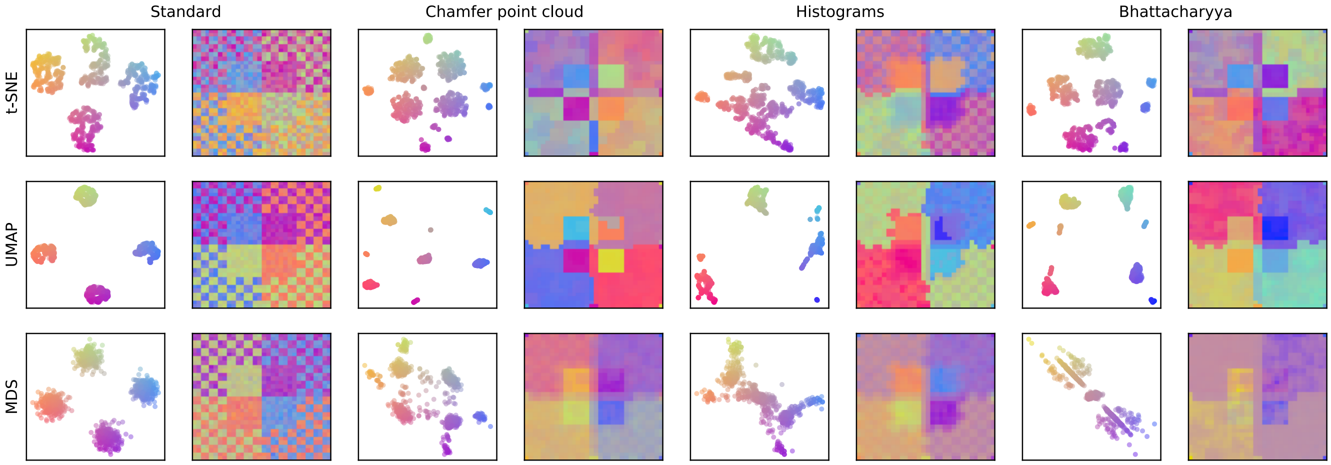
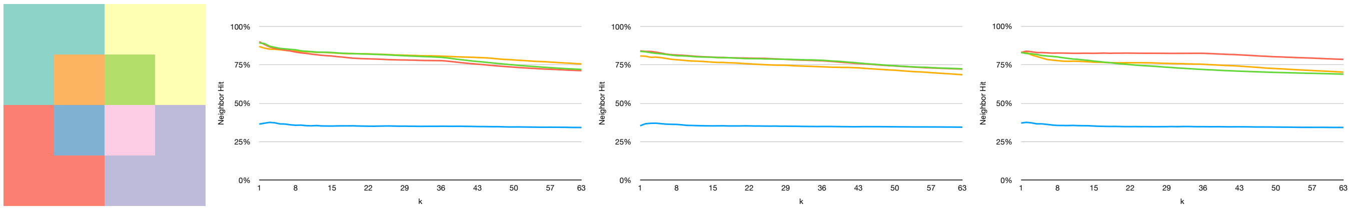


Figure 7: Spatially informed and standard embeddings with t-SNE, UMAP and MDS. Embeddings and recolored images as described in Fig. 4 in the main paper.



(a) Ground truth. (b) t-SNE (c) UMAP (d) MDS

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:

$$d^{PC}_{Haus}(\boldsymbol{T}_{\mathbf{i}},\boldsymbol{T}_{\mathbf{j}}) = \max\left\{\max_{\mathbf{q}\in\mathscr{N}^{S,\eta}_{\mathbf{i}}}\left(\min_{\mathbf{p}\in\mathscr{N}^{S,\eta}_{\mathbf{j}}}\|\mathbf{g}_{\mathbf{q}}-\mathbf{g}_{\mathbf{p}}\|^2_2\right), \max_{\mathbf{q}\in\mathscr{N}^{S,\eta}_{\mathbf{j}}}\left(\min_{\mathbf{p}\in\mathscr{N}^{S,\eta}_{\mathbf{i}}}\|\mathbf{g}_{\mathbf{q}}-\mathbf{g}_{\mathbf{p}}\|^2_2\right)\right\} \tag{15}$$

