Image Segmentation by Pairwise Nearest Neighbor Using Mumford-Shah Model

Nilima SHAH[a,b], Dhanesh PATEL[a] and Pasi FRÄNTI[b,1]

[a] *The M.S. University of Baroda, Vadodara, Gujarat, India*
[b] *University of Eastern Finland (UEF), Joensuu, Finland*

Abstract. Mumford-Shah model has been used for image segmentation by considering both homogeneity and the shape of the segments jointly. It has been previously optimized by complex mathematical optimization methods like Douglas-Rachford, and a faster but sub-optimal k-means. However, they both suffer from fragmentation caused by non-convex segments. In this paper, we present hierarchical algorithm called Pairwise nearest neighbor (PNN) to optimize the Mumford-Shah model. The merge-based strategy utilizing the connectivity of the pixels prevents isolated fragments to be formed, and in this way, reaches better quality in case of images containing complex shapes.

Keywords. Image segmentation, Mumford-Shah, agglomerative clustering, PNN

1. Introduction

Image segmentation is a fundamental pre-processing task applied in more complex machine vision applications. The goal is to find regions of interest in an image that possess homogeneity and spatial connectivity. Let function $h(x, y) \in R^3$ represent RGB color of a pixel in the image at location (x, y). We aim to find k segments $\Omega_1, \Omega_2, \dots \Omega_k$ so that the pixel colors or some other feature would have smooth variation within the segments but discontinuous over the segment borders. We characterize the problem in terms of approximation theory, as the problem of finding an approximation f of h so that each f_i represents the segment Ω_i via a piecewise smooth function.

Existing approaches to image segmentation [1] include thresholding, clustering and classification methods [2], [3], [4], region-based methods [5], and edge based active contour methods [6], [7], [8]. The segmentation problem has also been dealt as an energy minimization problem. Obtaining a generalized solution that works without prior knowledge about the objects, their characteristics like shape, color, texture, appearance of shadows and overlapping of objects, is still an open problem.

Clustering has also been applied to image segmentation via grouping the pixels by minimizing intra segment similarity using k-means algorithm [9]. K-means is known to be sensitive to initialization, but it can be improved significantly by better initialization technique and by repeating the algorithm 100 times [10]; or by using random swap algorithm [11] which practically never gets stuck to an inferior local minimum.

---

[1] Corresponding Author, Pasi FRÄNTI, University of Eastern Finland (UEF), Joensuu, Finland; E-mail: pasi.franti@uef.fi.

K-means and random swap minimize sum-of-squared error (*sse*) between the pixels and centroids:

$$sse = \frac{1}{n}\Sigma_i\|f - h\|^2 \tag{1}$$

However, minimizing *sse* leads to color quantization where every pixel is mapped to its nearest color regardless of its neighbors resulting in a fragmented segmentation. This is typically solved by adding spatial constraint to the equation or by using explicit convexity criterion in the algorithm.

Mumford-Shah is an energy minimization model that has been applied for image segmentation [12]. Its simplified version is defined as follows:

$$E(f,\tau) = \int \int_\Omega \|f - h\|^2 dxdy + \lambda L(\tau) \tag{2}$$

where the first term equals to *sse*, $L(\tau)$ denotes the total length of the segment boundaries, and λ is a control parameter. Large values of λ will produce shorter boundaries whereas smaller values of λ will put more emphasis on the homogeneity of the segments. The Mumford-Shah model has been extensive used in the areas of image denoising, segmentation and restoration [13].

Algorithmic techniques to minimize the Mumford-Shah includes Douglas-Rachford algorithm [14], implicit methods [15], and k-means [16]. The last one is virtually as fast as standard k-means and significantly faster than the other optimization techniques using Mumford-Shah model [15], [17], [18], [19], [20], [21]. However, the algorithm itself does not control the convexity and it can still lead to fragmentation with isolated sub-segments.

In this paper, we propose hierarchical merge-based clustering algorithm for optimizing Mumford-Shah model. It allows only neighbor segments to be merged, and in this way, prevents isolated sub-segments to be formed. Classical agglomerative clustering itself is a slow algorithm, but we show that utilizing the 2-D neighborhood we can reach $O(n^2)$ time complexity by rather straightforward implementation.

