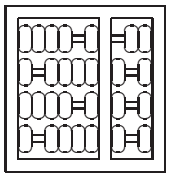

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

A new Hybrid Clustering Approach for Image Retrieval

Anderson Rocha *Jurandy Almeida*
Ricardo Torres *Siome Goldenstein*

Technical Report - IC-07-29 - Relatório Técnico

September - 2007 - Setembro

A new Hybrid Clustering Approach for Image Retrieval*

Anderson Rocha, Jurandy Almeida, Ricardo Torres, and Siome Goldenstein

September 15, 2007

Abstract

In this paper, we present a new *Hybrid Hierarchical Clustering* approach for Image Retrieval. Our method combines features from both divisive and agglomerative clustering paradigms in order to yield good-quality clustering solutions with reduced computational cost. We provide several experiments showing that our technique reduces the number of required comparisons to perform a retrieval without significant loss in effectiveness when compared to flat-based solutions.

1 Introduction

Pictures are worth more than a thousand words. Nowadays, consumers are increasingly creating large collections of digital photographs. There is a growing demand for automatic tools to organize, browse, and search such collections. Content-Based Image Retrieval (CBIR) systems [1] try to address these tasks.

However, CBIR approaches, in general, either are computational costly or present a result that do not satisfy the user. Traditional techniques address only one of these problems (c.f., survey [2]). The challenge in CBIR is to minimize the retrieval process time while keeping the effectiveness as high as possible.

In this paper, we focus on CBIR techniques to improve the efficiency of these systems. In a flat-based retrieval environment, an image needs to be compared to the whole image database to determine the closest matches for a retrieval. Although flat-based retrievals achieves good effectiveness they are computational expensive. There are two possible paradigms to address this problem: (1) data clustering, and (2) indexing structures. The first is the unsupervised classification of observations, data items, or feature vectors into groups (clusters) [3]. The second is data structures designed to improve the lookup performance [4, 5]. Data clustering and indexing structures are suitable for problems of explanatory pattern-analysis, grouping, decision-making, and machine-learning [3].

Data clustering algorithms can be hierarchical or partitional. Hierarchical algorithms find successive clusters using previously established clusters, whereas partitional algorithms determine all clusters at once.

*The authors thank the financial support of Fapesp (Grants 05/52959-3 and 05/58103-3), CNPq (Grants 301278/2004, 311309/2006-2, and 477039/2006-5), and Microsoft EScience Project.

The strategies in hierarchical clustering are divided into two basic paradigms: bottom-up agglomerative or top-down divisive. Agglomerative strategies begin with each element as a separate cluster and merge them into successively larger clusters. On the other hand, divisive strategies start with one cluster and divide it into two new clusters. The process is recursively repeated for each cluster. Zhao and Karypis [6] have suggested that divisive hierarchical clustering approaches are more appropriate for clustering large document datasets than agglomerative techniques.

We present a new *Hybrid **D**ivisive-**A**gglomerative **H**ierarchical **Cluster**ing* (DAH-Cluster) approach suitable it for Image Retrieval. DAH-Cluster combines features from both divisive and agglomerative clustering paradigms in order to yield good-quality clustering solutions. We provide several experiments showing that our technique not only greatly reduces the number of required comparisons to perform a retrieval but also do not present significant loss in effectiveness when compared to flat-based solutions.

2 Related work

A lot of researchers have proposed approaches for document classification and clustering [6–9]. Ferragina and Gulli [10] have presented *SnakeT*, a hierarchical clustering approach that organizes the search results from different web-search engines on-the-fly. They have used the resulting hierarchy to complement the view of the flat-ranked list of results returned by common available search engines. The clustering stage can be applied in different contexts. For instance, some approaches perform the clustering in the metric space itself [11]. On the other hand, clustering stages are used to find meaningful groups in micro-array descriptors and to reduce the impact of high-dimensionality data in such applications [12].

Some researchers have presented clustering techniques for Content-Based Image Retrieval (CBIR). Cooper et al. [13] have presented a clustering approach based on the computation of the similarity among photos' time-stamps. The authors have showed that photographs from the same event and taken in relatively close proximity in time can lead to 33% decrease in the speed of image retrieval. Shyu et al. [14] have introduced a unified framework to facilitate conceptual database clustering and CBIR using Markov Model Mediators (MMMs).

Heller and Ghahramani [15] have designed an algorithm for agglomerative hierarchical clustering based on marginal likelihoods of a probabilistic model. However, this approach is difficult to apply for CBIR given that it is hard to find a probabilistic model suitable to describe images.

Antani et al. [16] have developed clustering techniques for hybrid text/image query-retrieval for medical images. Malik et al. [17] have proposed a technique to overcome problems of region growing algorithms such as seed point selection and processing order. In their approach, pixel-based and neighboring pixels are merged in order to create representative clusters. In turn, Stehling et al. [18] have proposed an adaptative agglomerative clustering algorithm to segment images into high-similarity regions. Their approach is based on pixel-wise connected components and color similarity.