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

$$d^{PC}_{HM}(\boldsymbol{T}_{\mathbf{i}},\boldsymbol{T}_{\mathbf{j}}) = \frac{1}{2}\left(\operatorname*{median}_{\mathbf{q}\in\mathscr{N}^{S,\eta}_{\mathbf{i}}}\left(\min_{\mathbf{p}\in\mathscr{N}^{S,\eta}_{\mathbf{j}}}\|\mathbf{g}_{\mathbf{q}}-\mathbf{g}_{\mathbf{p}}\|^2_2\right) + \operatorname*{median}_{\mathbf{q}\in\mathscr{N}^{S,\eta}_{\mathbf{j}}}\left(\min_{\mathbf{p}\in\mathscr{N}^{S,\eta}_{\mathbf{i}}}\|\mathbf{g}_{\mathbf{q}}-\mathbf{g}_{\mathbf{p}}\|^2_2\right)\right) \tag{16}$$

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.

$$\begin{aligned} d^{PC}_{SSD}(\boldsymbol{T}_{\mathbf{i}},\boldsymbol{T}_{\mathbf{j}}) &= \frac{1}{|\mathscr{N}^{S,\eta}_{\mathbf{i}}|}\sum_{\mathbf{q}\in\mathscr{N}^{S,\eta}_{\mathbf{i}}}\frac{1}{|\mathscr{N}^{S,\eta}_{\mathbf{j}}|}\sum_{\mathbf{p}\in\mathscr{N}^{S,\eta}_{\mathbf{j}}}\|\mathbf{g}_{\mathbf{q}}-\mathbf{g}_{\mathbf{p}}\|^2_2 + \frac{1}{|\mathscr{N}^{S,\eta}_{\mathbf{j}}|}\sum_{\mathbf{q}\in\mathscr{N}^{S,\eta}_{\mathbf{j}}}\frac{1}{|\mathscr{N}^{S,\eta}_{\mathbf{i}}|}\sum_{\mathbf{p}\in\mathscr{N}^{S,\eta}_{\mathbf{i}}}\|\mathbf{g}_{\mathbf{q}}-\mathbf{g}_{\mathbf{p}}\|^2_2 \\ &= \frac{2}{|\mathscr{N}^{S,\eta}_{\mathbf{i}}|\cdot|\mathscr{N}^{S,\eta}_{\mathbf{j}}|}\sum_{\mathbf{q}\in\mathscr{N}^{S,\eta}_{\mathbf{j}}}\sum_{\mathbf{p}\in\mathscr{N}^{S,\eta}_{\mathbf{i}}}\|\mathbf{g}_{\mathbf{q}}-\mathbf{g}_{\mathbf{p}}\|^2_2 \end{aligned} \tag{17}$$

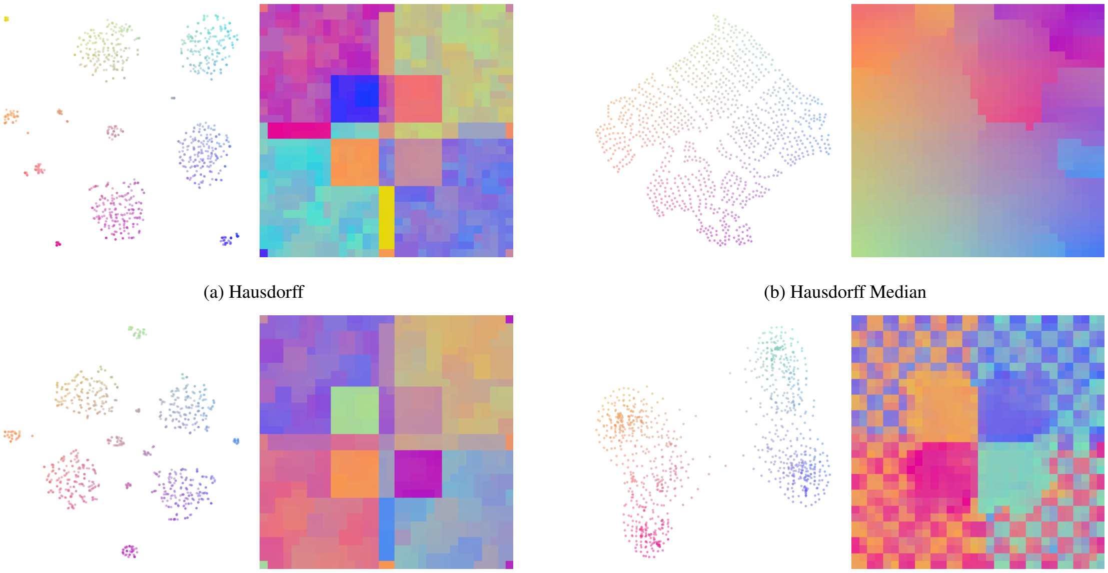
Weighted versions analogous to Sect. 4.6:

