Color Image Segmentation:
From the View of Projective Clustering

Song Gao, University of Alabama at Birmingham, USA

Chengcui Zhang, University of Alabama at Birmingham, USA

Wei-Bang Chen, Virginia State University, USA

ABSTRACT

An intuitive way of color image segmentation is through clustering in which each pixel in an image is treated as a data point in the feature space. A feature space is effective if it can provide high distinguishability among objects in images. Typically, in the preprocessing phase, various modalities or feature spaces are considered, such as color, texture, intensity, and spatial information. Feature selection or reduction can also be understood as transforming the original feature space into a more distinguishable space (or subspaces) for distinguishing different content in an image. Most clustering-based image segmentation algorithms work in the full feature space while considering the tradeoff between efficiency and effectiveness. The authors' observation indicates that often time objects in images can be simply detected by applying clustering algorithms in subspaces. In this paper, they propose an image segmentation framework, named Hill-Climbing based Projective Clustering (HCPC), which utilizes EPCH (an efficient projective clustering technique by histogram construction) as the core framework and Hill-Climbing K-means (HC) for dense region detection, and thereby being able to distinguish image contents within subspaces of a given feature space. Moreover, a new feature space, named $HSV_rV_gV_b$, is also explored which is derived from Hue, Saturation, and Value (HSV) color space. The scalability of the proposed algorithm is linear to the dimensionality of the feature space, and our segmentation results outperform that of HC and other projective clustering-based algorithms.

IGI GLOBAL PROOF

Keywords: *Clustering-Based Algorithms, Color-Based Image Segmentation, Hill-Climbing K-Means Algorithm, Hue-Saturation-Value (HSV), Projective Clustering*

1. INTRODUCTION

Image segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze (Shapiro & Stockman, 2011). Pixels within a segment have high coherence with respect to certain features, such as color, texture, intensity, and spatial information, while pixels from different segments have significant difference in the same feature space or subspace of the space. Image segmentation has been widely applied in a wide spectrum of applications. For example, in the medical field, digital images of histological slides can be used to classify skin biopsies as either melanoma or nevi (Osborne, Gao, Chen, Andea & Zhang, 2011). Image segmentation as the first step is performed to recognize different tissues in the slide. Another example is the *Content-based Image Retrieval* (CBIR). Users

DOI: 10.4018/jmdem.2012070104

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

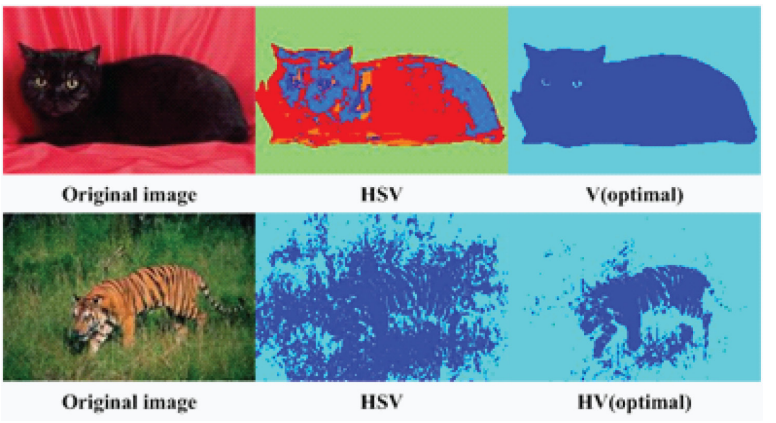
can submit a query based on an uploaded image, or further filter the results, which are returned by the keywords searching, according to their contents. Image segmentation partitions an image into more meaningful contents as "visual keywords" to allow for more powerful queries.

Clustering (Everitt, 2012; Jain, Murty & Flynn, 1999) provides a good way for image segmentation since pixels within the same segment share some common characteristics. The most well-known and classical partitioning algorithm is *K*-means (Han & Kamber, 2005; MacKay, 2003). Given a set of data points $X = (X_1, X_2, \ldots, X_n)$, where each data point is a *d*-dimensional vector. The goal of *K*-means is to partition the *n* data points into *K* groups ($K \leq n$) $G = (G_1, G_2, \ldots, G_K)$ so as to minimize the within-cluster sum of squares (criterion function), namely $\operatorname{argmin}_G \sum_{i=1}^{K} \sum_{x_j \in G_i} \left\| x_j - m_i \right\|^2$, where m_i is the mean of cluster G_i. First, it randomly selects *K* data points as the initial means or centers of *K* clusters. Then, the rest of data points are assigned to the most similar clusters, based on a similarity measure (e.g., Euclidean distance) between data points and cluster means. The new mean (center) of each cluster is updated based on assigned data points to the cluster. This process iterates until the criterion function converges or is smaller than a threshold. Most clustering algorithms in image segmentation domain work on the full feature space including single or multiple modalities. Spectral clustering (Ng, Jordan, & Weiss, 2001; Shi & Malik, 2000) starts from transforming the original data matrix into the eigenvectors of matrix, and then detects *k* well-separated clusters on the surface of the *k*-sphere. Hill-climbing *K*-means algorithm (HC) (Ohashi, Aghbari & Makinouchi, 2003) initially discretizes the full feature space (e.g., LUV or HSV color space) in order to find the local maxima by using the hill-climbing technique (Russell & Norvig, 2003). Each local maximum is treated as one of the initial cluster seeds which are the input for the later *K*-means clustering. On one hand, HC can automatically determine the number of clusters, and effectively generate the segmentation result. On the other hand, HC is a global method, so it cannot find clusters that are best represented in subspaces as shown in Figure 1.