Bhatia [19] has introduced a hierarchical clustering technique for image databases. In

this approach, the stored models are represented hierarchically into the database instead of using a flat structure. However, this technique presents the undesirable requirement of changing the way the images are physically stored thus breaking up the logical and physical data independence in the database.

Kinoshenko et al. [20] have proposed a technique to partition the image into disjoint subsets. In their approach, the system splits each query into representative subclasses and finds the most similar stored subclasses to each part of the query. However, the classes of images need to represent a structural hierarchy, for instance the relationship present in images of a car and its parts.

3 Image descriptors

In this section, we present four CBIR low-level descriptors which were implemented and used as reference in our experiments. Section 4 shows how we can reduce the number of required operations of these descriptors and how to improve their effectiveness.

3.1 Global Color Histogram (GCH)

The simplest approach to encode the information present in an image is the Global Color Histogram (GCH) [21]. A GCH is a set of ordered values, one for each distinct color, representing the probability of a pixel being of that color. Uniform quantization and normalization are used to reduce the number of distinct colors and to avoid scaling bias [21]. The L_1 (City-block) or L_2 (Euclidean) are the most used metrics for histogram comparison.

Histograms are effective for retrieval if there is uniqueness in the color pattern present in the images we want to compare. However,

3.2 Color Coherence Vectors (CCVs)

Zabih et al. [22] have presented an approach to compare images based on color coherence vectors. They define color's coherence as the degree to which pixels of that color are members of large similarly-colored regions. They refer to these significant regions as coherent regions. Coherent pixels are part of some sizable contiguous region, while incoherent pixels are not.

In order to compute the CCVs, first the method blurs and discretizes the image's color-space to eliminate small variations between neighboring pixels. Next, it finds the connected components in the image aiming to classify the pixels within a given color bucket as either coherent or incoherent.

3.3 Border/Interior Classification (BIC)

Stehling et al. [23] have presented the border/interior pixel classification (BIC), a compact approach to describe images. BIC relies on the RGB color-space uniformly quantized in $4 \times 4 \times 4 = 64$ colors. After the quantization, the image pixels are classified as *border* or

interior. A pixel is classified as *interior* if its 4-neighbors (top, bottom, left, and right) have the same quantized color. Otherwise, it is classified as *border*.

After the image pixels are classified, two color histograms are computed: one for border pixels and another for interior pixels. The two histograms are stored as single histogram with 128 bins. BIC compares the histograms using the *dLog* distance function [23]

$$dLog(q,d) = \sum_{i=0}^{i<M} \|f(q[i]) - f(d[i])\| \tag{1}$$

$$f(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } 0 < x < 1 \\ \lceil \log_2 x \rceil + 1, & \text{otherwise} \end{cases} \tag{2}$$

where q and d are two histograms with M bins each. The value $q[i]$ represents the ith bin of histogram q, and $d[i]$ represents the ith bin of histogram d.

3.4 Unser's descriptors

Unser [24] has presented a set of statistical texture image descriptors based on histograms of sums H_s and differences H_d. The method computes these histograms by summing/subtracting the pixel value on the image I in position (i,j) with the pixel's neighbors values defined in a neighborhood with radius (δ_i, δ_j)

$$H_s[I(i,j) + I(i+\delta_i, j+\delta_j)] \mathrel{+}= 1 \tag{3}$$

$$H_d[I(i,j) - I(i+\delta_i, j+\delta_j)] \mathrel{+}= 1. \tag{4}$$

In order to build the final feature vector, the method computes several statistical descriptors (e.g., mean (μ), contrast (C_n), homogeneity (H_g), energy (E_n), variance (σ^2), correlation (C_r), and entropy (H_n)) over the histograms.

4 DAH-Cluster

Common belief points out that agglomerative clustering in general leads to better solutions for clustering than partitional algorithms [6]. Contrary to this belief, in this section, we present a new *Hybrid Divisive-Agglomerative Hierarchical Clustering* (DAH-Cluster) approach for Image Retrieval.

Our approach is hybrid; it relies on the combination of features from both divisive and agglomerative clustering paradigms. This combination yields good-quality clustering solutions with fewer computational operations. Now, we present details about our technique as well as implementation considerations to make it suitable for image retrieval tasks.

4.1 How to perform a query

DAH-Cluster creates an offline hierarchical structure to represent groups of images as we show in Figure 2. The method creates overlapping groups of similar images. At each level

of the hierarchy, DAH-Cluster refines the groups formation in order to save computational comparisons and keep the effectiveness as high as possible when performing a retrieval.