$$d^{PC}_{Haus}(\boldsymbol{T}_{\mathbf{i}},\boldsymbol{T}_{\mathbf{j}}) = \max\left\{ \max_{\mathbf{q}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} \left(w_{\mathbf{q}} \min_{\mathbf{p}\in\mathcal{N}_{\mathbf{j}}^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right), \max_{\mathbf{q}\in\mathcal{N}_{\mathbf{j}}^{S,\eta}} \left(w_{\mathbf{q}} \min_{\mathbf{p}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right) \right\} \tag{18}$$

$$d^{PC}_{HM}(\boldsymbol{T}_{\mathbf{i}},\boldsymbol{T}_{\mathbf{j}}) = \frac{1}{2}\left(\operatorname*{median}_{\mathbf{q}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} \left(w_{\mathbf{q}} \min_{\mathbf{p}\in\mathcal{N}_{\mathbf{j}}^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right) + \operatorname*{median}_{\mathbf{q}\in\mathcal{N}_{\mathbf{j}}^{S,\eta}} \left(w_{\mathbf{q}} \min_{\mathbf{p}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \right) \right) \tag{19}$$

$$d^{PC}_{SSD}(\boldsymbol{T}_{\mathbf{i}},\boldsymbol{T}_{\mathbf{j}}) = \frac{2}{|\mathcal{N}_{\mathbf{i}}^{S,\eta}| \cdot |\mathcal{N}_{\mathbf{j}}^{S,\eta}|} \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{j}}^{S,\eta}} \sum_{\mathbf{p}\in\mathcal{N}_{\mathbf{i}}^{S,\eta}} (w_{\mathbf{q}} + w_{\mathbf{p}}) \|\mathbf{g}_{\mathbf{q}} - \mathbf{g}_{\mathbf{p}}\|_2^2 \tag{20}$$

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



(a) Hausdorff

(b) Hausdorff Median

(c) Chamfer

(d) SSD

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

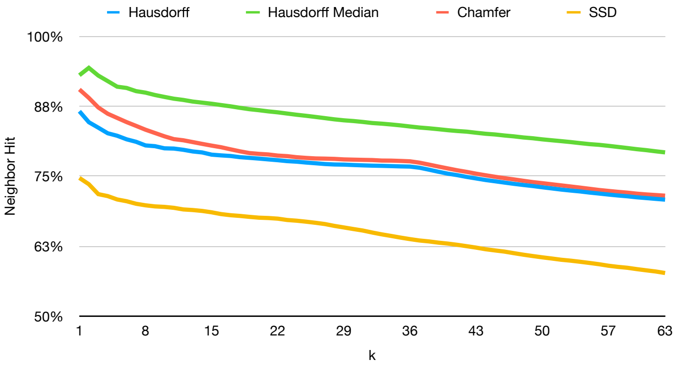
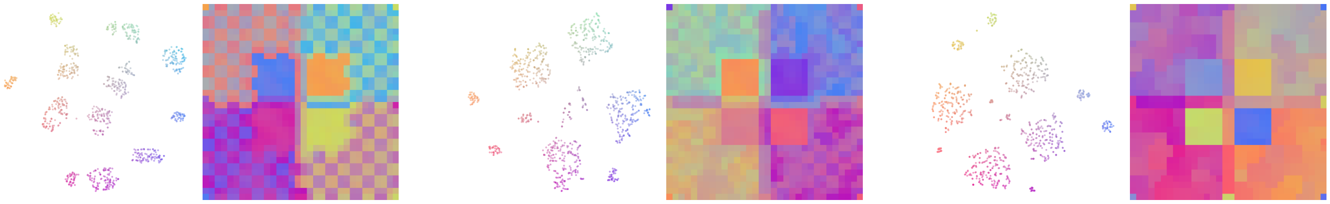