2. Pairwise nearest neighbor using Mumford-Shah

Agglomerative clustering [22], [23] is a popular alternative to k-means but a simple implementation can be several orders of magnitude slower. Initially, each pixel forms its own segment s_i. Two segments are then repeatedly merged until the desired number of segments is reached. The segment pair to be merged are selected using the Ward's method [24] so that the increase in *sse* is the least among all possible merges. Thus, it generates clusters by a sequence of locally optimal merge operations.

2.1. Pairwise nearest neighbor method

In the context of vector quantization, agglomerative clustering is known as the pairwise nearest neighbor (PNN) method due to Equitz [25]. Many variants of PNN exist. Most of them try to speed up the algorithm while some also try to improve the clustering quality. We next give a brief literature review of the agglomerative clustering.

A straightforward implementation of the PNN algorithm takes $O(n^3)$ time because there can be n steps in total, and every step takes $O(n^2)$ time. Even if the merge costs were stored in a matrix, the search for the best merge would still require exhaustive

search [26]. In [27] an alternative variant called fast exact PNN was proposed to reduce the time complexity to $O(\tau n^2)$, where τ is significantly smaller than n with all realistic data sets. The main idea is to maintain a nearest neighbor to store the lowest merge cost for each segment to avoid repeating unnecessary calculations [25]. The method is exact in a sense that it does not compromise the accuracy of the PNN.

In [28], an improved variant called Lazy-PNN was developed. The running time of the algorithm is dominated by $O(nk)$ time operations [25], [29]. However, some of the merge cost calculations can be delayed and a remarkable number of calculations can be avoided completely. Practical tests indicated that the lazy-PNN is about 35% times faster than the fast exact PNN [30], and 200-500 times faster than the original PNN [23], [30].

In [31], the fast exact PNN was also applied to multilevel nonparametric image thresholding, achieving $O(nlogn)$ but this does not generalize to segmentation in RGB space. In [32], the algorithm uses a heap data structure in which all pairwise merge costs are stored and the smallest merge cost is always found as the element of the root node of the heap. The computation time of the algorithm is at most $O(n^2 logn)$ at the cost of $O(n^2)$ memory consumption, which is not suitable for segmenting large images.

In [33], a more general approach called iterative shrinking is proposed. Instead of merging, the number of clusters is reduced by a sequence of cluster removal operations. Clusters are removed one at a time by reassigning the objects from the removed cluster to the nearby clusters. The PNN method can be considered as a special case of the iterative shrinking, in which the objects of the removed cluster are all forced to move to the same cluster selected for the merge [25].

In [34], an approximate k-nearest neighbor graph is used for reducing the number of merge cost calculations. The graph is utilized so that the search for the cluster with the smallest merge cost is limited only to the clusters that are directly connected by the graph. This reduces the time complexity from $O(n)$ to $O(k)$ for a single node. The graph construction becomes then the bottleneck but fast approximate variants using k-d tree, divide-and-conquer or projection-based search were considered in [35]. The time complexity of the algorithm can be improved accordingly from $O(\tau n^2)$ to $O(\tau n \text{Log } n)$ at the cost of slight increase in *sse* [22], [35]. All the above variants of agglomerative clustering aim at faster speed except the iterative shrinking which aims at better quality.

2.2. Adopting Mumford-Shah to PNN

In this paper, we adopt the PNN variant from [34] to image segmentation using Mumford-Shah model. The main difference is that we take spatial connectivity of the pixels into account instead of merely minimizing *sse*.

The basic structure of the PNN method is shown in Algorithm 1. Given a set of n pixels $X = \{x_1, x_2, \dots x_n, \}$, the method starts by assigning each pixel x_i to its own segment represented by centroid c_i. In each step, the number of segments is reduced by merging two segments *a* and *b*. The merge cost is calculated as follows:

$$cost_{i,nb} = \frac{n_i n_{nb}}{n_i + n_{nb}} \|c_i - c_{nb}\|^2 - \lambda \cdot 2 \cdot common_{i,nb} \qquad (3)$$

