Automatic Graph Building Approach for Spectral Clustering

Andrés Eduardo Castro-Ospina, Andrés Marino Álvarez-Meza,
and César Germán Castellanos-Domínguez

Signal Processing and Recognition Group, Universidad Nacional de Colombia,
Manizales, Colombia
{aecastroo,amalvarezme,cgcastellanosd}@unal.edu.co

Abstract. Spectral clustering techniques have shown their capability to identify the data relationships using graph analysis, achieving better accuracy than traditional algorithms as k-means. Here, we propose a methodology to build automatically a graph representation over the input data for spectral clustering based approaches by taking into account the local and global sample structure. Regarding this, both the Euclidean and the geodesic distances are used to identify the main relationships between a given point and neighboring samples around it. Then, given the information about the local data structure, we estimate an affinity matrix by means of Gaussian kernel. Synthetic and real-world datasets are tested. Attained results show how our approach outperforms, in most of the cases, benchmark methods.

Keywords: Graph analysis, kernel function, spectral clustering.

1 Introduction

Clustering techniques are widely used to explore data patterns and they provide the advantage to work with unlabeled data. These techniques have been addressed in many disciplines as data mining, image segmentation, and pattern classification [1, 2]. Although, well-known algorithms, such as k-means, are employed in clustering applications, however, they only consider similarity values from instances to a fixed number of centers. Moreover, they require extra information about cluster shape, which is not always available.

Therefore, two approaches have emerged as an alternative to analyze clusters that are non-linearly separable, namely, kernel k-means and spectral clustering. Spectral techniques seek data representation as a graph, with a set of nodes and an affinity matrix capturing relationships among samples [1]. In addition, using an affinity matrix allows to employ powerful operators such as kernel functions, in order to reveal the main data structures. Regarding this, fixing kernel operators is crucial for the clustering performance. In [3], a local scaling parameter is introduced to identify a suitable kernel function considering the neighborhood relationships. Nonetheless, it requires to fix a free parameter that

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part I, LNCS 8258, pp. 190–197, 2013.

is not always a straightforward task. Moreover, due to the fact that the method considers a different local scaling for a given sample, the obtained representation does not correspond to conventional kernel function class satisfying the Mercer conditions [4]. Though some applications are discussed on this matter [5–7], this method can not longer be framed as a suitable kernel based representation. Moreover, as shown in our experiments, it is not always a good alternative to build the graph for spectral clustering.

We propose a new alternative to construct automatically the graph representation in spectral clustering approaches. Particularly, inspired by a previous method that allows to identify the local and global data structures for manifold learning tasks [8], two different operators (namely, the Euclidean and the geodesic distances) are used to highlight the main relationships between a given point and the neighboring samples. To this end, a neighborhood size is calculated for each sample, looking for the largest patch that allows to model each neighborhood as locally linear. Provided that local data structure information is encoded into neighborhood sizes, we estimate an affinity matrix by means of a Gaussian kernel fixing the band-width parameter as a function of the found neighborhoods. For the sake of assessing the proposed methodology performance, some experiments are done over synthetic and real-world datasets. Obtained results are compare against state of the art approaches [3,5,6].