Projective clustering attempts to assign each point in the feature space to a unique cluster, but clusters may exist in different subspaces. A projected cluster (Aggarwal & Yu, 2000; Ng, Fu, & Wong, 2005) is defined as *a set ε of vectors together with a set* C *of data points such that the points in* C *are closely clustered in the subspace defined by the vectors ε. The subspace defined by the vectors in ε may*

IGI GLOBAL PROOF

Figure 1. Optimal segmentation results from certain subspace of HSV by using HC algorithm



Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

have much lower dimensionality than the full dimensional space. PROCLUS (Aggarwal, Wolf, Yu, Procopiuc, & Park, 1999) can detect interesting patterns in subspaces with the spread direction parallel to the original axes, while ORCLUS (Aggarwal & Yu, 2000) is applicable in arbitrary spread direction. Both algorithms require the number of expected clusters as an input parameter. The average number of dimensions for the clusters and the size of the subspace dimensionality are also required by PROCLUS and ORCLUS, respectively. In EPCH (Ng, Fu, & Wong, 2005), fixed bin width histograms are constructed to generate "signature," where a signature corresponds to a set of dense regions in a set of subspaces, and signatures with regions covering a large enough number of data objects are identified as subspace clusters. Compared with the above two algorithms, EPCH requires less prior knowledge on the dataset. A general user only needs to provide the maximum number of clusters the user is interested to uncover. Some tuning parameters are configured with default values. REVBH (Relative Entropy on Variable Bin width Histogram) (Gao, Zhang, & Chen, 2010, 2011) utilizes variable bin width histograms to describe the local data distribution instead of using fixed bin width histograms, and uses relative entropy as a measure to detect dense regions in each subspace, consequently further improving the clustering quality. EPCH and REVBH are both scalable to dimensionality of dataset. In this paper, we propose a projective clustering algorithm, HCPC (Hill-Climbing based Projective Clustering), that combines the strength of both EPCH and HC to explore meaningful regions in subspaces of a low-level visual feature space. Our main contributions are summarized as follows:

1. HCPC eliminates one important tuning parameter in EPCH, namely the upper bound f of the spread of a cluster in each projection domain that is used to adjust the global threshold for detecting dense regions in each subspace. However, the proportion of dense regions in different subspaces may be different. In HCPC, Hill-Climbing finds automatically the local maxima according to the local distribution.
2. The works presented (Gao, Zhang, & Chen, 2010, 2011; Ng, Fu, & Wong, 2005) are both histogram-based projective clustering algorithms. However, a too-fine histogram will incur expensive computations while a too-rough histogram fails to represent the data distribution. In contrast, K-means in HCPC avoids such problems by working directly on the data object level rather than the bin level.
3. HC alone cannot detect clusters in subspaces, while the framework of projective clustering in HCPC provides such a capability.
4. We also propose a new feature space, named $HSV_rV_gV_b$, derived from HSV color space. According to our observation, the new space is more suitable for subspace exploration in the domain of image segmentation as many segments can be better detected in a subspace of this new space than that in the original HSV space.

IGI GLOBAL PROOF

In the meantime, we also realize the challenges in this research field. First, many existing segmentation algorithms use visual features from multiple modalities. Their main purpose is to increase the difference between segments by accumulating difference from different modalities. However, fusing information from different modalities is challenging in itself, because irrelevant or redundant features may have negative effect on clustering. Second, searching subspaces is one way to determine the optimal subset of features for a specific segment. It is time consuming if every subspace needs to be explored. The tradeoff between efficiency and effectiveness is always challenging. And the exploration of subspaces in a multimodality space adds another level of difficulty to the already challenging problem. Therefore, in this paper we focus on visual features from one single modality only – the color space.

The rest of this paper is organized as follows. A new color space, named $HSV_rV_gV_b$, is introduced in Section 2. Section 3 includes the

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

introduction of EPCH and REVBH, and presents the improvement based on the framework of EPCH in our proposed algorithm. Experimental results are reported in Section 4. And the paper is concluded in Section 5.

2. $HSV_RV_GV_B$ COLOR SPACE

HSV color space, which stands for hue, saturation and value, is equally converted from RGB color space. The *V* value which represents the lightness of each pixel in an image is calculated as the normalization of the maximum of *R*, *G*, and *B* values:

$$V = \max\left(R, G, B\right) \tag{1}$$

However, the fact that only one of the three color channels contributes to the value of *V* is lost during conversion. Not only that, it diminishes the difference between two equal *V* values that are contributed by different color channels. Pixels from different objects may be able to be distinguished by adding such knowledge into the HSV feature space. Therefore, we redefine the HSV color space by further dividing *V* into three independent features, namely V_r, V_g, and V_b, in order to distinguish from which channel this *V* comes from. For each pixel, three *V*s are calculated as:

$$V_i = \begin{cases} V, if \;\; i == \text{argmax}(R,G,B) \\ 0, \qquad\quad otherwise \end{cases}, \quad i \in \left\{r, g, b\right\} \tag{2}$$

If the maximum *V* value of a pixel corresponds to multiple channels, it will be assigned to the first V_i channel in the order of {V_r, V_g, and V_b}.

In order to examine the influence of the new feature space on the segmentation result, we performed HC on the full space of HSV and $HSV_rV_gV_b$, respectively, on 300 natural scene images. Three users participated in the evaluation of these two groups of segmentation results. In average, similar/comparable segmentation results are observed for 75 images, which could be good or bad, from both feature spaces. Among the remaining 225 images, $HSV_rV_gV_b$ outperforms HSV on 134 images, while HSV yields a better segmentation result on 91 images on average. Some segmentation results from the two spaces are shown in Figure 2a.

We also performed HC on each subspace of HSV and $HSV_rV_gV_b$, respectively, on a small set of 120 color images, attempting to find out whether the optimal segmentation results can be obtained from certain subspace (or the full space) of $HSV_rV_gV_b$ compared with certain subspace (or the full space) in HSV. 47% of images, including 42% both good and 5% both bad results, yielded comparable segmentation results between the two spaces, while 43% of images yielded better segmentation results in certain subspace of the new feature space $HSV_rV_gV_b$. Only for 10% of all the images HSV outperformed $HSV_rV_gV_b$ on subspace segmentation.

Since the new feature space is modified from HSV, the value of H channel still ranges from 0° to 360°. However, this characteristic is not taken into full consideration when *H* value is involved in calculating the Euclidean distance between two data points. Namely, the red color (0°) and its neighbors (nearly-red, a little less than 360°) are far from each other with respect to Euclidean distance measure, but actually they are close if calculating the distance by modulus. A good example of this problem is shown in the third row of Figure 2b: the nearly-red stripe and the orange (nearly-yellow) stripe belong to different clusters in both HSV and $HSV_rV_gV_b$ feature spaces, because in the original hue space, the distance between nearly-red and nearly-yellow almost covers the whole range. However, these two colors are visually similar. Another problem caused by separating *V* channel is that when the *V* value is too low in a region, or *V* value is too high while *S* value is too low, the region is relatively "dark" or "bright," and the hue (*H*) values of those pixels in that region tend to be randomly distributed. Those pixels could be clustered into one group if *V* is not

IGI GLOBAL PROOF

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 2. (a) Sample segmentation results by applying HC on HSV and on $HSV_rV_gV_b$, respectively, (b) Segmentation examples that manifest the problems in HSV and $HSV_rV_gV_b$

IGI GLOBAL PROOF

split into three channels, while three separate *V*s just make the distribution of those pixels in the feature space even sparser. Figure 2b lists some examples that manifest the mentioned two problems. For example, in the second image of the last row, the segments with orange and khaki colors, respectively, should belong to the same cluster since most pixels from both of them represent the white background. However, the mean values from each channel of these two segments are [0.1751, 0.0475, 0.8344] in the orange segment, and [0.0023, 0.0020, 0.8779] in the khaki segment, respectively. Both segments have similar high *V* values, and low *S* values, but obviously different *H* values, and thereby resulting in different segments by using HC algorithm.