The first term calculates how much *sse* is increased by the merge, and second term how much the boundary length is reduced. It can be calculated as follows:

$$common_{i,nb} = (blen_i + blen_{nb} - blen_{i+nb})/2 \qquad (4)$$

where $blen_i$ is the boundary length of the segment i. The PNN method applies a local optimization strategy: all possible cluster pairs are considered and the one increasing the Mumford-Shah model according to (3) is always chosen:

$$a, b = \arg \min_{\substack{i,j \in [1,k] \\ i \neq j}} cost_{i,j} \tag{5}$$

Algorithm 1: Pairwise nearest neighbor

$PNN(X, k) \rightarrow S$
    FOR $i \leftarrow 1$ to n DO
        $S_i \rightarrow \{x_i\}$;
    REPEAT
        $(S_a, S_b) \leftarrow$ SearchNearestClusters(S)
        Merge(S_a, S_b)
    UNTIL $|S| = k$

2.3. Implementation of Mumford-Shah PNN

The proposed algorithm follows the basic structure of the PNN method in [34] with two main differences. First, we use merge cost derived from Mumford-Shah model. Second, since we operate on image pixels, we can implement the search for the nearest neighbors more efficiently by utilizing the connectivity of the segments.

In the beginning, each pixel forms its own segment. The centroid of the segment is the pixel value as such, $c_i = x_i$, and the segment size is set to $csize_i = 1$. An array of its four neighbors is recorded (nb_i). For each segment, we also maintain its nearest neighbor (nn_i) and the corresponding merge cost value (nnc_i). These are found by calculating the merge cost (3) of the segment and with its all neighbors. We explain next how these data structures are updated during the merge process.

The next merge is always found by linear search among all the nearest neighbors. In case the merge cost of two or more neighbors are equal, we select the neighbor as per the row-major order. While the merging process, the algorithm considers only spatially connected neighbors. This prevents fragmentation to appear which happens with the k-means variants that only minimizes (3) without explicit connectivity constraint.

We maintain a heap structure of all the segments according to the descending order of nnc. After the merge, the centroid of the merged cluster is updated as:

$$c_i \leftarrow \frac{c_i \cdot csize_i + c_{nb_i} \cdot csize_{nn_i}}{csize_i + csize_{nn_i}} \tag{6}$$

The boundary length is also updated using (4), and the list of neighbors of cluster i is updated by selecting the unique neighbors of the two merged clusters avoiding duplicates. The nearest neighbor of the merged cluster i is then resolved by recalculating the merge cost to all its neighbors.

We reach slightly better time complexity than in [27], $O(n^2) < O(\tau n^2)$ although the difference is not huge because τ is data dependent variable which is usually small. In comparison to [34], our algorithm is still slower as $O(n\log n) < O(n^2)$. The main reason is that the original PNN does not need to maintain the partition whereas we need to update the pixel labels in order to calculate the boundary length. The Mumford-Shah PNN is slower than the Mumford-Shah k-means. This is the price we need to pay for the better segmentation quality.

3. Experiments

We compare the proposed method (MS-PNN) with two other methods: *regularized k-means* (reg-KM) [36], and *Mumford-Shah k-means* (MS-KM) [16]. The experiments were conducted on desktop with Intel Core i5 processor with 2.50GHz speed, 8 GB RAM, 64-bit Windows 10 operating system. To evaluate the quality of the segmentations we used two measures: *bi-directional consistency error* (*BCE**) [37] and *structural similarity index* (SSIM) [38]. As test set, we use 11 images from the Weizmann dataset [37] which has human segmented ground truths.

The results are summarized in Table 1. They show that the proposed method outperforms the other methods by clear margin. The average BCE* values are 0.33 (MS-PNN), 0.38 (MS-KM), 0.46 (KM) and 0.47 Reg-KM. The SSIM values are best among 9 out 11 images. The better quality comes at the cost of higher processing time; 476s for MS-PM compared to, 166s (Reg-KM) and <1s (KM and MS-KM).

The visual quality in Fig. 1 shows the main benefit of MS-PNN; it is the only method without isolated fragments which is observed in all k-means variants. While k-means is reasonably good in optimizing the Mumford-Shah model, these isolated sub-segments are persistent and only the merge-based algorithm can avoid them completely.

The main limitation of the algorithm is the λ parameter of the Mumford-Shah model which needs to be manually tuned. Fig. 2 shows it effects on few sample images. Here we have used the values of λ=0.25 and λ=0.50.

Table 1. BCE* (the lower the better) and SSIM (the higher the better) for the three methods.

