Immersive Interactive SAR Image Representation Using Non-negative Matrix Factorization

Mohammadreza Babaee, *Student Member, IEEE*, Xuejie Yu, Gerhard Rigoll, *Senior Member, IEEE*, and Mihai Datcu, *Fellow Member, IEEE*

Abstract—Earth Observation (EO) images clustering is a challenging problem in data mining, where each image is represented by a high-dimensional feature vector. However, the feature vectors might not be appropriate to express the semantic content of images, which eventually lead to poor results in clustering and classification. To tackle this problem, we propose an interactive approach to generate compact and informative features from images content. To this end, we utilize a 3D interactive application to support user-images interactions. These interactions are used in the context of two novel Non-negative Matrix Factorization (NMF) algorithms to generate new features. We assess the quality of new features by applying k-means clustering on the generated features and compare the obtained clustering results with those achieved by original features. We perform experiments on a Synthetic Aperture Radar (SAR) image dataset represented by different state-of-the-art features and demonstrate the effectiveness of the proposed method. Moreover, we propose a divide-and-conquer approach to cluster a massive amount of images using a small subset of interactions.

Index Terms—Immersive interactive system, non-negative matrix factorization, feature learning, clustering.

I. INTRODUCTION

THE amount of collected Earth Observation (EO) images has been increasing exponentially since the last decade. In order to index and retrieve this massive amount of images, the content of each image is represented by a high-dimensional feature vector. Generally, the similarity between two images is expressed by the distance in feature space, which could be different from human understanding. Hence, the gap between the human and machine description of input image, the so-called *semantic gap*, is a crucial issue in clustering and classification [1]. Numerous methods have been proposed to bridge the semantic gap in Content Based Image Retrieval (CBIR) systems [2], [3]. Basically, the related work can be categorized into two main groups. First, there are methods utilizing user-provided constraints to learn a distance metric [4]–[6]. The goal of these methods is to learn a kernel matrix that approximates the semantic distance within the Euclidean feature space. Second, there are methods that use constraints to guide a clustering algorithm such as [7], [8]. For instance, in constrained k-means, the user-provided constraints are considered in assigning images to the clusters. If one assignment violates the constraints, it will be ignored.

M. Babaee, X. Yu and G. Rigoll are with the Institute for Human-Machine Communication, Technische Universität München, Theresienstr 90, Munich, Germany. e-mail: (reza.babaee@tum.de).

M. Datcu is with Remote Sensing Technology Institute, German Aerospace Center (DLR).

Non-negative Matrix Factorization (NMF) is a linear dimensionality reduction technique that has been considered to generate a parts-based representation of images [9]–[11]. In contrast to non-linear dimensionality reduction techniques such as Laplacian Eigenmap (LE) [12], Locally Linear Embedding (LLE) [13], and Isomap [14], NMF is a parameter free algorithm that can efficiently incorporate new constraints. This property motivates us to develop new variants of NMF algorithm to employ the constraints provided by the user in order to have an interactive dimensionality reduction. Recently, some variants of NMF have been introduced to represent images for clustering [15]–[17]. For instance, semi-supervised NMF algorithms use label information in a regularizer coupled with the main objective function [18].

In this paper, we introduce two novel NMF algorithms, namely a Variance Constrained NMF (VNMF) and a Center Map NMF (CMNMF) to be used in an interactive system to represent the images, bridging the semantic gap and eventually increasing the performance of clustering. The main contributions of our work are

- an immersive interactive interface to allow the user to interact with the images,
- two NMF-based interactive algorithms that utilize user-provided similarity constraints to generate new representations of images,
- a divide-and-conquer approach [19] decreasing the computation time.

We extract the user interactions via a 3D interactive application that can run on two systems. The first one is a regular desktop PC, whereas the second one is a Cave Automated Virtual Environment (CAVE). In contrast to the desktop, the CAVE provides the user with the ability to navigate inside the images and explore the results of clustering in 3D. To visualize the images, we apply k-means clustering on the original representation of the images and visualize the cluster centers. Then, the images are positioned around their corresponding cluster centers based on their distances. The user navigates inside the images and interacts with them by assigning the mis-clustered images to their desired clusters (see Fig. 1(d)). A set of these interactions is used to build up a semantic similarity matrix, which is exploited in the main objective function of NMF. The optimization of this function eventually leads to a new image representation. Comparing the clustering results between the original and new representations, demonstrates the effectiveness of the proposed algorithm. We are furthermore able to confirm that the clustering accuracy increases with the
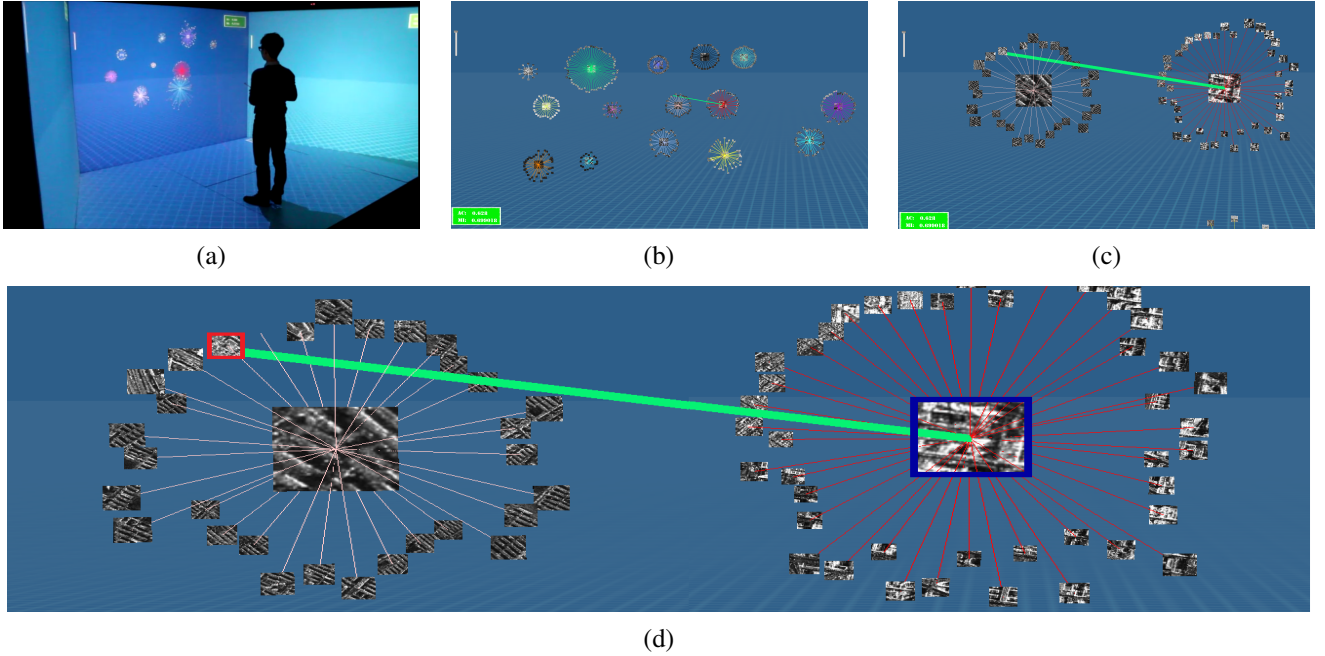
Fig. 1. (a) the visualization of the clustering result in the CAVE. Images are positioned around their cluster centers based on their distances. A sample image of each cluster is used to depict the cluster center. (b), (c) show user interactions on a desktop. A mis-clustered image is connected to a semantically cluster center by a green line. (d) a mis-clustered image (the image with red border) is connected (green line) to the cluster center of target cluster (with blue border). This interaction updated the semantic similarity matrix W, which is used in our novel NMF algorithms.