Figure 10: Nearest neighbor hit for the four embeddings in Fig. 9

S4: Varying neighborhood sizes and spatial weighting

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

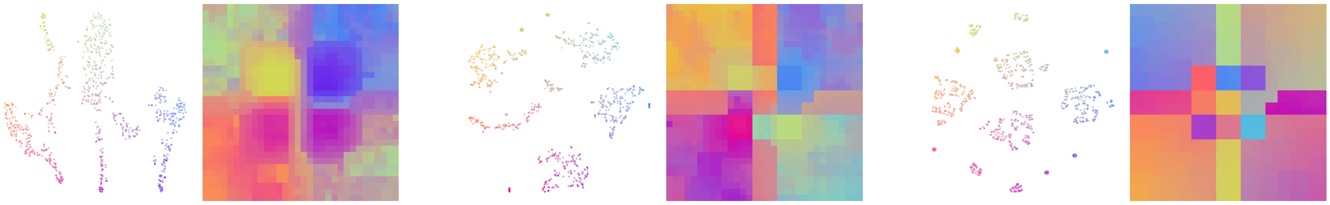


(a) Local histograms

(b) Covariance matrix and means

(c) Chamfer point cloud distance

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

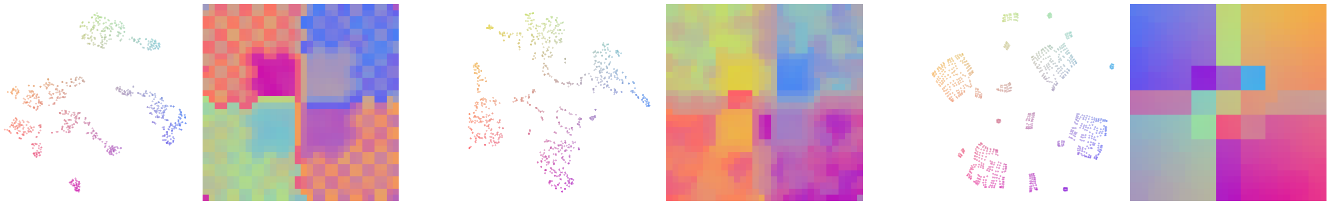


(a) Local histograms

(b) Covariance matrix and means

(c) Chamfer point cloud distance

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

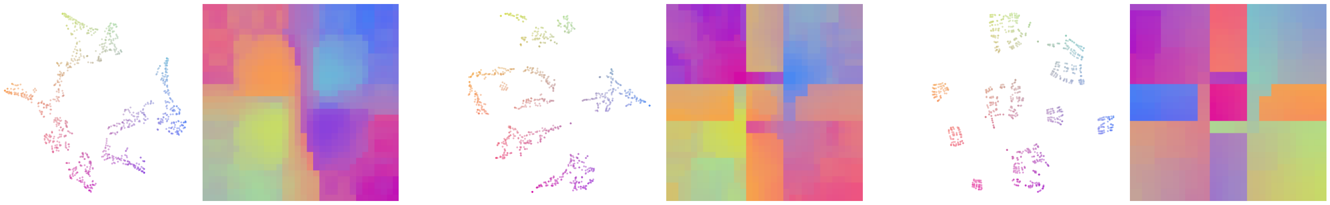


(a) Local histograms

(b) Covariance matrix and means

(c) Chamfer point cloud distance

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

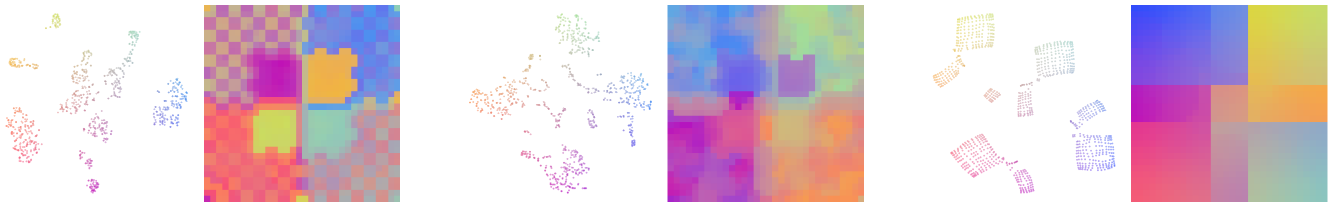


(a) Local histograms

(b) Covariance matrix and means