2 Methods

2.1 Spectral Clustering Main Concepts

Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ be an input data matrix holding n samples and p features. To discover the input data structure, relationships among samples can be highlighted by means of a complete, weighted, undirected graph representation $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{\Omega})$, which contains a set of nodes $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ corresponding to the n samples. Edge weights for connecting nodes i and j $(i \neq j)$ are defined through an affinity matrix $\boldsymbol{\Omega} \in \mathbb{R}^{n\times n}$, with $\Omega_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, being $\kappa(\cdot,\cdot)$ a kernel function, mostly, the Gaussian kernel [1]. Using a kernel function ensures an stable spectral decomposition, due to it must satisfy the Mercer conditions. The goal of clustering approaches is to decompose $\boldsymbol{V}$ into C disjoint subsets as $\boldsymbol{V} = \cup_{c=1}^{C} \boldsymbol{V}_c$, with $\boldsymbol{V}_l \cap \boldsymbol{V}_c = \emptyset \; \forall l \neq c$. To this end, spectral information and orthogonal mappings from $\boldsymbol{\Omega}$ are employed to represent suitably inputs [2]. Thus, using spectral concepts of graph analysis, the so-called Laplacian matrix is estimated as $\boldsymbol{L} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{\Omega} \boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose elements $d_{ii} = \sum_{i=1}^{n} \Omega_{ij}$ are the degree of the nodes in $\boldsymbol{G}$. Spectral decomposition of $\boldsymbol{L}$ gives useful information about graph properties, being able to cluster together similar patterns [1]. Therefore, spectral clustering methods find a new representation of patterns from the first C eigenvectors of graph Laplacian $\boldsymbol{L}$. Then, given a matrix $\boldsymbol{Z} \in \mathbb{R}^{n\times C}$ whose column vectors stack the found eigenvectors, each of them with unit length, a clustering algorithm, such as K-means, is employed to minimize distortion. Note that the $\boldsymbol{Z}$ matrix can be viewed as a data mapping into a unit hypersphere, where a traditional clustering approach is used to estimate the disjoint subsets $\boldsymbol{V}_c$ and the label vector $\boldsymbol{y} \in \mathbb{R}^{n \times 1}$ containing the subset membership $y_i \in \{1, \ldots, C\}$ for each $\boldsymbol{x}_i$.

2.2 Local Data Analysis for Automatic Graph Building - AGB

Computation of affinity matrix $\boldsymbol{\Omega}$ is a crucial step in spectral clustering, since it models both local and global data properties. Commonly, the relationships among samples are identified by means of a Gaussian kernel, defined as $\Omega_{ij} = \exp\left(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2/2\sigma^2\right)$. However, the question arises as how to select the kernel band-width $\sigma \in \mathbb{R}^+$ for revealing the real data structure. In [3], as an alternative solution, a local scaling is introduced that finds a different band-width for each pair of points, namely, $\Omega_{ij} = \exp\left(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2/2\sigma_i\sigma_j\right)$, where $\sigma_i = ||\boldsymbol{x}_i - \boldsymbol{x}_h||_2^2$, being $\boldsymbol{x}_h$ the h-th neighbor of $\boldsymbol{x}_i$ according to the Euclidean distance. Nonetheless, selection of h is not a straightforward task. In [3,5], h is empirically fixed as 7, but as shown in our experiments, it is not always a suitable value. Moreover, taking into account that a kernel representation induces a nonlinear mapping $\varphi : \mathbb{R}^{n \times p} \rightarrow \mathcal{H}$, where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space - RKHS, choosing a different kernel generates a different RKHS for each pair of nodes (i, j). Therefore, variation of Gaussian kernel band-width, as the product $\sigma_i\sigma_j$, generates a different RKHS for each input sample. Hence, matrix $\boldsymbol{\Omega}$ should not correspond to a kernel representation satisfying Mercer conditions [4]. Certainly, the above mentioned procedure is often carried out in practice, but it can no longer be framed as a suitable kernel based representation.

In this work, we propose an alternative solution to build the graph $\boldsymbol{G}$ in spectral clustering based approaches, considering both the density and the linearity of each sample neighborhood. Inspired by a previous approach for fixing the neighborhood size of each sample $\boldsymbol{x}_i$ in manifold learning related tasks [8], the local data structure is studied using two main distance operators: the Euclidean and the geodesic distances. The main idea is to construct patches, i.e., neighborhoods, as large as possible, in order to conserve the global data properties, but ensuring that any data point and its nearest neighbors can be modeled as locally linear, preserving the local data structure. For each point, the nonlinear properties of its neighboring region are highlighted comparing the neighborhood found by the Euclidean distance against the neighborhood found by the geodesic distance. If the region around a point is linear and dense, the Euclidean and geodesic distances should obtain a similar set of nearest neighbors for each $\boldsymbol{x}_i$. Otherwise, the neighborhood computed using Euclidean distance should contain short circuits, while geodesic distance will be able to correctly identify the neighbors of each sample avoiding such short circuits, because it is able to model nonlinear data structures. Mainly, the algorithm to find each neighborhood size can be summarized as follow.