3. PROJECTIVE CLUSTERING

As discussed in Section 1, some best segmentation results may exist in subspaces of HSV or $HSV_rV_gV_b$. Therefore, our goal is to utilize projective clustering to detect segments in subspaces, in which a better representation of segments of an image can be obtained compared with that in full dimensional space, as well as achieving high scalability so that the proposed algorithm scales well up to high dimensional feature spaces.

3.1. EPCH Algorithm

Given a dataset $X = (X_1, X_2,\ldots, X_n)$, in which each data point is a D dimensional vector,

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

namely $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. In order to reduce the computational cost in exploring the full dimensional space, each data point is represented by a signature with each entry as the dense region ID in a k dimensional subspace ($k \leq D$). After such discretization, the complexity of calculating the similarity between two signatures is relatively lower compared with calculating the Euclidean distance between two points. The total number of all possible k dimensional subspaces is. When a very small k is considered (e.g., $k=1$), detected dense regions may hide the existence of noises, sparseness and/or even small dense regions in higher dimensional spaces. This happens because of possible overlap of data points when projected onto lower dimensional spaces. This hidden information can be exposed in a higher dimensional space, such as in a 2D space, but again that is not always true. Moreover, the higher the k is, the more expensive the computation is. EPCH enables us to trade a small amount of quality for much higher efficiency.

The sketch of EPCH can be described in five steps: (1) constructing k-d histograms for all possible k-d subspaces; (2) detecting dense regions in each histogram; (3) converting each data point into a signature; (4) generating cluster candidates by merging same/similar signatures; and (5) associating data points to corresponding clusters. Each step is briefly illustrated in Figure 3.

Assume each point in the 3D feature space (e.g., HSV or Lab) represents a pixel of an image. All values are normalized in the preprocessing phase. First, fixed bin width histograms are built for subspaces XY, XZ, and YZ. Each dimension has the same number of bins calculated by using Sturges' rule (Scott, 1992):

$$N_H = 1 + log_2 N \tag{3}$$

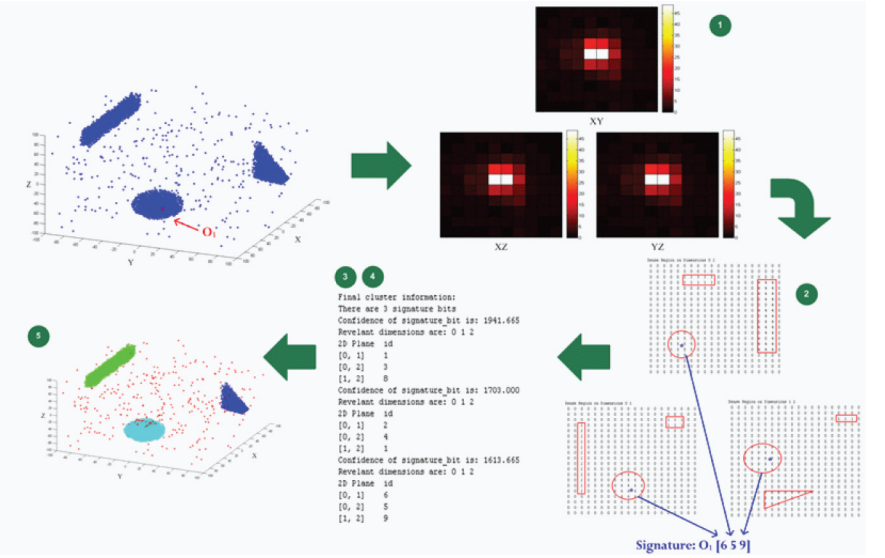
where N_H is the number of bins along one dimension, N is the total number of data points.

Assume that dense regions which usually have normal distribution are surrounded by sparse regions which have near-uniform distribution. In Step 2, dense bins in each histogram are detected if the bin height is larger than certain threshold which is defined in Equation 4.

$$t = \mu + c\sigma, c < \sqrt{1/f - 1} \tag{4}$$

IGI GLOBAL PROOF

Figure 3. The sketch of EPCH algorithm



Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

where μ and σ represents the mean and the standard deviation of bins, respectively, in the current 2D histogram. As mentioned before, f is the proportion of dense bins in each histogram, which is configured as a global parameter, thereby is c as well. The new μ, σ, and the threshold in each histogram are updated after removing dense bins from the histogram. This process iterates until no more bin's height is larger than the threshold.

In Step 3, a signature is generated for each data point. Each entry of a signature represents the ID of a dense region in a 2D histogram to which the data point is projected. If the data point drops into a non-dense region in a histogram, the corresponding entry value is 0. In (Ng, Fu & Wong, 2005), given subspaces S_1, S_2,…,S_L (where $C_D^k = L$), a signature Q_i for data object X_i is defined as an ordered list of L entries, where the j-th entry represents the dense region, if any, where the data object is located in subspace S_j. Specifically, Q_i=[Q_{i1}, Q_{i2}, …, Q_{iL}] where

IGI GLOBAL PROOF

$$Q_{ij} = \begin{cases} 0, & X_i \text{ projects into a nondense region in subspace } S_j \\ r, & X_i \text{ projects into the } r^{th} \text{ dense region in subspace } S_j \end{cases}. \quad (5)$$

Same and similar signatures are merged successively in Step 4 in order to form the description of cluster candidates. First, each signature is assigned an initial weight 1. Then, identical signatures are merged, and the total number of data points with the signature is assigned as the new weight of this signature. All distinct signatures are sorted in the descending order of their weights. The higher the weight, the more data points are in the corresponding subspace described by the signature. Merging similar signature is constrained by a similarity threshold $T_{similarity}$ defined in Equation 6.

$$T_{similarity} = \frac{|\, common\, dense\, regions \,|}{|\, distinct\, dense\, regions \,|}. \quad (6)$$

The merging starts by checking the signature with the highest weight in the signature list. If the similarity between that and one of the subsequent signatures is larger than a threshold, the two are merged. The signature with a higher weight serves as the merged signature, and another signature is removed from the list. Then, the next merging iteration proceeds until no more signatures in the list can be merged. There is an upper limit for the number of expected clusters, which is determined as a user input parameter, named "*max_no_cluster*." Only the top *max_no_cluster* cluster candidates are kept in the signature list as the projective descriptions of clusters. Because of the use of one single global threshold f (or c) in each histogram, even though two adjacent regions are both above the threshold, and one is relatively denser than the other, EPCH will treat them as one dense region. In other words, using one single global threshold is not sufficient to distinguish adjacent dense regions.