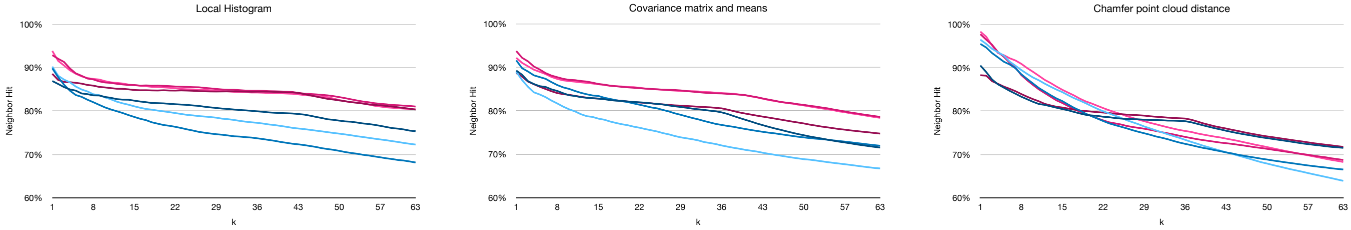
(c) Chamfer point cloud distance

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



(a) Local histograms (b) Covariance matrix and means (c) Chamfer point cloud distance

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



(a) Local histograms (b) Covariance matrix and means (c) Chamfer point cloud distance

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.

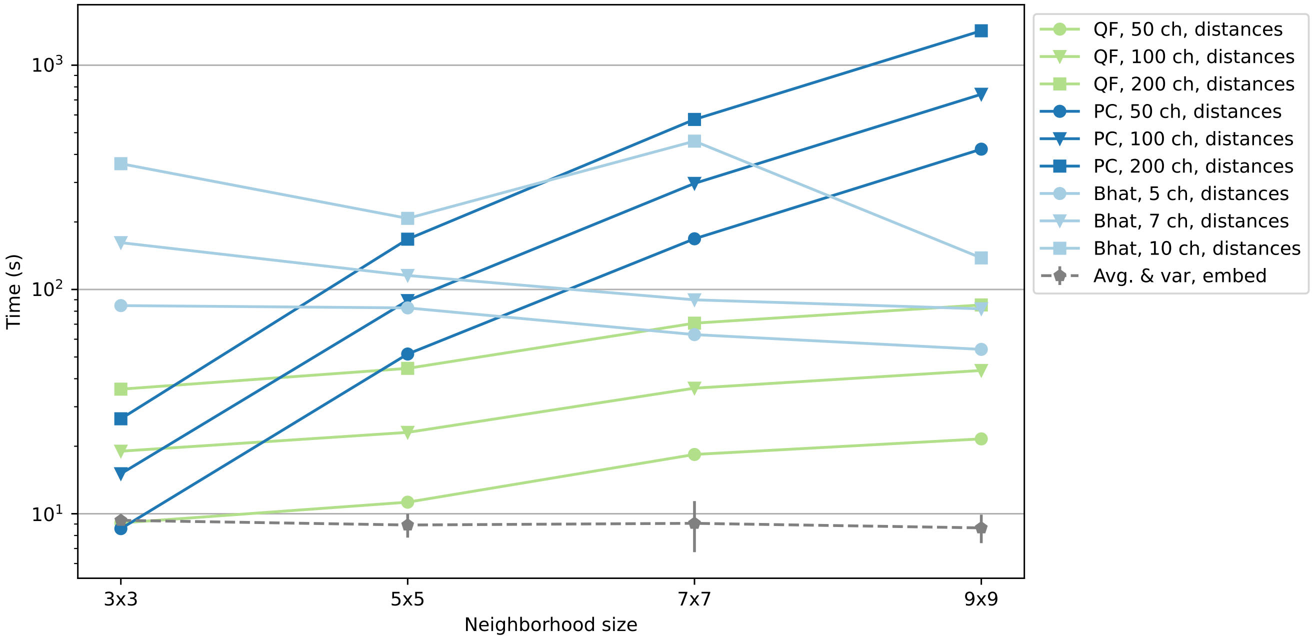


Figure 17: Computation time of the distance computation for varying neighborhood sizes: local histogram comparison with the quadratic form distance (QF), covariance matrix feature comparison with the Bhattacharyya distance (Bat) and the Chamfer point cloud distance (PC). The Indian Pines data set with 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.