number of user interactions.

The rest of the paper is organized as follows: Section II provides an overview of NMF concept. In Section III we introduce our two novel NMF algorithms followed by the details of the immersive interactive interface. Section IV provides the details on the performed experiments. First, the datasets and the evaluation metrics of clustering are introduced. We then continue with a description of the experimental setup followed by a presentation and discussion of our results. Finally, in Section V we draw a conclusion and discuss potential topics for future work.

II. A BRIEF REVIEW OF NMF

Non-negative Matrix Factorization (NMF) [9] is a widely used matrix factorization method which provides a part-based representation of data by enforcing non-negative constraint to the matrix factors. Given a non-negative matrix $X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N}$, where each column is a feature vector representing a data sample, the goal of NMF is to factorize X into two non-negative matrices U and V such that:

$$X \approx UV^T, \quad \text{where} \quad U \in \mathbb{R}^{D \times K}, \quad V \in \mathbb{R}^{N \times K} \quad (1)$$

and we are interested in a decomposition such that $K << D$. This factorization is a constrained non-convex optimization problem with the cost function equal to:

$$F = \|X - UV^T\|_F^2 \quad (2)$$
$$\text{s.t.} \quad U = [u_{ik}] \geq 0,$$
$$V = [v_{jk}] \geq 0$$

The cost function is convex only in U or V, but not convex in both together. Therefore, there is no global solution for the

algorithm, but Lee and Seung [9], [20] presented an iterative update algorithm to find a local minimum as follows:

$$u_{ik} \leftarrow \frac{(XV)_{ik}}{(UV^T V)_{ik}}, \quad v_{jk} \leftarrow v_{jk} \frac{(X^T U)_{jk}}{(VU^T U)_{jk}} \quad (3)$$

It is proved that the updating rules can converge to a local minimum of the cost function [20]. Basically, the matrix U represents the bases of new space and the matrix V holds the new representation of data points. Later on in experiments section, we apply k-means clustering to the new representation (feature) of data points.

III. INTERACTIVE LEARNING ALGORITHMS

A. Variance Constrained NMF

The goal of VNMF is to factorize some input data X into two non-negative matrices U and V, subject to a minimum value for the variance of V (i.e., σ_V^2). Intuitively, by decreasing the variance of data points belonging to the same class, we aim to force the data points as close as possible. Evidently, if some data points come from same class, they should be very close together. So, by minimizing the variance we can reach this goal. Therefore, we minimize the following objective function:

$$F = \|X - UV^T\|_F^2 + \lambda \sigma_v^2$$
$$\text{s.t.} \quad U = [u_{ik}] \geq 0,$$
$$V = [v_{jk}] \geq 0 \quad (4)$$

The user interactions are kept in the matrix $W \in \mathbb{R}^{N \times N}$, whose elements W_{ij} are 1 if the images i and j are connected or 0 if they are not. When the user links an image i to a cluster center c containing p images, the corresponding p elements of

the row i of matrix W would be 1. In other words, $W_{ij} \neq 0$ shows the images i and j are semantically similar to each other. Finally, matrix W is updated as a weight matrix with $W_{ij} = \sum_{l=1}^{M} W_{il} = 1$, where M is the total number of non-zero elements in row i. In order to compute the variance of new features, their expectation value should be computed first. Thus, we scale the matrix W so that its rows always sum to 1, yielding matrix $\widetilde{W}$. The multiplication of $\widetilde{W}$ and V finally results in a matrix $\overline{V}$, holding the mean features of similar images. For example, given a dataset of four images, where image 1 is connected to image 2, we get:

$$\overline{V} = \widetilde{W}V = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{pmatrix} \tag{5}$$

By introducing new matrix $T = I - \widetilde{W}$, we can write

$$\sigma_V^2 = \left\| V - \overline{V} \right\|_F^2 = \left\| V - \widetilde{W}V \right\|_F^2 = \|TV\|_F^2. \tag{6}$$

In order to control the variance, another scalar parameter θ is introduced inside the regularizer. Finally, the objective function to be minimized is

$$\begin{aligned} C &= \|X - UV^T\|_F^2 + \lambda\left(n\theta - \|TV\|_F^2\right)^2 \\ &= \sum_i^D \sum_j^N \left(x_{ij} - \sum_{k=1}^K u_{ik}v_{jk}\right)^2 \\ &\quad + \lambda\left(N\theta - \sum_i^N \sum_j^K (v_{ij} - \bar{v}_{ij})^2\right)^2. \end{aligned} \tag{7}$$

where λ controls the overall contribution of the regularizer.