Image	BCE*				SSIM			
	KM	Reg-KM	MS-KM	MS-PNN	KM	Reg-KM	MS-KM	MS-PNN
1	0.22	0.65	0.54	0.05	0.77	0.57	0.82	0.96
2	0.73	0.72	0.44	0.46	0.43	0.43	0.54	0.74
3	0.53	0.57	0.42	0.51	0.52	0.52	0.63	0.72
4	0.05	0.03	0.31	0.03	0.95	0.95	0.81	0.95
5	0.18	0.23	0.24	0.08	0.80	0.80	0.82	0.94
6	0.76	0.65	0.52	0.62	0.59	0.59	0.70	0.56
7	0.44	0.20	0.20	0.16	0.61	0.77	0.78	0.89
8	0.57	0.58	0.23	0.21	0.42	0.42	0.83	0.82
9	0.44	0.58	0.41	0.33	0.57	0.57	0.72	0.80
10	0.61	0.53	0.50	0.79	0.46	0.46	0.46	0.60
11	0.54	-	0.35	0.38	0.41	-	0.50	0.66
Average	0.46	0.47	0.38	**0.33**	0.59	0.61	0.69	**0.79**

4. Conclusions

We have introduced a new approach by embedding the well-known Mumford-Shah model into the merge-based hierarchical PNN algorithm. The results of the proposed MS-PNN are much better than the k-means variants in terms of segmentation quality (BCE*) and reconstruction quality (SSIM) of the images. The resulting segment boundaries of MS-PNN are much smoother without fragmentation. The drawback of the algorithm is slower than the corresponding k-means variant using Mumford-Shah but this can be tolerated as much better segmentation quality is obtained. Furthermore, it is quite possible to speed-up the method further from $O(n^2)$ to $O(n\log n)$ by utilizing better data structures. This is our future work.
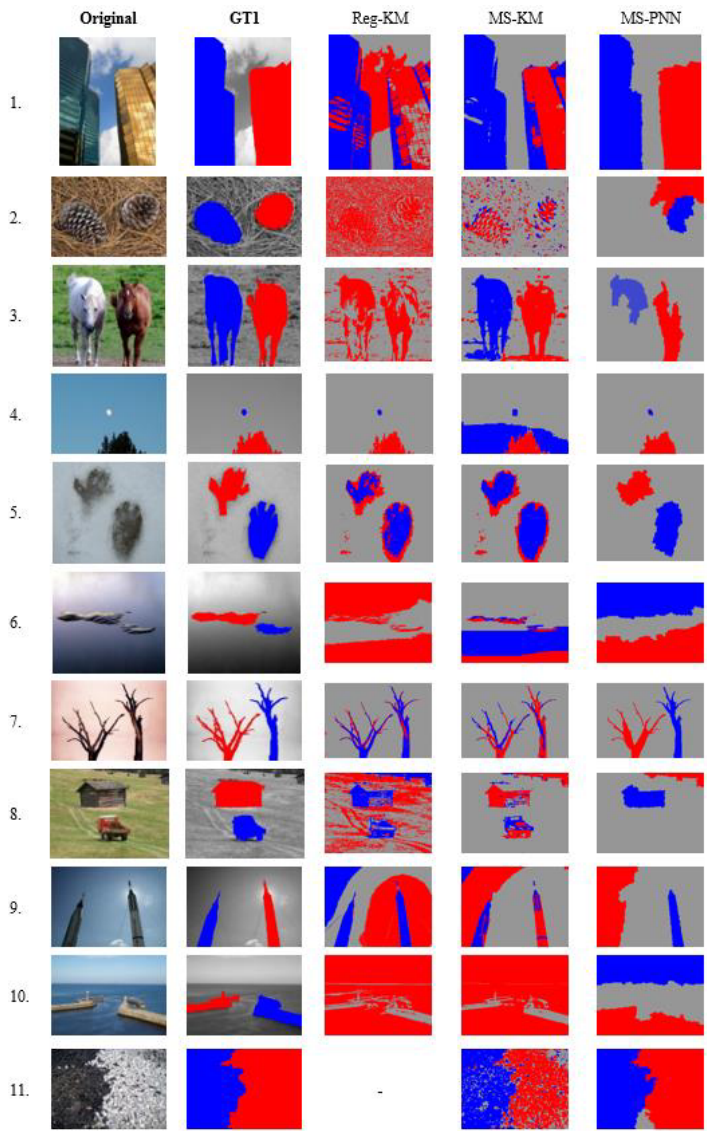
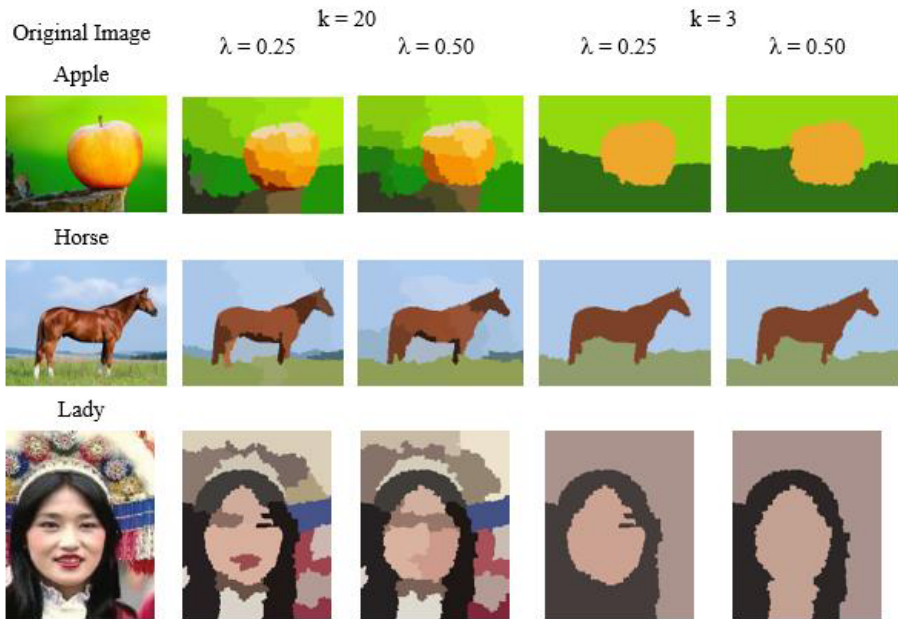**Figure 1.** Visual comparison of the segmentation results using three methods.