In the end, each data point is associated with one cluster with which it has the highest similarity. The similarity is calculated between the signature of a data point and the signature of a cluster. Data objects that do not belong to any cluster (the similarity is below a threshold) are treated as outliers. In image segmentation, only those data points that have similarity 0 with all clusters are treated as outliers, and they will be grouped as one "cluster." Similarity 0 with all clusters could happen because some signatures are removed from the signature list that is constrained by the "*max_no_cluster*."

3.2. REVBH Algorithm

REVBH improves EPCH in three aspects. First, variable bin-width histogram is used as the tradeoff between computational cost and accurate description of a data distribution. On one hand, even though a fine bin-width histogram can well describe the underlying distribution of a given dataset, the computation on such histogram is costly because of the large number of bins. On the other hand, a rough bin-width histogram with fewer bins may hide

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

the existence of multiple dense regions, and thereby degrading the clustering quality. In the meantime, the computational cost with fewer bins should be lower. As a tradeoff, a variable bin-width histogram has a relatively finer bin-width setting in the dense region, and uses bins with rougher bin-width to cover sparse regions. A binary tree structure, named *dimension tree* as shown in Figure 4, is used to record the division of sub-ranges in each dimension. For each dimension tree T_i, a tree node TN_{ij} records the bin information of the j-th sub-range in the i-th dimension, including the number of bins (binNum), the minValue and maxValue of that subrange, the left and the right pointers to the left/right sub-ranges (ptrLeft and ptrRight). The bin width of current sub-range can be calculated based on the above information. Initially, the whole data range on one dimension is divided into three sub-ranges with respect to the first quartile q_1 and the third quartile q_3. The bin width *mBinWidth* of the middle sub-range is calculated by using Freedman and Diaconis's rule (Freedman & Diaconis, 1981), as detailed in the following. The bin numbers of the left sub-range and the right sub-range are estimated, respectively, with respect to *mBinWidth* (estimated under fixed bin width histograms). If the estimated bin number of the current sub-range is greater than the bin number of the middle sub-range, a new division is performed on the current sub-range. The above operations are iteratively performed until no more sub-ranges can be generated.

Second, Freedman and Diaconis's rule (Freedman & Diaconis, 1981) as shown in Equation 7 is applied for calculating bin-width instead of Sturges' rule (Equation 3) which does not work well on large dataset (Scott, 2009).

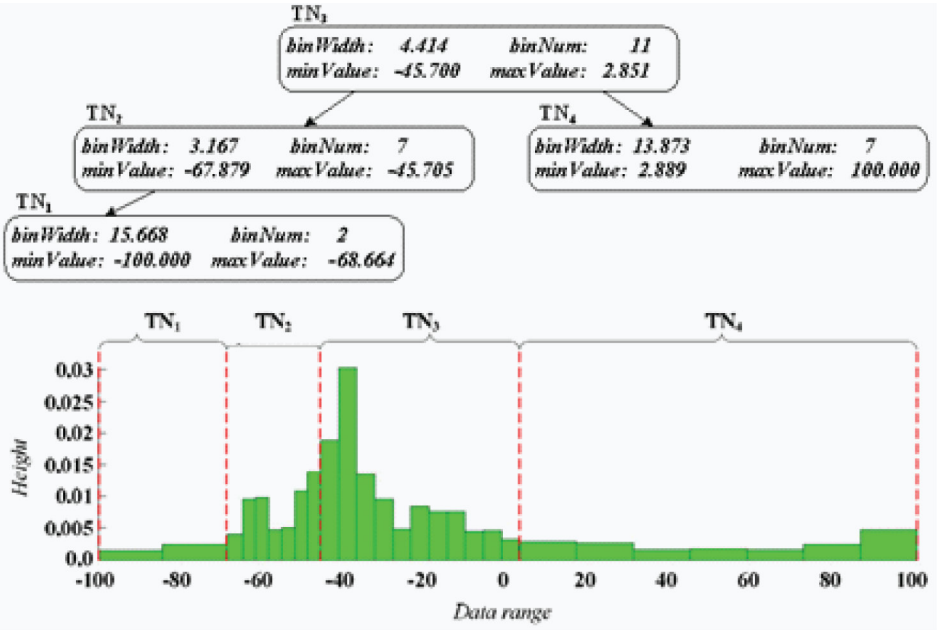
$$bw = 2 \times IQR \times n^{-1/3} \tag{7}$$

where *IQR* represents the sample interquartile range, and n is the number of samples.

Third, relative entropy (Gao, Zhang, & Chen, 2010; Kullback & Leibler, 1951) which has its own origin in probability theory and information theory is used as a measure instead of Equation 4 to detect dense bins in each his-

IGI GLOBAL PROOF

Figure 4. A sample dimension tree structure for a 1d variable bin-width histogram



Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

togram. It is a non-symmetric measure of the difference between two probability distributions P and Q. In general, P represents the real distribution of a given dataset in a feature space, while Q is the expected distribution of the same dataset in the same space. As we mentioned in Section 3.1, dense regions are usually surrounded by sparse regions with near-uniform distribution. Therefore, uniform distribution is the expected distribution in each 2D subspace if there is no dense region, i.e., the density in each bin is almost identical. During the iterative detection of dense bins, relative entropy of a histogram describes the extent to which the distribution of current bins approaches the uniform distribution. The closer a real distribution approaches the uniform distribution, the closer the relative entropy approaches zero. Equations related to calculating relative entropy of a histogram are shown as follows:

$$RE_h\left(X\right) = \sum_{i=1}^{N_b} p\left(x_i\right) log_2\left(p\left(x_i\right) / q\left(x_i\right)\right), \left|X\right| = N_b \tag{8}$$

$$p\left(x_i\right) = \frac{n_i / (n \times bw_i)}{\sum_{i=1}^{N_b}(n_i / (n \times bw_i))} = \frac{n_i / bw_i}{\sum_{i=1}^{N_b}(n_i / bw_i)} \tag{9}$$

$$q\left(x_i\right) = \frac{\frac{n(bw_i / S)}{n \times bw_i}}{\sum_{i=1}^{N_b} \frac{n(bw_i / S)}{n \times bw_i}} = \frac{1 / S}{N_b / S} = \frac{1}{N_b} \tag{10}$$

$$RE_b(x_i) = p\left(x_i\right) log_2\left(N_b \times p\left(x_i\right)\right), x_i \in X \tag{11}$$