Firstly, to conserve the global data properties, a set of possible neighborhood size values k are calculated, where a lower bound is computed as the minimum k that allows to construct a connected graph $\boldsymbol{G}$ over $\boldsymbol{X}$. Second, varying the patch size two kind of neighbor sets are obtained according to each distance operator. Then, the vector $\boldsymbol{k} \in \mathbb{R}^{n \times 1}$ that holds the size of each computed neighborhood is calculated, where k_i is fixed as the largest neighborhood size that shares the maximum percentage of neighbors between the two kind of sets. Finally, vector $\boldsymbol{k}$ is refined by an outlier detection stage to avoid the influence of noisy samples. For a complete description about the algorithm, see [8].

Given a vector $\boldsymbol{k}$ holding information about the local data structure, our goal is to estimate an affinity matrix by means of a kernel function that allows to model properly the data. In this regard, to fix the Gaussian kernel band-width parameter, a $\sigma_i^\dagger$ value is computed for each sample as $\sigma_i^\dagger = \|\boldsymbol{x}_i - \boldsymbol{x}_{k_i}\|_2$, where $\boldsymbol{x}_{k_i}$ is the k_i-th nearest neighbor of $\boldsymbol{x}_i$. Note that $\sigma_i^\dagger$ provides information about the data dispersion into the largest local linear patch around each node in the graph. Afterwards, the kernel band-width value is computed as $\hat{\sigma} = E\{\sigma_i^\dagger\}$, where $\boldsymbol{\mathcal{E}}\left\{\cdot\right\}$ stands for expectation operator. Finally, the graph $\boldsymbol{G}$ is built over $\boldsymbol{X}$ using the $\hat{\sigma}$ value to estimate $\boldsymbol{\Omega}$. Fig. 1 presents the general scheme of the proposed approach, termed *Automatic Graph Building* - AGB.

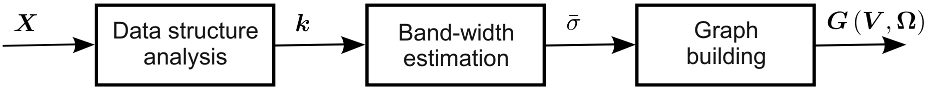


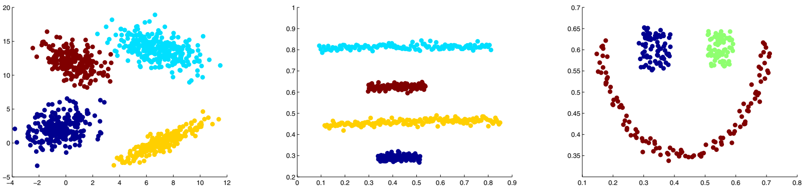
Fig. 1. Automatic graph building general scheme

3 Experimental Set-Up and Results

To test the capability of the proposed approach AGB for finding a suitable graph representation in spectral clustering based methods, some synthetic and real-world dataset are used. AGB is employed to compute the affinity matrix $\boldsymbol{\Omega}$ building a graph $\boldsymbol{G}$ over the input data. Then, a spectral clustering method is employed to estimate the label vector $\boldsymbol{y}$. Firstly, three well-known synthetic datasets are studied: four Gaussians, elongated groups, and happy face [3]. All datasets encode complex structures and are commonly used to test the capability of clustering algorithms. For concrete testing, the number of groups C is manually fixed as 4, 4, and 3, respectively, as detailed in [3]. Synthetic data clustering results are depicted in Fig. 2, which can be visually evaluated.