Figure 2. Examples of the effect of λ on the segmentation result.

References

[1] N. Shah, D. Patel and A. Jivani, "Review on image segmentation, clustering and boundary encoding," International Journal of Innovative Research in Science, Engineering and Technology, vol. 2, pp. 6309-6314, 2013.

[2] Y. Yang and S. Huang, "Image segmentation by fuzzy c-means clustering algorithm with a novel penalty term," Computing and Informatics, vol. 26, pp. 17-31, 2007.

[3] Y. Tolias and S. Panas, "Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions," IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 28, pp. 359-369, 1998.

[4] F. Gibou and R. Fedkiw, "A Fast Hybrid k-Means Level Set Algorithm For Segmentation," 4th Annual Hawaii International Conference on Statistics and Mathematics, 8 2003.

[5] J. Martin, "Implementing the region growing method Part 1: The piecewise constant case," 2002.

[6] T. Chan and L. Vese, "An active contour model without edges," in Proc. Scale-space theories in computer vision. Scale-Space 1999, Lecture Notes in Computer Science, Springer, 1999.

[7] T. Chan and L. Vese, "Active contours without edges," IEEE Transactions on Image Processing, vol. 10, pp. 266-277, 2 2001.

[8] T. Chan and L. Vese, "A level set algorithm for minimizing the Mumford-Shah functional in image processing," in Proc. Variational and Level Set Methods in Computer Vision Workshop, 2001.

[9] V. K. Dehariya, S. K. Shrivastava and R. C. Jain, "Clustering of Image Data Set Using K-Means and Fuzzy K-Means Algorithms," in 2010 International Conference on Computational Intelligence and Communication Networks, 2010.

[10] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?," Pattern Recognition, vol. 93, pp. 95-112, 2019.

[11] P. Fränti and S. Sieranoja, "K-means Properties on Six Clustering Benchmark Datasets," Applied Intelligence, vol. 48, pp. 4743-4759, 12 2018.

[12] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," Communications on Pure and Applied Mathematics, vol. 42, pp. 577-685, 1989.

[13] L. Bar, T. Chan, G. Chung, M. Jung, N. Kiryati, R. Mohieddine, N. Sochen and L. Vese, Mumford and Shah Model and its applications to image segmentation and image restoration, 2 ed., Springer Science+Business Media LLC, 2011, pp. 1097-1154.