In Equation 8, $RE_h(X)$ is the relative entropy of a histogram; X is the set of remaining bins of a histogram; N_b is the total number of remaining bins; $p(x_i)$ represents the normalized density of the i-th bin under real distribution; and $q(x_i)$ represents the normalized density of the i-th bin by assuming all samples are uniformly distributed. In Equation 9, n_i is the number of data points that are projected into the i-th bin. Since 2D histograms (for dimensions j and k) are used, $bw_i = bw_i^j \times bw_i^k$ represents the volume of the i-th bin which is related to the j-th and the k-th dimensions. Basically, $q(x_i)$ is a constant which is the reciprocal of the number of remaining bins in the current histogram. $RE_b(x_i)$ in Equation 11 is the relative entropy of the i-th bin. In each iteration, the remaining bins with RE_b larger than $(1/N_b) \times RE_h(X)$ are deemed dense and removed from the current histogram. Then, the new $RE_b(x_i)$, RE_h, and N_b are calculated based on the remaining bins.

3.3. Hill-Climbing based Projective Clustering (HCPC)

As mentioned, REVBH avoids using global parameters, such as c and f, in EPCH. However, REVBH as a histogram-based projective clustering algorithm still inherits some of the problems of EPCH. For example, adjacent dense regions which may represent different objects in an image may be detected as one dense region if the bin width is not properly selected. And REVBH also requires the user to input the maximum number of expected clusters. In order to avoid these problems and to reduce the need for a priori knowledge on the dataset, we propose a new algorithm, named Hill-Climbing based Projective Clustering (HCPC), which uses the HC algorithm instead of histogram-based methods for detecting dense regions directly applied to data points (pixels) in each 2D subspace. Because of the usage of hill-climbing technique that can automatically determine the initial number of clusters in each 2D subspace, there is less a need for a priori knowledge on the distribution of the dataset in the current subspace. The new algorithm also eliminates the need to manually configure the processes of creating histograms and detecting dense regions.

The main differences between HCPC and EPCH/REVBH are listed as follows:

IGI GLOBAL PROOF

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

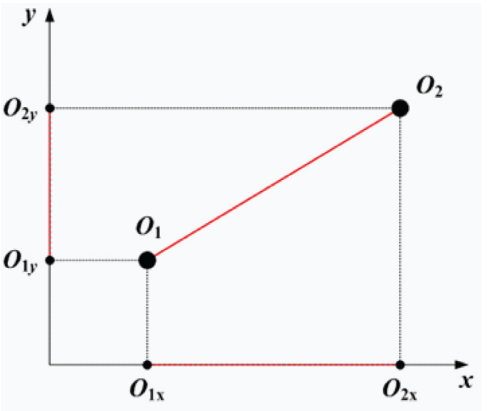
- Since *K*-means clustering is applied in the detection of dense regions, each data point in a 2D subspace will always belong to one dense region. Therefore, initially, a signature of a data point consists of detected cluster IDs from different 2D subspaces where this object belongs to. In EPCH/REVBH, a data point may drop into a non-dense region which has a signature entry as 0.
- *K*-means also generates the centroid for each dense region, therefore a *centroid vector* is defined for each signature as follows: given a signature $Q_i = [Q_{i1}, Q_{i2},\ldots, Q_{iL}]$, its corresponding centroid vector is $C_i = [C_{Qi1}, C_{Qi2},\ldots, C_{QiL}]$, where $C_{Qij} = (x_{Qij}, y_{Qij})$ is a coordinate pair that represents the centroid location of the corresponding dense region in the *j*-th 2D subspace. The purpose for defining such a structure is that when one signature has identical similarity with multiple signatures, it will merge with the "closest" one. Such closeness is measured by the *approximate distance*, which represents the maximum 2D projected Euclidean distance between two centroid vectors. As shown in Figure 5, two data points O_1 (O_{1x}, O_{1y}) and O_2 (O_{2x}, O_{2y}) live in a 2D space. The projected distance between two data points on each dimension cannot accurately reflect their real distance in the full feature space. However, a larger projected distance in general better approximates the real distance than that of a smaller projected distance (e.g., $\|O_{1x}, O_{2x}\|$ versus $\|O_{1y}, O_{2y}\|$). Equation 12 is used to calculate the approximate distance between two data points O_1 and O_2. Even though calculating the distance between data points and dense region centers in HC is computationally more expensive than that of directly detecting dense bins from histograms in EPCH/REVBH, the performance does not drop too much when the dimensionality is low (e.g., k=2). In EPCH/REVBH, the strategy for merging similar signatures in the above case is not available.

$$dist\left(Q_1, Q_2\right) = \max_{1\le j\le L} Eu\_dist(C_{Q1j}, C_{Q2j}) \quad (12)$$

- EPCH/REVBH uses an input parameter to determine the maximum number of expected clusters, which is not always available as a priori knowledge, because it is difficult in image segmentation to specify a fixed number of clusters due to vastly different scenes present in image databases.

IGI GLOBAL PROOF

Figure 5. Distance between two objects in a 2D space versus their projected distance on each 1D space



Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

In HCPC, we try to avoid using this parameter. First, the hill-climbing technique can automatically determine the initial number of dense regions in each 2D subspace. Second, a weight threshold is used to further removing signatures that contain too few data points from the list. The weight threshold can be defined as an absolute threshold (e.g., weight > 200 (data points/pixels)), or a relative threshold (e.g., weight > 0.1*max_weight). max_weight is defined as the maximum weight of all signatures in the current signature list. After merging identical signatures, among the remaining signatures, those with weight 1 are considered trivial and removed from the signature list prior to the merging of similar signature. In next step, a post-processing is performed to re-assign each trivial signature (containing one data point) to one cluster candidate that best matches the data point, and the weight of that cluster candidate in the signature list is updated accordingly. Then, signatures with weight less than the weight threshold are removed from the signature list. Finally, each data point is compared with each remaining signature again, and re-assigned to the one with maximum similarity. Compared with EPCH, one more round of association between data points and clusters are performed with the proposed algorithm.