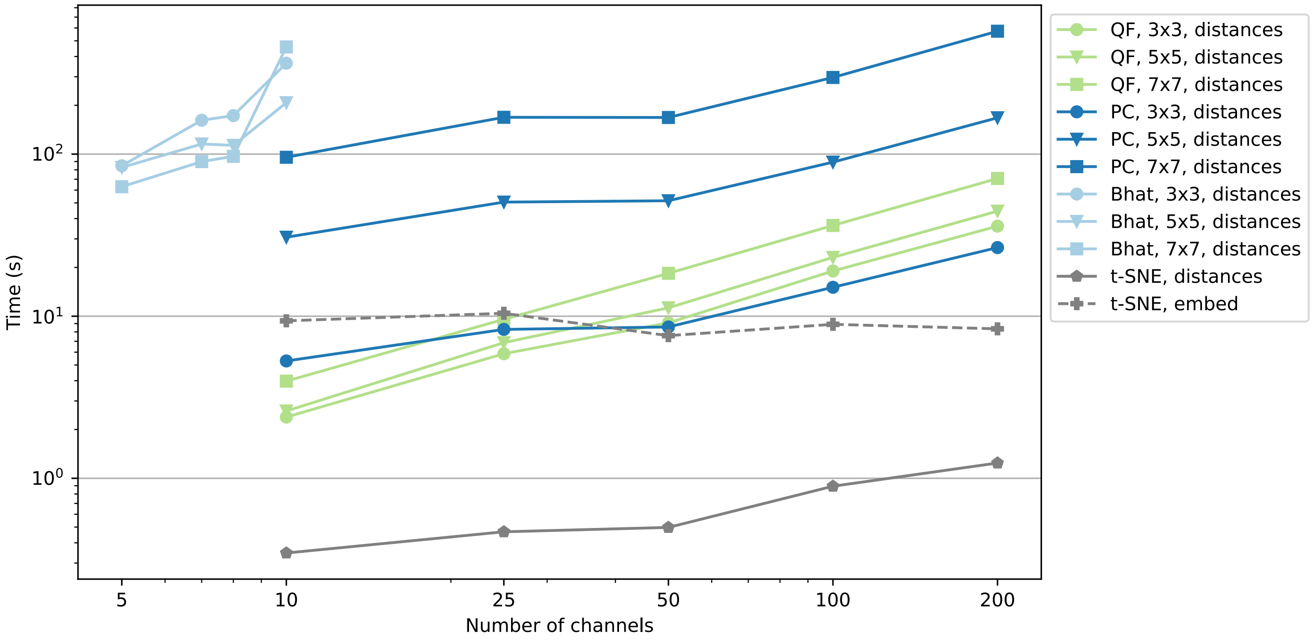
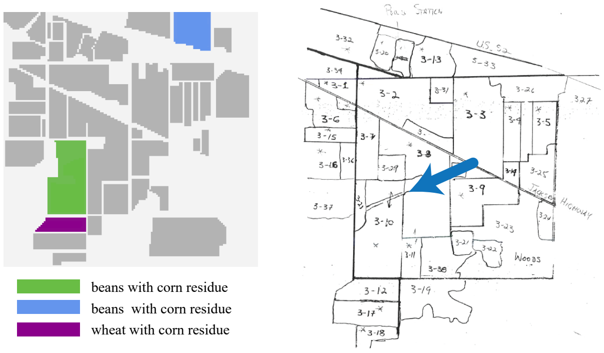


Figure 18: Computation time of the distance computation for channel numbers: local histogram comparison with the quadratic form distance (QF), covariance matrix feature comparison with the Bhattacharyya distance (Bat) and the Chamfer point cloud distance (PC). The Indian Pines data set with 21.025 data 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 influences by the distance metric and the shown embedding times for the standard t-SNE procedure is representative for the embeddings times of all runs.

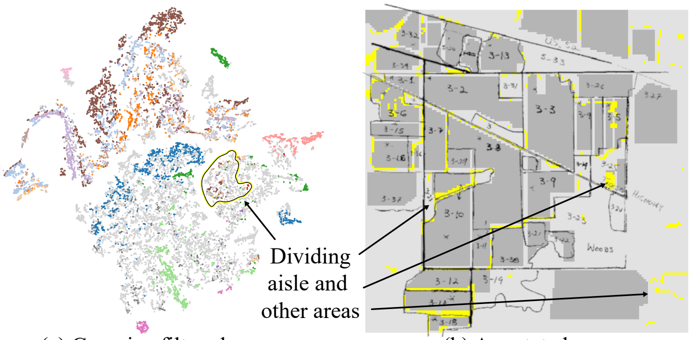
S6: Indian Pines - additional figures



(a) Highlights in the ground truth.