1) Optimization rules: To minimize the cost function given in Eq. (7), we first expand it to

$$\begin{aligned} C &= Tr\left((X - UV^T)(X - UV^T)^T\right) \\ &\quad + \lambda\left(N\theta - Tr\left((TV)(TV)^T\right)\right)^2 \\ &= Tr(XX^T) - 2Tr(XVU^T) + Tr(UV^TVU^T) + \lambda Z^2, \end{aligned} \tag{8}$$

where

$$Z = N\theta - Tr(TVV^TT^T). \tag{9}$$

We define Lagrange multipliers α_{ik} and β_{jk} with the constraints $u_{ik} \geq 0$ and $v_{jk} \geq 0$, respectively. Therefore, by defining $A = [\alpha_{ik}]$ and $B = [\beta_{jk}]$, the Lagrangian $\mathcal{L}$ is

$$\begin{aligned} \mathcal{L} &= Tr(XX^T) - 2Tr(XVU^T) + Tr(UV^TVU^T) \\ &\quad + \lambda Z^2 + Tr(AU) + Tr(BV). \end{aligned} \tag{10}$$

The partial derivatives of $\mathcal{L}$ with respect to U and V are

$$\frac{\partial \mathcal{L}}{\partial U} = -2XV + 2UV^TV + A \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial V} = -2X^TU + 2VU^TU - 2\lambda ZT^TTV + B. \tag{12}$$

Using the Karush-Kuhn-Tucker (KKT) conditions [21], where $\alpha_{ij}u_{ij} = 0$ and $\beta_{jk}v_{jk} = 0$, the following equations are obtained:

$$-(XV)_{ik}u_{ik} + (UV^TV)_{ik}u_{ik} = 0 \tag{13}$$

$$[-X^TU + VU^TU - \lambda ZT^TTV]_{jk}v_{jk} = 0 \tag{14}$$

With the symmetric matrices $T = T^+ - T^-$, where $T_{ij}^+ = (|T_{ij}| + T_{ij})/2$ and $T_{ij}^- = (|T_{ij}| - T_{ij})/2$, the update rules for U and V can be rewritten as:

$$u_{ik} \leftarrow u_{ik} \frac{(XV)_{ik}}{(UV^TV)_{ik}} \tag{15}$$

$$v_{jk} \leftarrow v_{jk} \frac{(X^TU - 2\lambda ZT^+T^-V)_{jk}}{(VU^TU - \lambda ZT^+T^+V - \lambda ZT^-T^-V)_{jk}} \tag{16}$$

Here matrix V contains the newly generated features that represent the content of images. These features are used in k-means algorithm to cluster the images.

B. Center Map NMF

In CMNMF, the user-interaction/semantic information is incorporated/injected inside the main function of NMF. We create matrix $\widetilde{W}$ in the same way we have described in Section III-A and denote it here as W. In addition, we introduce an auxiliary matrix Z in order to get matrix V. For example, suppose there are four images, the operation is:

$$V = WZ = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ z_{31} & z_{32} & z_{33} & z_{34} \\ z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{z_{11}+z_{21}}{2} & \frac{z_{12}+z_{22}}{2} & \frac{z_{13}+z_{23}}{2} & \frac{z_{14}+z_{24}}{2} \\ \frac{z_{11}+z_{21}}{2} & \frac{z_{12}+z_{22}}{2} & \frac{z_{13}+z_{23}}{2} & \frac{z_{14}+z_{24}}{2} \\ z_{31} & z_{32} & z_{33} & z_{34} \\ z_{41} & z_{42} & z_{43} & z_{44} \end{pmatrix} \tag{17}$$

From this example, we can see that the new representation guarantees that the first two images (first two rows of V) will be assigned to the same cluster center. Adding the introduced terms to the NMF formulation leads to the following minimization objective:

$$\min_{U,Z} \mathcal{O} = \min_{U,Z} \left\| X - UZ^TW^T \right\| \tag{18}$$

where X, the original image representations, is decomposed into U and V, V being the new representation of the images and U being the bases. For the derivation of the update rules we expand this objective to

$$\begin{aligned} \mathcal{O} &= Tr((X - UZ^TW^T)(X - UZ^TW^T)^T) \\ &= Tr(XX^T) - 2Tr(XWZU^T) + Tr(UZ^TW^TWZU^T) \end{aligned} \tag{19}$$

and introduce Lagrange multipliers $\Phi = [\phi_{ik}]$ and $\Psi = [\psi_{jk}]$ for the constraints $[u_{ik}] \geq 0$ and $[v_{jk}] \geq 0$, respectively. This leads to the Lagrangian

$$\mathcal{L} = \mathcal{O} + Tr(\Phi U^T) + Tr(\Psi Z^T) \tag{20}$$

The partial derivatives of $\mathcal{L}$ with respect to U and Z are

$$\frac{\partial \mathcal{L}}{\partial U} = -2XWZ + 2UZ^T W^T WZ + \Phi \qquad (21)$$

$$\frac{\partial \mathcal{L}}{\partial Z} = -2W^T X^T U + 2W^T WZU^T U + \Psi \qquad (22)$$

Using the Karush-Kuhn-Tucker (KKT) conditions [21], we arrive at the following update rules for U and Z:

$$u_{ij} \leftarrow u_{ij} \frac{(XWZ)_{ij}}{(UZ^T W^T WZ)_{ij}} \qquad (23)$$

$$z_{jk} \leftarrow z_{jk} \frac{(W^T X^T U)_{jk}}{(W^T WZU^T U)_{jk}} \qquad (24)$$

Like VNMF, the matrix V contains the newly generated features that represent the content of images. These features are used in k-means algorithm to cluster the images.

C. Immersive Interface

We utilize a 3D interactive application development software, namely 3D Via Studio, to create our interactive visualization system. Our application can be run on a desktop PC or in an immersive 3D virtual environment such as a Cave Automated Virtual Environment (CAVE) [22], [23]. The CAVE is composed of four room-sized walls, i.e., displays, and a tracking system to capture the human motion. The user is provided with an Xbox 360 gamepad controller to navigate inside the virtual space and interact with the images.