HC works more efficiently than HCPC in lower dimensional feature spaces, such as HSV and Lab. For example, if each dimension of the HSV color space is equally partitioned into 10 sub-ranges, totally 1000 bins in the 3D histogram need to be explored by HC to determine the initial seeds. It is known that the time complexity of K-means is $O(n^{DK+1}log(n))$, where K is the number of clusters and D is the number of dimensions. When the dimensionality of the feature space grows, the time complexity of HC algorithm will increase exponentially, because hill-climbing algorithm searches the local maxima in the full dimensional grid, and K-means clustering also calculates the distance between points and seeds in the full dimensional space. Therefore, HC algorithm is not suitable for processing high dimensional dataset. On the contrary, HC in HCPC only works in each lower dimensional k-d subspace (e.g., k = 2), so the time complexity of HCPC increases linearly. When the dimensionality increases from D to D+1, the number of 2D subspaces for K-means in HCPC increases by $C_D^1 = D$, and the time complexity for K-means increases by $O(Dn^{2K+1}log(n))$. Moreover, the total number of entries in the signature list increases by nD.

4. EXPERIMENTS ON $HSV_RV_GV_B$ FEATURE SPACE

300 Corel (http://www.cs.princeton.edu/cass/benchmark/) color images of natural scenes with dimensions 192×128 or 128×192 are collected as the experimental dataset. All experiments are performed on a PC with Intel(R) Core(TM)2 CPU E7400 2.8GHz, 4GB RAM, Windows 7. The source code is implemented in MATLAB. The default similarity threshold and weight threshold are 0.6 and (0.1*max_weight), respectively, in our experiments.

IGI GLOBAL PROOF

The criterion used to evaluate the segmentation quality during the comparison is defined as follows:

Acceptable Segmentation Quality: If the majority of each salient object in the image can be discovered and represented by a non-trivial segment in the segmentation map, such a result is acceptable. Otherwise, it is not.

We compare the segmentation results of our proposed algorithm with the results from REVBH and HC in the new feature space $HSV_rV_gV_b$, respectively, using two performance measures:

Measure 1: The number of images for which one algorithm outperforms the other on the segmentation quality. If there is no clear winner, the comparison result is thus 'comparable.' This measure is for the comparison of relative segmentation quality.

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Measure 2: The number of images for which the segmentation results are acceptable. The results produced by the two algorithms in comparison could be both acceptable while one could outperform the other. This is a measure of absolute segmentation quality.

Since the performance measures are subjective, for fair comparison, a website is created for blind evaluation. Therefore, the user does not know which algorithm generates the result. The final results are summarized from the evaluation results of multiple general users.

First, HCPC and REVBH are performed on the new feature space for the 300 images. Table 1a indicates that even though there are 86 images (28.67%) for which both algorithms produce comparable segmentation results, HCPC has an obvious advantage (181 images vs. 33 images) in segmentation quality with respect to Measure 1. That is because REVBH is a histogram-based algorithm which is not sufficient to differentiate adjacent dense regions. On the contrary, HCPC performs clustering (i.e., *K*-means) in the pixel level when detecting dense regions in each subspace. Therefore, it is more suitable for detecting objects in an image. Measure 2 focuses on absolute segmentation quality (acceptable/not) rather than relative segmentation quality, and thereby some images that yield better segmentation results in HCPC or REVBH according to Measure 1 now become acceptable with both algorithms, resulting in 169 images for which both algorithms produce satisfactory results. However, the performance of HCPC still outperforms that of REVBH with respect to Measure 2.

As indicated in Table 1b, although HCPC clearly outperforms HC in the relative segmentation quality, it does not have an obvious advantage in the absolute segmentation quality over HC in the new feature space, since both algorithms perform clustering in the pixel level. However, HCPC extends HC with the capability to process high dimensional dataset in a more efficient way, because it describes high dimensional dense regions with a set of dense regions in lower dimensional subspaces.

In order to measure the time cost with varying dimensionality of feature space among these three algorithms, we transform a color image with size 192×128 into three datasets with {3, 4, 5} features, respectively. The 3D dataset includes *H*, *S*, and *V* values of each pixel. To construct the 4D dataset, we select *H*, *S*, and another two features of $\{V_r, V_g, V_b\}$, which have the most non-zeros values among the three *V* channels. The 5D dataset is generated in the $HSV_rV_gV_b$ feature space. As we mentioned before, the time complexity of HC increases exponentially when the dimensionality of the feature space grows. This fact is reflected in Figure 6a, which also indicates that *K*-means clustering cannot effectively handle the problem

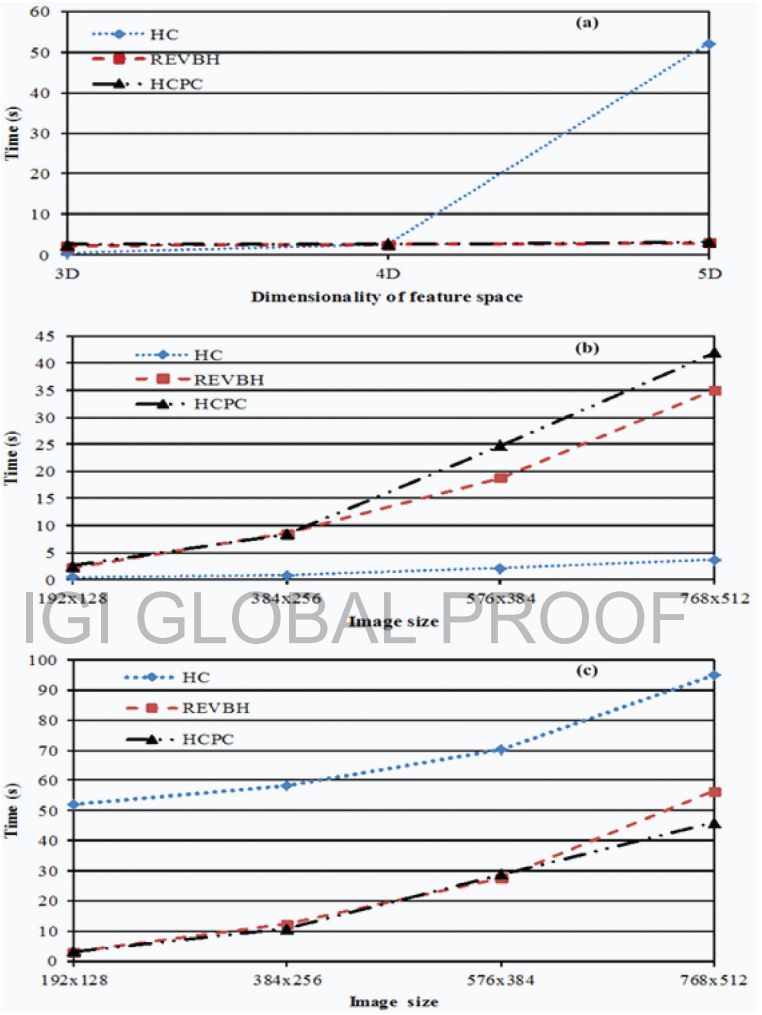
IGI GLOBAL PROOF

Table 1. (a) Experimental comparison between HCPC and REVBH in $HSV_rV_gV_b$ space; (b) Experimental comparison between HCPC and HC in $HSV_rV_gV_b$ space