Regarding to real-world datasets experiments, some well-known images for segmentation tasks are employed. More precisely, several samples of the free access Berkeley Segmentation dataset are studied [1]. It is important noting that the dataset also provides hand-labeled segmentations. In our experiments, randomly selected images identified as *100075-bears*, *113044-horses*, *12003-starfish*, *388016-woman*, *56028-wall*, and *124084-flowers* are studied. Again, AGB is employed to represent properly relationships among samples, taking into account the RGB color space and the 2D position of each pixel as an input sample. However, due to limitations in memory usage, images are resized at 15%. Furthermore, a closed approach, termed

[1] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`



(a) Four gaussians (b) Elongated groups (c) Happy face

Fig. 2. AGB clustering results over synthetic data

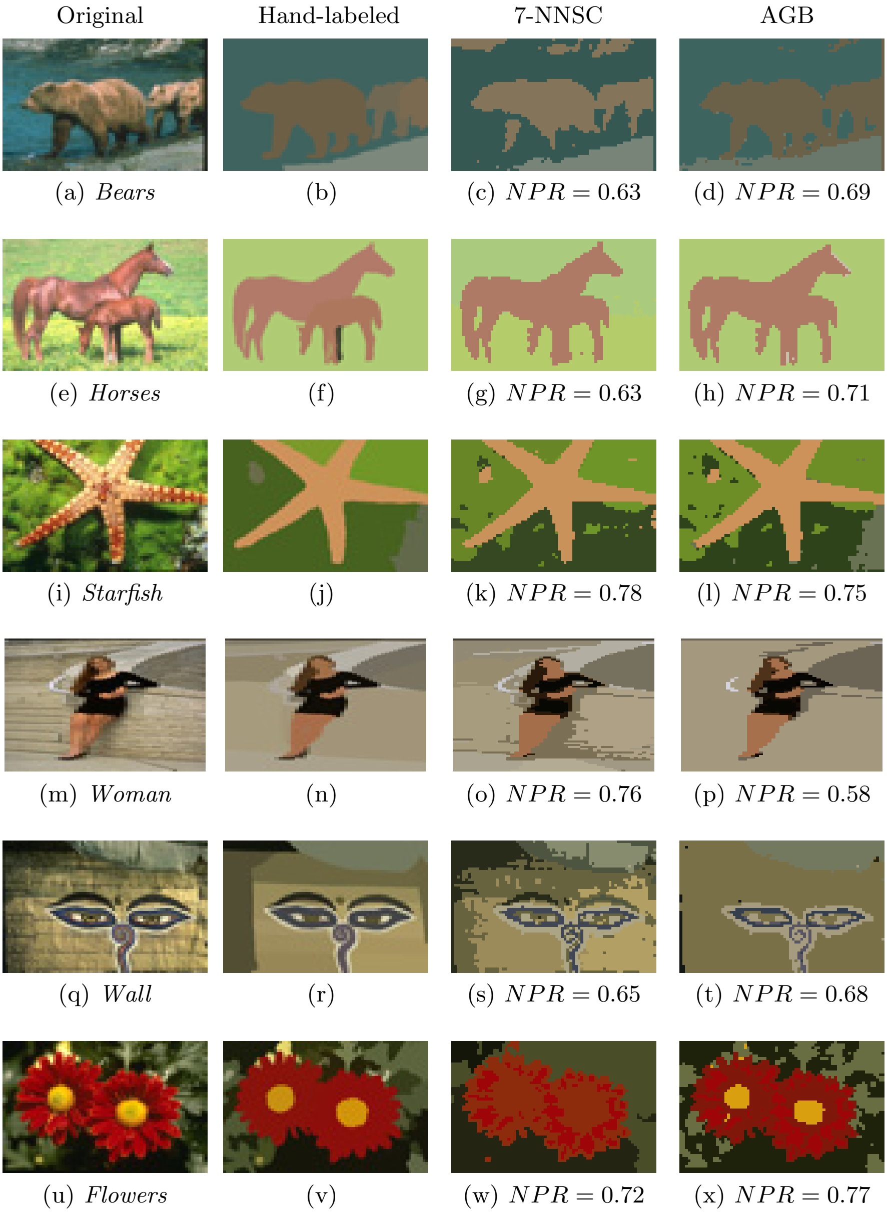
7-Nearest Neighbor Spectral Clustering - 7-NNSC, is tested. 7-NNSC is based on a local scaling analysis to build $\boldsymbol{G}$, as discussed in section 2.2 (for details see [3,5]). Besides, an index, called Normalized Probabilistic Rand - NPR, is computed to quantify the image segmentation performance, since it allows to compare a test segmentation with multiple hand-labeled ground-truth images [9]. NPR can be seen as a function $\phi\left(S,H\right)$, which compares a test segmentation S with a multiple hand-labeled ground truth images H, through soft nonuniform weighting of pixel pairs as function of the variability in the ground-truth set [9]. Fig. 3 shows images segmentation results.