The 3D positions of images are determined by the clustering result of the k-means method, as well as the distances between the images and their corresponding centers. More precisely, the distance between an image and its cluster center is proportional to its actual distance in the high-dimensional space. Thus, the user can first scan the images that are far away from their respective cluster centers, because they are prone to be misclustered.

In addition to the CAVE, we also implemented a desktop version of our application running on a single PC. A snapshot of the immersive visualization of the clustering results in the CAVE is depicted in Fig. 1(a). Fig. 1(b) and Fig. 1(c) depict two views of a clustering result visualized on a desktop PC. The main advantage of using the immersive visualization technology is that the user gets an overview of the whole dataset and can therefore identify those images that should be relabeled more efficiently. This is especially important if the user is dealing with a high amount of images that cannot be visualized on a limited-space monitor.

IV. EXPERIMENTS

A. Data set

The dataset used in our experiments is a Synthetic Aperture Radar (SAR) dataset [24] represented by three different features. It consists of a collection of 3434 SAR images of the size 160×160 pixels, pre-categorized into 15 classes/labels (by a SAR expert) based on the presences of forests, water, roads

and urban area density. For instance, one image is categorized as "sea" and another as "industrial part". Three different feature vectors, namely Weber Local Descriptor (WLD) [25], [26], Mean-Variance [27], [28] and Image Intensity [27] were extracted from the images, leading to a total of 64 dimensions in these three cases. All features are normalized to lie between 0 and 1.
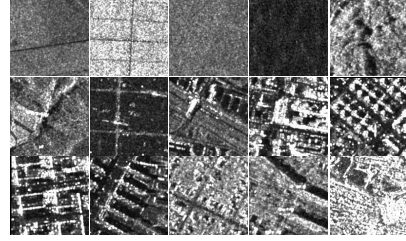

Fig. 2. Sample images from the SAR data set. There are 15 images, each one is representing one class.

B. Evaluation Metrics

We use two metrics to evaluate the clustering result of new image representations, namely Accuracy (AC) and Normalized Mutual Information (nMI) [24].

1) Accuracy: The accuracy represents the percentage of correctly predicted labels compared to the ground true labels. Given a dataset with N samples, where for each sample, t_i indicates its true label given by the data set and p_i is the label predicted by a clustering algorithm, the accuracy is defined as

$$AC = \frac{\sum_{i=1}^{N} \delta(t_i, \mathrm{map}(p_i))}{N}, \qquad (25)$$

where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise, and $\mathrm{map}(p_i)$ is a function that maps each label to the corresponding label in the data set. The permutation mapping is determined using the Kuhn-Munkres algorithm [24].

2) Normalized mutual information: The similarity of two clusters is determined by normalized mutual information. Given two sets of clusters $C = \{c_1, ..., c_k\}$ and $\acute{C} = \{\acute{c}_1, ..., \acute{c}_k\}$, the mutual information metric is computed by

$$MI(C, \acute{C}) = \sum_{c_i \in C, \acute{c}_j \in \acute{C}} p(c_i, \acute{c}_j) \log \frac{p(c_i, \acute{c}_j)}{p(c_i) p(\acute{c}_j)}, \qquad (26)$$

where $p(c_i), p(\acute{c}_j)$ represent the probability that an arbitrarily selected data point belongs to the clusters C or $\acute{C}_j$, respectively, and $p(c_i, \acute{c}_j)$ represents the joint probability that a point belongs to both clusters simultaneously. As the similarity of the two clusters increases, the mutual information $MI(C, \acute{C})$ increases from 0 to $\max\left\{H(C), H(\acute{C})\right\}$. $H(C), H(\acute{C})$ represent the entropy of the clusters $C, \acute{C}$ respectively. Dividing the mutual information by $\max\left\{H(C), H(\acute{C})\right\}$ leads to the normalized mutual information, which takes values between 0 and 1:

$$nMI(C, \acute{C}) = \frac{MI(C, \acute{C})}{\max\left\{H(C), H(\acute{C})\right\}}. \qquad (27)$$

C. Design

Given a set of N images, we randomly selected $10-15$ percent of the images as training data. In our experiment, 500 images are chosen as training data. For this new image set, we applied k-means clustering algorithm and visualized the images by the cluster-based visualization system. we visualized the images by 15 clusters. The user navigated inside the data and corrected the mis-clustered images by drawing a green line between the image and the center of the desired target cluster. The interactions were saved in a matrix with 2 columns and I rows where I is the number of interactions. First column stored the ID of the interacted image and the second column saved the ID of the target cluster. The interactive matrix was used to create the weighted similarity matrix for systems based on VNMF and CMNMF. The images would have new representations that showed better cluster accuracy. For example, the user moved image i_2 to the cluster c_2. And there were two images i_6 and i_7 belonged to c_2. The images i_2, i_6 and i_7 would be regarded as similar images and the positions in similarity matrix $(2,6),(2,7),(6,2),(6,7),(7,2)$ and $(7,6)$ would be set to 1. After updating the matrix W, the matrix $\widetilde{W}$ was calculated by calculated row-wise average, namely

$$\widetilde{W}_{ij} = \frac{W_{ij}}{\sum_{l=1}^{N} W_{il}} \quad (28)$$

where N was the number of images in total. Then, for VNMF, the matrix T was updated by $T = I - \widetilde{W}$ where I was an identity matrix. The new matrix T would be used in VNMF to calculate the new data representation V where V would show better clustering result in low dimensional space. For CMNMF, the matrix $\widetilde{W}$ was directly used in updating rules to obtain the new data representation. All similar images would be mapped to their semantic centers and as a result, the new representations provided good clustering accuracy.

The dimension of new representation matrix V was set to the number of images by the number of classes. For our SAR data sets it was set to be $N \times 15$. For CMNMF, the similarity matrix was directly applied to the updating rules since the matrix V was replaced by WZ. Thus, no parameter was needed. For VNMF, the regularization parameter was used to control the contribution of the user interactions. The parameter was chosen by tuning the value λ through searching the grid $\{10^{-7}, 10^{-6}, 10^{-5}, ..., 10^2, 10^3\}$. We apply VNMF algorithm on the training data with different lambda values from the grid and then apply the k-means algorithm 10 times to the obtained new features (matrix V). The lambda value with highest average accuracy is selected as the proper lambda for the test dataset. Since the new representation V consisted of small values$(0 \sim 0.6)$and the updating rules were sensitive to the regularization terms, the parameter between 10^{-6} and 10^{-2} was chosen in most cases.