(b) Dividing aisle.

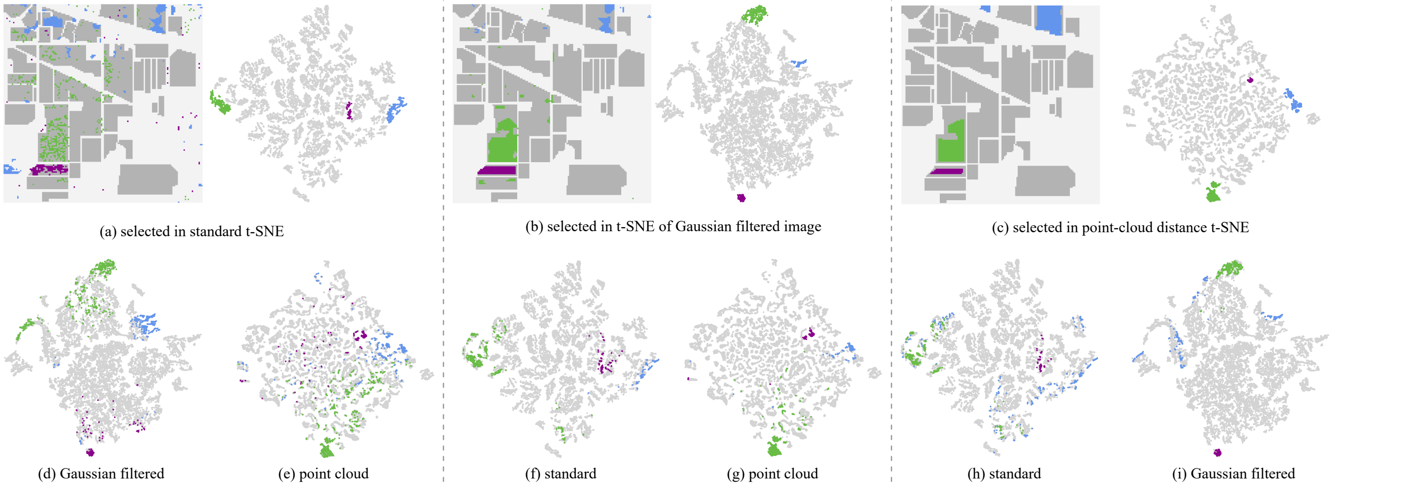
Figure 19: Ground truth of Indian Pines Site 3 with highlighted areas as further discussed in Fig. 21 and an annotated map of the Site without a color-coded background.



(a) Gaussian filtered

(b) Annotated map

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.



(a) selected in standard t-SNE

(b) selected in t-SNE of Gaussian filtered image

(c) selected in point-cloud distance t-SNE

(d) Gaussian filtered

(e) point cloud

(f) standard

(g) point cloud

(h) standard

(i) Gaussian filtered

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).

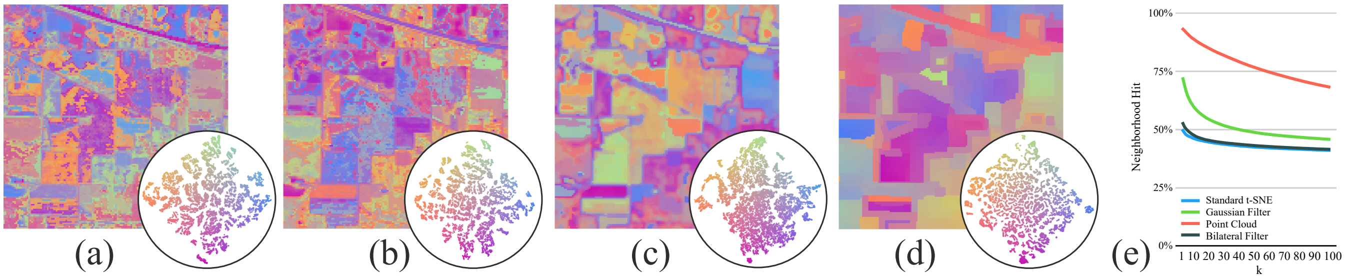


Figure 22: **Overview of the Indian Pines dataset** with different embedding methods. Coloring based on colormapping embedding coordinates, as discussed for Fig. 4. (a) standard t-SNE, (b) standard t-SNE applied to a bilaterally filtered version of the image, (c) same with a Gaussian filter, (d) our 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.