Finally, some classification experiments are developed to verify the advantages of our AGB approach for highlighting the main data structures. Thus, the UCR time-series dataset is used [2]. This repository contains contributed labeled time-series datasets from different fields, such as: shape identification on images, time-series extracted from physical process, or even synthetic data. All datasets contain different number of classes, observations, and lengths. Moreover, it is assumed to be used on both classification and clustering tasks. As recommended in UCR, we test the 1-Nearest Neighbor - 1-NN classifier using the Euclidean distance as benchmark. UCR databases are divided into training and testing sets. In this case, AGB is employed to compute the affinity matrix $\boldsymbol{\Omega}$ over the training set, which is employed as features in the 1-NN classifier. So, given a new sample $\boldsymbol{x}_{new}$ (testing set), the similarity among $\boldsymbol{x}_{new}$ and the training set is calculated using the AGB kernel band-width. Then, the 1-NN estimated testing set labels are used to compute the system performance. Also, 7-NNSC approach is used to compare the performance of the proposed algorithm. The attained time-series classification results are presented in Table 1.

4 Discussion

Taking into account the synthetic clustering results, from Fig. 2 it can be seen how the proposed AGB methodology is able to find a suitable kernel function, i.e. Gaussian kernel band-width, which allows to build the graph $\boldsymbol{G}$ over the input data, identifying the complex synthetic dataset structures. Note that, even when some dataset are composed by disjoints data structures, with different properties, our algorithm allows to find a complete graph that encodes the main

[2] `http://www.cs.ucr.edu/~eamonn/time_series_data/`



Original

Hand-labeled

7-NNSC

AGB

(a) *Bears*

(b)

(c) $NPR = 0.63$

(d) $NPR = 0.69$

(e) *Horses*

(f)

(g) $NPR = 0.63$

(h) $NPR = 0.71$

(i) *Starfish*

(j)

(k) $NPR = 0.78$

(l) $NPR = 0.75$

(m) *Woman*

(n)

(o) $NPR = 0.76$

(p) $NPR = 0.58$

(q) *Wall*

(r)

(s) $NPR = 0.65$

(t) $NPR = 0.68$

(u) *Flowers*

(v)

(w) $NPR = 0.72$

(x) $NPR = 0.77$

Fig. 3. Images segmentation results

Table 1. Time-series classification results - testing set accuracy percentage

Dataset	Benchmark	7-NNSC	AGB	Dataset	Benchmark	7-NNSC	AGB
synthetic control	88.00	**99.33**	**98.33**	OSULeaf	51.65	47.52	**54.55**
Gun Point	**91.33**	66.67	86.00	50words	63.08	51.87	**63.52**
ECG200	**88.00**	79.00	**88.00**	Trace	76.00	53.00	**77.00**
FaceAll	**71.36**	35.50	67.28	wafer	**99.55**	32.17	99.43
SwedishLeaf	78.88	71.04	**81.44**	Lighting2	**75.41**	67.21	**75.41**
CBF	85.22	57.00	**91.67**	Lighting7	57.53	42.47	**63.01**
Coffee	**75.00**	50.00	71.43	Adiac	**61.13**	37.85	56.27
OliveOil	**86.67**	73.33	80.00	FISH	**78.29**	58.86	72.00
Two Patterns	**90.67**	48.25	90.47	Beef	**53.33**	36.67	46.67
yoga	**83.03**	52.37	79.47	FaceFour	78.41	37.50	**80.68**

relationships among samples, as can be visually corroborated in Figs. 2(a), 2(b), and 2(c). Namely, Fig. 2(b) and Fig. 2(c) describes how AGB performance is in agreement with a benchmark approach presented in [3].