After training, we used these three learning algorithms in two different ways for test data: 1) cluster the test data following the change of interaction number and 2) cluster the test data following the change of the dimension of new representations.

Traditionally, after training, the test data was processed as

a whole data set, called batch processing. Since the size of test data set is much larger than the size of training data, the divide-and-conquer processing is used while clustering the test data. Compared with batch processing, divide-and-conquer processing provides similar performance in clustering accuracy with much less running time. The test data was divided into parts with the same size of training data. Then we mixed the training data with each part and applied learning algorithms to obtain new representations for each part. The clustering results of k-means algorithm on each part were averaged as the final result of the whole data set. A schematic of this process is depicted in Fig. 3.
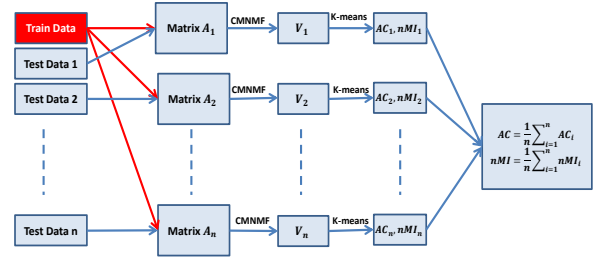

Fig. 3. A schematic view of the proposed divide-and-conquer approach to get a new representation of the data for clustering. Here, the training data is mixed with each part of test data and is fed into VNMF/CMNMF to get new representation V. k-means is applied on each V separately and the results are mixed as the final results of clustering.

When clustering the test data following the change of interaction number, the dimension of the matrix V are fixed to the number of images by the number of classes. Based on different numbers of interactions, the test data are classified and the clustering results are calculated.

When clustering the test data following the change of the dimension of new representations, the number of interactions is fixed to 180. The new representation matrix V will have the size of $N \times k$, where N is the number of images, k changes in the range $3, 6, 9, 12, 15$ for the SAR data set. For matrix V with different sizes, the clustering results are calculated and compared with other algorithms.

To enhance the user's effect, the locality property [29] was used to propagate the user's interaction. In another word, the user's interaction on training data would be applied to their nearest neighbors in test data. The idea is shown in Fig.4. The usage of locality property could produce errors since it
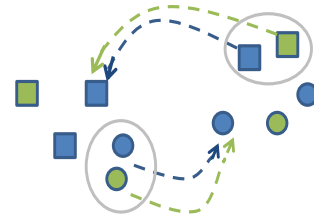

Fig. 4. The objects in blue are the training data, and in green are test data. The square and circle indicates different classes. The dash line shows the similarity interaction. The blue dash interactions are done by the user while training the data. The green dash interactions are done by the system while applying the locality property

uses Euclidean distance to find nearest neighbor. However, it strengthened the user interaction and the learning algorithm would correct the error produced by locality property.

D. Results

We compared the clustering results of new representations obtained from VNMF and CMNMF with that of the k-means clustering algorithm on original high-dimensional features, PCA, and NMF as a function of number of interactions and dimension of subspace (i.e., the dimension of newly generated features) separately. The three columns in both Fig. 5 and Fig.6 shows the results of SAR images represented by Mean Variance, Image Intensity, and WLD features, respectively.

Fig. 5 shows the experimental results of clustering test data with the change of interaction number. As the Fig. 5 shows, by increasing the number of interactions, the clustering accuracy of VNMF and CMNMF for all data sets is increased by $10 - 15\%$. The user interaction provides more improvement in Mean-Variance and WLD features than Image Intensity features. For mutual information, all algorithms present similar performance within the range of $\pm 3\%$ which is reasonable for heuristic algorithms.

Fig. 6 presents the experimental results of clustering test data with the change of dimension of subspace (i.e., the dimension of newly generated features). Here, the dimension of subspace is the number of columns of new data representation V. As shown in the Fig. 6, with the increase of the dimension of new representations, all algorithms except k-means show the improvement in accuracy and mutual information. Among these algorithms, VNMF and CMNMF shown in green and blue line respectively, offers better performance than other algorithms for all dimensions. It provides about $5 - 10\%$ improvement than other algorithms. Compared among these three features, the user interaction improves the accuracy most for Mean-Variance features with more than 10%. For the mutual information, all algorithms have similar performances. Additionally, by observing the accuracy in Fig. 6, we can find that for feature Mean-Variance and Image Intensity, once the dimension of subspace reaches 6, further increment in dimension of subspace cannot improve the clustering results in accuracy and mutual information. For WLD feature, after the dimension reaches 9, the accuracy and mutual information also reach their highest points in this dimension range. The results imply that, instead of setting the dimension of new representation to the number of classes, choosing some smaller values, like 6 for Mean-Variance and Image Intensity and 9 for WLD, will not affect the clustering result but decrease the size of new representations a lot. The computation time of each algorithm applied to a dataset consisting of 1000 samples is presented in Fig. 9. As the figure show, the VNMF algorithm has lowest computational cost in comparison to CMNMF.

Additionally, in Fig. 7, we present exemplary results of clustering algorithm (i.e., k-means) applied to the original features and also the new features obtained from VNMF and CMNMF algorithms. Here, we randomly select four classes and 16 images for each class. The correct and incorrect images are depicted by green and red borders, respectively. As the

results show, the number of errors is decreasing by applying VNMFand CMNMF algorithms. There are several differences between these two algorithms. First, in VNMF, there is a parameter that controls the variance of new feature, while we dont have any parameter in CMNMF algorithm. On the other side, VNMF algorithm is much faster than CMNMF algorithm. However, CMNMF produces at least comparable results with VNMF. In conclusion, VNMF is faster, but its performance depends on the parameter.

E. Convergence

Fig.8 shows the convergence speed of NMF, VNMF and CMNMF for features Mean-Variance, Image Intensity and WLD of SAR. As shown in figures above, these three algorithms converge within 10 iterations. VNMF provides a better objective value that is much smaller than that in NMF and CMNMF provides smallest objective value.