[14] N. Shah, D. Patel and P. Fränti, "Fast Mumford-Shah two phase image segmentation using proximal splitting scheme," *Journal of Applied Mathematics*, vol. 2021, 6618505, 13 pages, 2021.

[15] L. Ambrosio and V. Tortorelli, "Approximation of functional depending on jumps by elliptic functional via t-convergence," Communications on Pure and Applied Mathematics, vol. 43, pp. 999-1036, 12 1990.

[16] N. Shah, D. Patel and P. Fränti, "K-means image segmentation using Mumford-Shah model," Journal of Electronic Imaging, 2021. (submitted)

[17] E. Brown, T. Chan and X. Bresson, "Convex formulation and exact global solutions for multi-phase piecewise constant Mumford-Shah image segmentation," 2009.

[18] X. Cai, R. Chan and T. Zeng, "Image segmentation by convex approximation of the Mumford-Shah model," UCLA CAM Report, 2012.

[19] W. Fleming and R. Rishel, "An integral formula for total gradient variation," Arch. Math., vol. 11, pp. 218-222, 1960.

[20] G. Koepfler, C. Lopez and J. Morel, "A multiscale algorithm for image segmentation by variational method," SIAM Journal of Numerical Analysis, vol. 31, pp. 282-299, 1994.

[21] T. Pock, A. Chambolle, D. Cremers and H. Bischof, "A convex relaxation approach for computing minimal partitions," IEEE Conference on Computer Vision and Pattern Recognition, pp. 810-817, 2009.

[22] S. W. Ra and J.-K. Kim, "A fast mean-distance-ordered partial codebook search algorithm for image vector quantization," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, pp. 576-579, 1993.

[23] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies: 1. Hierarchical systems," The computer journal, vol. 9, pp. 373-380, 1967.

[24] J. H. Ward, "Hierarchical grouping to optimize an objective function," Journal of the American Statistical Association, vol. 58, pp. 236-244, 1963.

[25] W. H. Equitz, "A new vector quantization clustering algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, pp. 1568-1575, 1989.

[26] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression, Springer, Boston, MA, 1992.

[27] P. Fränti, T. Kaukoranta, D.-F. Shen and K.-S. Chang, "Fast and Memory Efficient Implementation of the Exact PNN," IEEE Transactions on Image Processing, vol. 9, pp. 773-777, 2000.

[28] T. Kaukoranta, P. Fränti and O. Nevalainen, "Vector quantization by lazy pairwise nearest neighbor method," Optical Engineering, vol. 38, pp. 1862-1868, 1999.

[29] D.-F. Shen and K.-S. Chang, "Fast PNN algorithm for design of VQ initial codebook," in Visual Communications and Image Processing'98, 1998.

[30] P. Fränti and T. Kaukoranta, "Fast implementation of the optimal PNN method," in Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269), 1998.

[31] O. Virmajoki and P. Fränti, "Fast pairwise nearest neighbor based algorithm for multilevel thresholding," Journal of Electronic Imaging, vol. 12, pp. 648-660, 2003.

[32] T. Kurita, "An efficient agglomerative clustering algorithm using a heap," Pattern Recognition, vol. 24, pp. 205-209, 1991.

[33] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," Pattern Recognition, vol. 39, pp. 761-775, 2006.

[34] P. Fränti, O. Virmajoki and V. Hautamaki, "Fast agglomerative clustering using a k-nearest neighbor graph," IEEE transactions on pattern analysis and machine intelligence, vol. 28, pp. 1875-1881, 2006.

[35] O. Virmajoki and P. Fränti, "Divide-and-conquer algorithm for creating neighborhood graph for clustering," in Proceedings of the 17th International Conference on Pattern Recognition. ICPR 2004., 2004.

[36] S. Kang, B. Sandberg and A. Yip, "A regularized k-means and multiphase scale segmentation," 2011.

[37] D. Martin, C. Fowlkes, D. Tal and J. Malik, "A Database of Human Segmented Natural Images and its Application to evaluating seg algo and measuring ecological statistics," in Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.

[38] A. C. Brooks, X. Zhao and T. N. Pappas, "Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortions," IEEE Transactions on Image Processing, vol. 17, pp. 1261-1273, 2008.