Regarding to the images segmentation results described in Fig. 3, overall, our algorithm obtains a better performance in comparison with the benchmark method 7-NNSC. Particularly, for Bears, Horses, Wall, and Flowers AGB is able to find the local and global relationships among samples, highlighting the main details of each cluster. Due to each pixel is modeled with the largest linear neighborhood around it, the whole image structure is properly revealed by the estimated graph representation. However, for Starfish and Woman AGB obtains a lower performance than 7-NNSC, which can be explained by the fact that such images contain many details, that could be hand-labeled subjectively. For example, the Woman image AGB segmentation is smoother than the 7-NNSC, which is biased by abrupt changes. Even though the NPR values are higher for the Woman and Starfish 7-NNSC segmentations, the obtained AGB results are visually acceptable. In addition, because of 7-NNSC employs a fixed neighborhood size for all the samples, it is sensitive to outliers, thus is, noisy data structures. Moreover, 7-NNSC can no longer be framed as a suitable kernel based representation from a theoretical view, as explained in section 2.2. In these experiments, we also demonstrated that such drawback is also revealed in practice.

Finally, from the time-series classification results (Table 1), even though AGB based approach does not overcome the baseline results for all the provided datasets, it achieves competitive results. For example, for synthetic control, ECG200, SwedishLeaf, CBG, OSULeaf, 50words, Trace, Lighting2, Lighting7, and FaceFour dataset our approach attained the best performance. Again, the AGB local and global analysis encoded into the neighborhood size estimation allows to deal with the complex relationships among time-series. Now, 7-NNSC based classification is not able to unfold the complex data structures, because such technique assumes an unique neighborhood size. It is important to note that some of the time-series datasets are composed for many classes, which can not be suitable modeled by one kernel function, being necessary to extend the data structure analysis considering different affinity matrices.

5 Conclusions

A methodology to build automatically a graph representation over the input data for spectral clustering based approaches was proposed. For such purpose, a data structure analysis is performed using the Euclidean and geodesic distances to identify the linear and density properties of each sample neighborhood. Thus, the local and global properties of the data are revealed to estimate a suitable kernel function, which is used to construct a data graph representation. Our approach, AGB, was tested over synthetic and real-world data. Attained results showed how our approach achieved good results for clustering, image segmentation, and even classification tasks. A benchmark approach 7-NNSC, which aims to make a local scaling analysis to build the graph, was also tested. However, 7-NNSC is not able to unfold complex data structures in many cases. Such issues were demonstrated from both theoretic and experiments. As future work, it would be interesting to deal with multi-kernel methods for finding a suitable graph representation that allows to deal with non-stationary signals. Furthermore, it would be interesting to test different data model for building the graph and other association measures could be tested to highlight different data properties, e.g., entropy and rank based correlations.

Acknowledgements. Under grants provided by *Jóvenes Investigadores e Innovadores - 2012*, and a Ph.D. scholarship funded by Colciencias.

References

1. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. Pattern Recognition 41(1), 176–190 (2008)
2. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems 2, 849–856 (2002)
3. Perona, P., Zelnik-Manor, L.: Self-tuning spectral clustering. Advances in Neural Information Processing Systems 17, 1601–1608 (2004)
4. Pokharel, R., Seth, S., Príncipe, J.: Additive kernel least mean square. In: IJCNN (2013)
5. Liping, C., Xuchuan, Z., Jiancheng, S.: The approach of adaptive spectral clustering analyze on high dimensional data. In: ICCIS, pp. 160–162 (2010)
6. Garcia, C., Flenner, A., Percus, A.: Multiclass semi-supervised learning on graphs using ginzburg-landau functional minimization. In: Pattern Recognition Applications and Methods (2013)
7. Kontschieder, P., Donoser, M., Bischof, H.: Beyond pairwise shape similarity analysis. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009, Part III. LNCS, vol. 5996, pp. 655–666. Springer, Heidelberg (2010)
8. Álvarez, A., Valencia, J., Daza, G., Castellanos, G.: Global and local choice of the number of nearest neighbors in locally linear embedding. Pattern Recognition Letters 32(16), 2171–2177 (2011)
9. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 929–944 (2007)