The computation times of divide-and-conquer and batch processing are depicted in Fig. 10. The results confirm that the divide-and-conquer approach is about four times faster than batch processing. Fortunately, all matrix elements are updated independently from each other. Therefore, the factorization can be implemented on GPU [30] and consequently, the computation times would decrease much more.
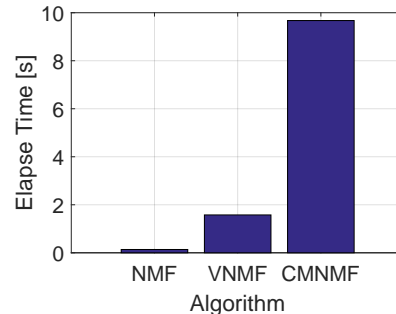

Fig. 9. Computation time of NMF, VNMF, and CMNMF applied to a dataset of 1000 samples.
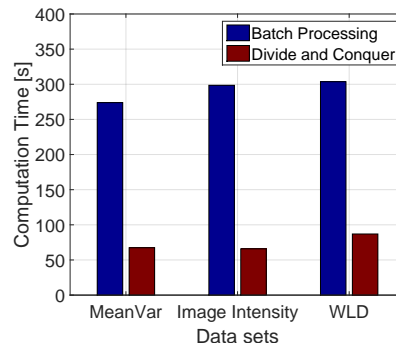

Fig. 10. Computation times of divide-and-conquer compared to batch processing for all three features. The experiments were executed on a desktop PC with an Intel Core2Quad 2,8GHz CPU and 8GB of RAM. The divide-and-conquer approach is on average four times faster than batch processing.
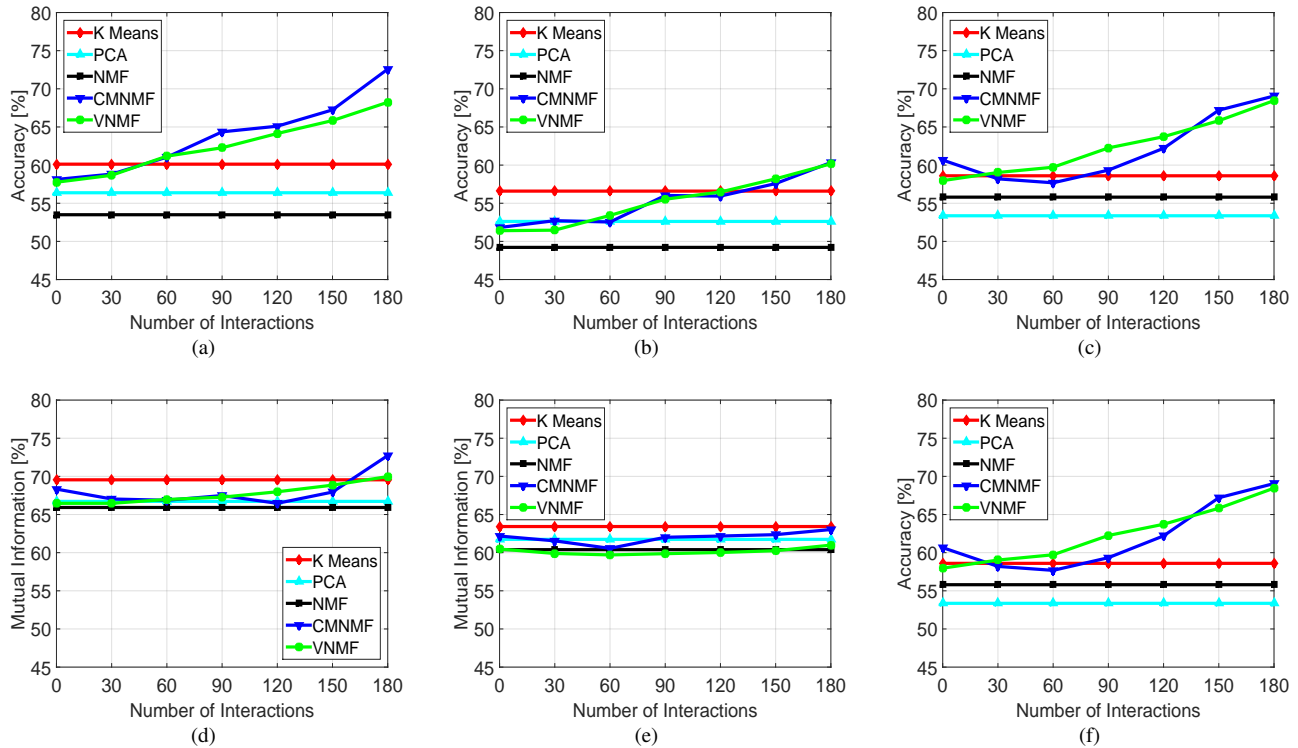
Fig. 5. Clustering results of the proposed divide-and-conquer approach as a function of the number of interactions, represented by accuracy (first row) and normalized mutual information (second row). The first, second and third columns show the results SAR images represented by Mean-Variance, Image intensity and WLD features, respectively.
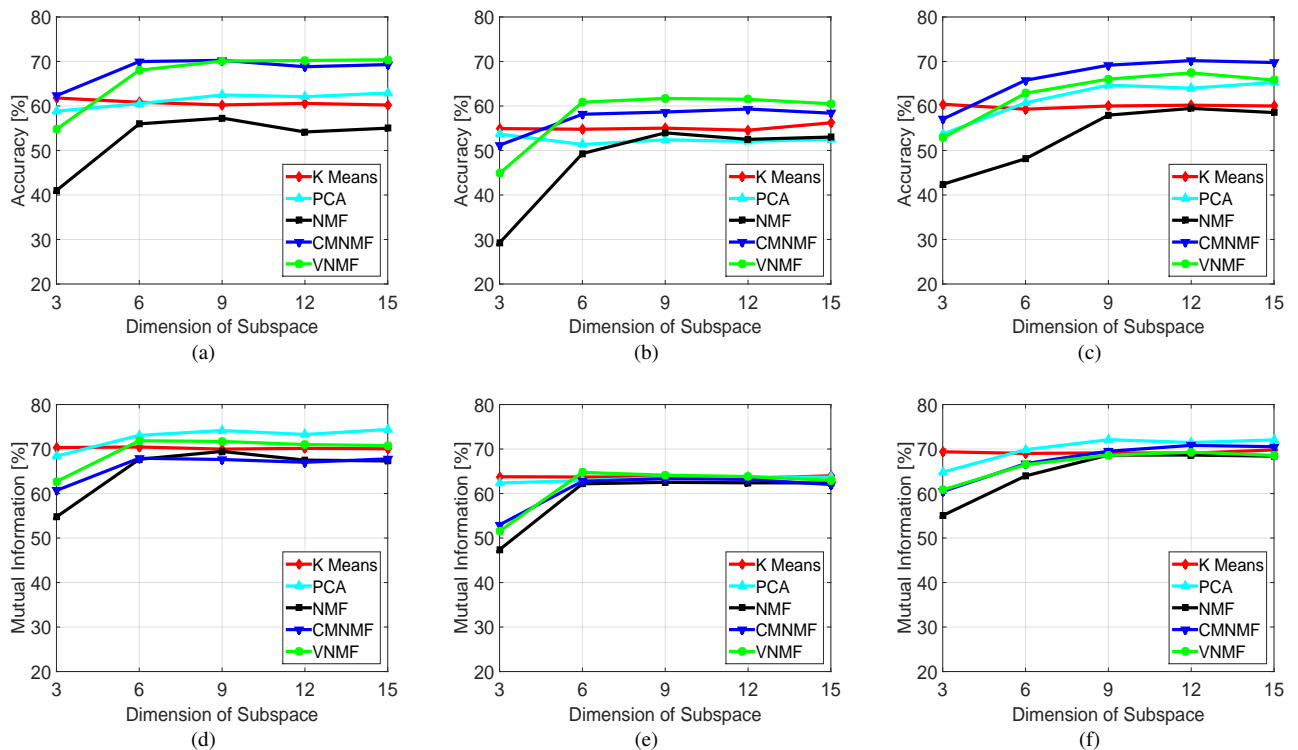