(a)	Measure 1	**HCPC wins**	**REVBH wins**	**Comparable**	
		181	33	86	
	Measure 2	**HCPC is acceptable but REVBH is not**	**REVBH is acceptable but HCPC is not**	**Both acceptable**	**Both unacceptable**
		88	16	169	27
(b)	Measure 1	**HCPC wins**	**HC wins**	**Comparable**	
		133	62	105	
	Measure 2	**HCPC is acceptable but HC is not**	**HC is acceptable but HCPC is not**	**Both acceptable**	**Both unacceptable**
		61	25	211	3

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 6. (a) Time cost comparison among HC, REVBH, and HCPC with varying dimensionality of feature spaces, (b) Time cost comparison among HC, REVBH, and HCPC in HSV (3D) space with varying image sizes, (c) Time cost comparison among HC, REVBH, and HCPC in $HSV_rV_gV_b$ (5D) space with varying image sizes



IGI GLOBAL PROOF

of *curse of dimensionality* (Hinneburg & Keim, 1999). In the domain of image segmentation, it is not uncommon to use more than 10 features during segmentation, especially when textures and edges (Duvenaud, 2010; Farabet, Couprie, Najman, & LeCun, 2012) are considered, which could easily make HC practically infeasible. On the contrary, the time cost in REVBH and HCPC increases linearly, making them more practical for efficient processing.

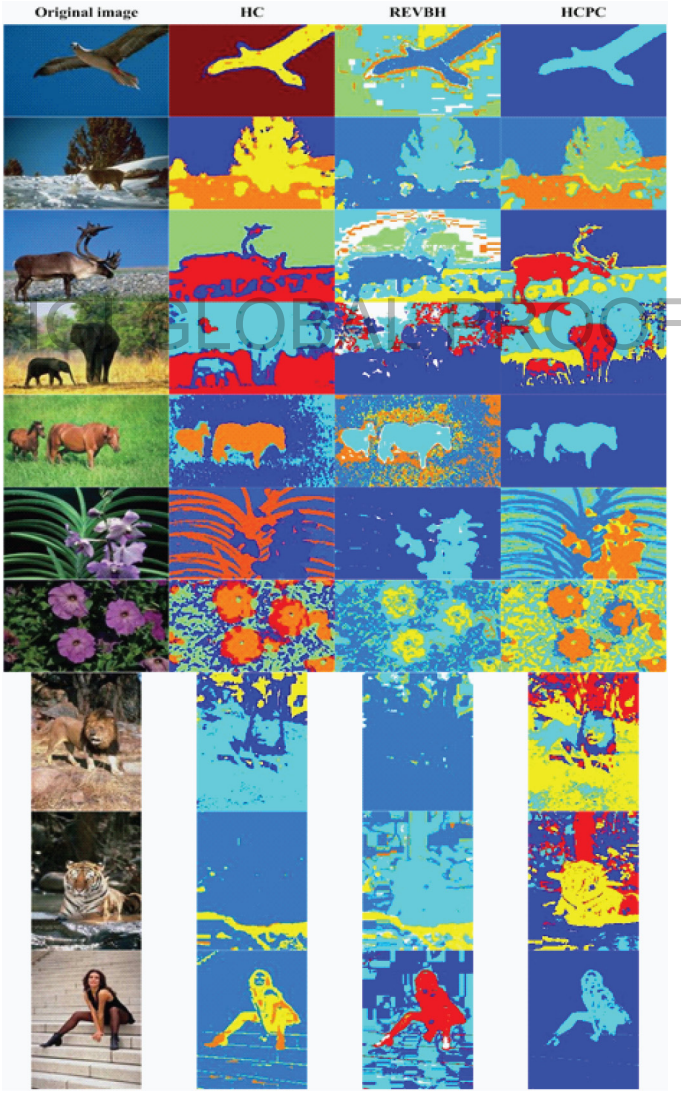
We further test the effect of image size on the time cost when the dimensionality of feature space is fixed (k=3). An image with size 192×128 is enlarged into 4×-, 9×-, and 16×-, respectively. All test images are converted into a 3D dataset in the HSV space. As shown in Figure 6b, when the image size increases, HCPC and REVBH are computationally more costly than that of HC. The signature set in HCPC/REVBH has the same size as that of HSV da-

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

taset in HC (3 entries in a signature vs. 3 channels in HSV), but HCPC has to perform HC in each 2D subspace, and there is also additional cost associated with the signature merging. On the contrary, HC is only applied once on the HSV dataset, making it much more efficient in processing lower dimensional dataset.

However, when more features are extracted from an image, the performance of HC degrades dramatically as shown in Figure 6c. We summarize that HC is suitable for processing large images in low dimensional spaces, while HCPC is good at processing relatively small image in high dimensional feature space. Figure 7 illustrates some segmentation results produced by HC, REVBH, and HCPC, respectively.

Figure 7. Segmentation results generated by HC, REVBH, and HCPC in $HSV_rV_gV_b$ space



Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose a new projective clustering algorithm Hill-Climbing based Projective Clustering (HCPC) for color image segmentation. It combines the framework of a histogram-based projective clustering algorithm EPCH which uses a set of dense regions detected from a lower-dimensional space to approximate dense regions in the high-dimensional space, and the idea of hill-climbing *K*-means algorithm for dense region detection. HCPC not only avoids using histogram to estimate the local data distribution in histogram based algorithm, but also extend HC with the capability for processing high dimensional data. Because HCPC processes images in pixel (data point) level, it has a better segmentation quality compared with histogram-based algorithms. The time complexity of HCPC increases linearly when the dimensionality of feature space becomes higher, and thereby HCPC is more scalable than the original HC algorithm.