Fig. 6. Clustering results of different subspaces (new features) represented by accuracy (first row) and normalized mutual information (second row). The first, second and third columns show the results of SAR images represented by Mean-Variance, Image intensity and WLD features, respectively.
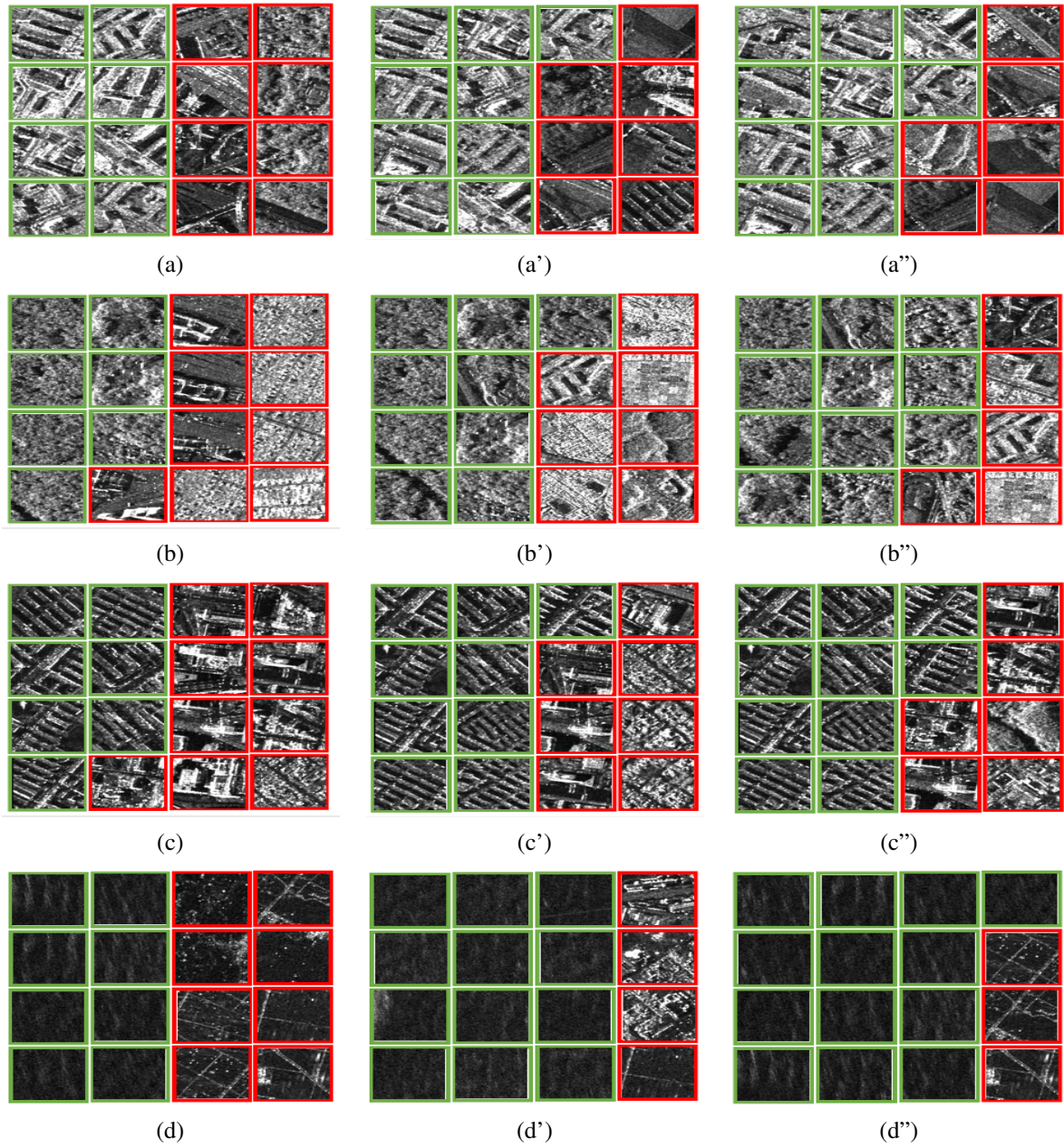
Fig. 7. Exemplary images of four randomly chosen classes after applying k-means clustering to original features (left column) and the new features obtained by VNMF (middle column) and CMNMF (right column). Each row shows sample images of a random chosen class. Green borders depict correct clustered images and the red ones depict incorrect clustered images.
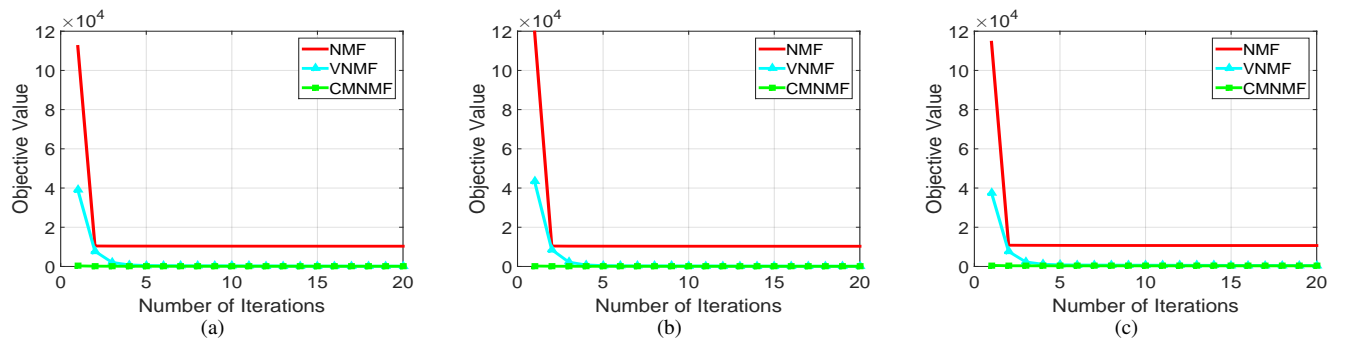

Fig. 8. The convergence speed for NMF, VNMF and CMNMF applied on three features; a)Mean-Variance; b) Image Intensity; c) WLD.

V. Conclusions

We proposed two novel NMF based algorithms (i.e., CMNMF and VNMF) utilizing user-data interactions in order to bridge the semantic gap between human and machine perception of images. For interactions, we visualize the images in a 3D interactive environment using k-means. The visualization application can run on a regular desktop PC or in a CAVE. The user-provided interactions are exploited in the main objective function of CMNMF and VNMF to generate semantically more meaningful representations. Experimental results on a SAR data set have shown that the algorithm increases the clustering accuracy significantly. To decrease the computation time, we proposed a divide-and-conquer approach to factorize the matrices which clearly outperforms regular batch processing.

References

[1] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 12, pp. 1349–1380, 2000.

[2] C. Dorai and S. Venkatesh, "Bridging the semantic gap with computational media aesthetics," *IEEE multimedia*, vol. 10, no. 2, pp. 15–17, 2003.

[3] F. Hu, G.-S. Xia, Z. Wang, X. Huang, L. Zhang, and H. Sun, "Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 5, pp. 2015–2030, May 2015.

[4] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in neural information processing systems*, 2002, pp. 505–512.

[5] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 209–216.

[6] J. Yi, R. Jin, S. Jain, T. Yang, and A. K. Jain, "Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning," in *Advances in Neural Information Processing Systems*, 2012, pp. 1772–1780.

[7] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine learning*, vol. 74, no. 1, pp. 1–22, 2009.

[8] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 11.

[9] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[10] M. Babaee, S. Tsoukalas, M. Babaee, G. Rigoll, and M. Datcu, "Discriminative nonnegative matrix factorization for dimensionality reduction," *Neurocomputing*, vol. 173, pp. 212–223, 2016.

[11] M. Babaee, S. Tsoukalas, G. Rigoll, and M. Datcu, "Immersive visualization of visual data using nonnegative matrix factorization," *Neurocomputing*, vol. 173, pp. 245–255, 2016.

[12] J. J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*. Springer, 2007.

[13] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[14] J. B. Tenenbaum, V. de Silva, and J. C. Langford, *a global geometric framework for nonlinear dimensionality reduction*. Science, 2000.

[15] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 577–582.

[16] Y. Chen, M. Rege, M. Dong, and J. Hua, "Non-negative matrix factorization for semi-supervised data clustering," *Knowledge and Information Systems*, vol. 17, no. 3, pp. 355–379, 2008.

[17] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization." in *SDM*. SIAM, 2008, pp. 1–12.

[18] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *Signal Processing Letters, IEEE*, vol. 17, no. 1, pp. 4–7, 2010.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein *et al.*, *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.

[20] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

[21] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

[22] H. Creagh, "Cave automatic virtual environment," in *Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Technology Conference, 2003. Proceedings*. IEEE, 2003, pp. 499–504.

[23] M. Babaee, G. Rigoll, and M. Datcu, "Immersive interactive information mining with application to earth observation data retrieval," in *Availability, Reliability, and Security in Information Systems and HCI*. Springer, 2013, pp. 376–386.

[24] M. Babaee, R. Bahmanyar, G. Rigoll, and M. Datcu, "Farness preserving non-negative matrix factorization," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3023–3027.

[25] J. Chen, S. Shan, G. Zhao, X. Chen, W. Gao, and M. Pietikainen, "A robust descriptor based on weber's law," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–7.

[26] M. Babaee, S. Tsoukalas, G. Rigoll, and M. Datcu, "Visualization-based active learning for the annotation of sar images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2015.

[27] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *Image processing, IEEE Transactions on*, vol. 11, no. 4, pp. 467–476, 2002.

[28] S. Cui, C. O. Dumitru, and M. Datcu, "Ratio-detector-based feature extraction for very high resolution sar image patch indexing," *Geoscience and Remote Sensing Letters, IEEE*, vol. 10, no. 5, pp. 1175–1179, 2013.

[29] H. R., C. S., and L. Y., "Dimensionality reduction by learning an invariant mapping," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006.

[30] E. Battenberg and D. Wessel, "Accelerating non-negative matrix factorization for audio source separation on multi-core and many-core architectures." in *ISMIR*, 2009, pp. 501–506.