There is still a lot of room in this algorithm to improve. First, similarity threshold and weight threshold are simply set as default parameters with constant values. How to determine their values automatically or avoid using such parameters should be considered in the future work. Second, one cluster may contain multiple spatially disconnected segments. Therefore, removing trivial segments is more meaningful than removing trivial clusters. Third, processing images in pixel level is not an efficient way. Some more abstract and meaningful representation of an image, such as superpixel (http://www.cs.sfu.ca/~mori/research/superpixels/) (Kim, Lee, & Lee, 2010; Li, Wu, & Chang, 2012; Mori, 2005; Mori, Ren, Efros & Malik, 2004; Ren & Malik, 2003), could be used in our algorithm for both efficiency and effectiveness purposes.

IGI GLOBAL PROOF

ACKNOWLEDGMENT

This work was supported in part by NSF IOS-1127911.

REFERENCES

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999). Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 61-72).

Aggarwal, C. C., & Yu, P. S. (2000). Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 70-81).

Duvenaud, D. (2010). *Multiscale conditional random fields for machine vision* (Unpublished master's thesis). The University of British Columbia, Vancouver, BC, Canada. Retrieved from http://mlg.eng.cam.ac.uk/duvenaud/papers/multiscale_crf_thesis.pdf

Everitt, B. (2012). *Cluster analysis* (5th ed.). Chichester, UK: John Wiley & Sons.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2012). Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *Proceedings of the 29th International Conference on Machine Learning*.

Freedman, D., & Diaconis, P. (1981). On the histogram as a density estimator: L2 theory. *Probability Theory and Related Fields*, *57*(4), 453–476.

Gao, S., Zhang, C., & Chen, W.-B. (2010). A variable bin width histogram based image clustering algorithm. In *Proceedings of the 4th IEEE International Conference on Semantic Computing* (pp. 166-171).

Gao, S., Zhang, C., & Chen, W.-B. (2011). Identifying image spam authorship with a variable bin width histogram-based projective clustering. In *Proceedings of the IEEE International Conference on Multimedia and Expo* (pp. 1-6).

Han, J. W., & Kamber, M. (2005). *Data mining: Concepts and techniques* (2nd ed., pp. 402–403). San Francisco, CA: Morgan Kaufmann.

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Hinneburg, A., & Keim, D. A. (1999). Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proceedings of the 25th International Conference on Very Large Data Bases* (pp. 506-517).

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, *31*(3), 264–323. doi:10.1145/331499.331504

Kim, T., Lee, K., & Lee, S. (2010). Learning full pairwise affinities for spectral segmentation. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2101-2108).

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*(1), 79–86. doi:10.1214/aoms/1177729694

Li, Z., Wu, X., & Chang, S. (2012). Segmentation using superpixels: A bipartite graph partitioning approach. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (pp. 789-796).

MacKay, D. J. C. (2003). *Information theory, inference and learning algorithms* (pp. 284–292). Cambridge, UK: Cambridge University Press.

Mori, G. (2005). Guiding model search using segmentation. In *Proceedings of the 10th IEEE International Conference on Computer Vision* (pp. 1417-1423).

Mori, G., Ren, X., Efros, A., & Malik, J. (2004). Recovering human body configurations: Combining segmentation and recognition. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In Neural Information Processing Foundation (Eds.), *Advances in neural information processing systems* (pp. 849-856). Cambridge, MA: MIT Press.

Ng, E. K. K., Fu, A., & Wong, R. C. W. (2005). Projective clustering by histograms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 369–383. doi:10.1109/TKDE.2005.47

Ohashi, T., Aghbari, Z., & Makinouchi, A. (2003). Hillclimbing algorithm for efficient color-based image segmentation. In *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications.*

Osborne, J. D., Gao, S., Chen, W.-B., Andea, A., & Zhang, C. (2011). Machine classification of melanoma and nevi from skin lesions. In *Proceedings of the ACM Symposium on Applied Computing* (pp. 100-105).

Ren, X., & Malik, J. (2003). Learning a classification model for segmentation. In *Proceedings of the 9th International Conference on Computer Vision* (pp. 10-17).

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed., pp. 111–114). Upper Saddle River, NJ: Prentice Hall.

Scott, D. W. (1992). *Multivariate density estimation: Theory, practice, and visualization*. New York, NY: John Wiley & Sons. doi:10.1002/9780470316849

Scott, D. W. (2009). Sturges' rule. *WIREs Computational Statistics*, *1*(3), 303–306. doi:10.1002/wics.35

Shapiro, L. G., & Stockman, G. C. (2011). *Computer vision* (pp. 279–325). Upper Saddle River, NJ: Prentice Hall.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905. doi:10.1109/34.868688

IGI GLOBAL PROOF

Song Gao is a PhD student in the Knowledge Discovery and Data Mining Lab of the Department of Computer and Information Sciences at the University of Alabama at Birmingham. His research interests include data mining, machine learning, and multimedia database systems. The projects he participated include A Photographic Method for Human Body Composition Assessment, Using Computer and Information Sciences to Understand Organizational Engagement, Identification of Image Spam Authorship, Machine Classification of Melanoma and Nevi from Skin Lesions, Texture-based Authorship Classification of web images, etc. Gao has an MS in computer science from Chongqing University of Posts and Telecommunications, China.

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Chengcui Zhang is an associate professor of computer and information sciences at the University of Alabama at Birmingham. Her research has been funded by Natural Science Foundation, eBay Inc., IBM, and NIH. She has authored and co-authored more than 100 research papers focusing in the areas of multimedia databases, multimedia data mining, bioinformatics, and geographic information systems. She is currently serving as a program co-chair of the 2012 IEEE International Conference on Information Reuse and Integration (IRI'12). She was a program co-chair for 2011 IEEE International Symposium on Multimedia and the 2010 International Conference on Multimedia Ubiquitous Computing. She has served on more than 90 program committees of major conferences and workshops in multimedia and database systems. Zhang also serves on the editorial boards for several international journals, including International Journal of Multimedia Data Engineering and Management and International Journal of Computer and Informatics.

Wei-Bang Chen is an assistant professor at the Department of Mathematics and Computer Science, Virginia State University. He received his PhD in computer science from the University of Alabama at Birmingham in 2012. He has an MS in genetics from Institute of Genome Sciences, National Yang-Ming University, Taiwan, and an MS in computer science from the University of Alabama at Birmingham. He earned his BS degree from the Department of Biotechnology and Laboratory Science in Medicine, National Yang-Ming University, Taiwan. Chen is also a Microsoft Certified Systems Engineer (MCSE). His research interests include bioinformatics, health informatics, multimedia data mining, image processing, laboratory automation, logistics research, and cyber security.

IGI GLOBAL PROOF

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.