Using DAH-Cluster's structure, we perform a retrieval comparing the cluster representatives to find the closest cluster to the query at each level. In the lowest level, we sort the cluster representatives. Finally, we sort the elements in each cluster given by the ordering in the cluster representatives until we complete the minimum required number of elements in the retrieval (top m elements, for instance).

To illustrate how the method works, we create a toy example with 24 images of 8 classes (3 images per class) of Corel Photo Gallery. Figure 1 depicts a query image and its top-3 results for BIC descriptor. The direct metric evaluation (flat-based) can lead to undesirable results as we see in the R_3 result which clearly is not of the same semantical class of the query image Q.

Sometimes, DAH-Cluster can improve the overall retrieval effectiveness as we show in the top-3 results in Figure 1(b). That is because at each level of the hierarchy, wrong elements are attracted to new clusters which tends to represent their real classes. Hence when we perform a retrieval, this process tend to eliminate some outliers, i.e, representative elements far from the class of the query image.

In Figure 2, we show DAH-Cluster results for the selected 24 images of this toy example. For instance, the wrong result R_3 for the query Q in Figure 1 is inserted in the cluster c_{51} which correctly puts it in a new cluster c_{511} whose representative is far from the query image Q.

(a) (b)

Figure 1: Flat (left) and DAH-Cluster (right) toy example's top-3 results.

4.2 Method's description

In this section, we present the formalization of our method. Let c represent a cluster, c_{rep} be a representative element of the cluster c, $c_{elements}$ be a set of elements in the cluster c, and c_{child} be a pointer to the next level c in a hierarchy of clusters. In addition, let C be a set of clusters, k be the number of clusters in all clustering tasks, $f \in [0, 1)$ be a factor of re-clustering, E be a set of elements under analysis, and D a metric measuring the dissimilarities among elements in E.

In Algorithm 1, we present DAH-Cluster algorithm. It consists of three steps: (1) we perform a clustering stage over the initial set of elements E stored in the database (line 2); (2) For each resulting cluster $c \in C$ in (1), we build a new set E^* with elements $c_{elements}$ of the $\lfloor f \times k \rfloor$ closest clusters to $c.rep$ (lines 4-8); (3) If the number of elements for each resulting E^* is greater than the number of clusters k, we create a new level c_{child} in the

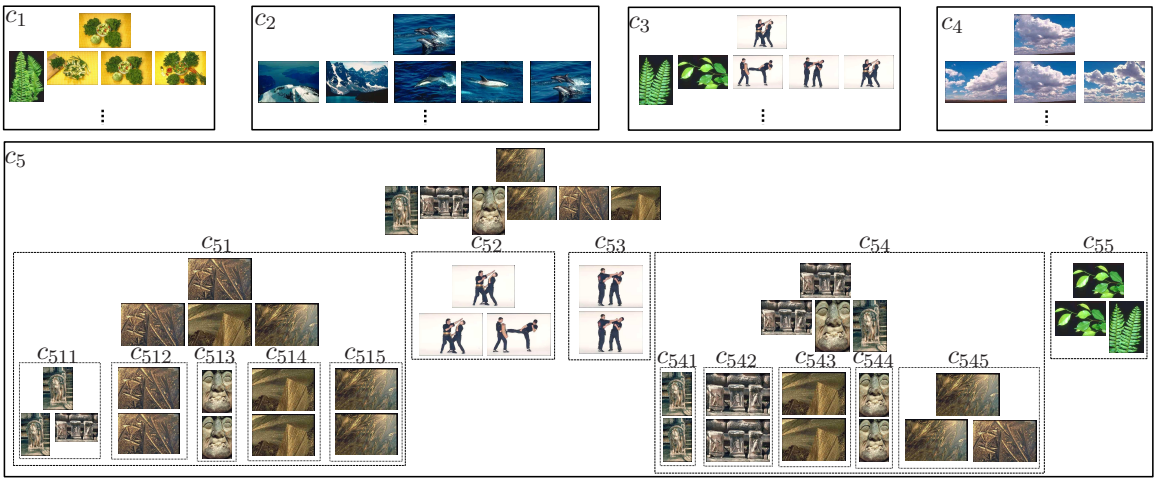

Figure 2: DAH-Cluster toy example for 24 images and 8 classes.

clustering hierarchy of c and we iterate the steps (1-3) for the smaller set of elements E^* (lines 9-11).

The method represents an agglomerative clustering paradigm in the sense that it starts with each element as a representative cluster and it finds k clusters. It represents a divisive clustering paradigm in the sense that it starts with a set of elements E and it iteratively partition E into subsets E^*. This partitioning is a factor of re-clustering that intends to put together some similar elements that otherwise would be separated.

The procedure $\textsc{Cluster}(k, E, D)$ can be any agglomerative or partitional clustering method such as K-means or K-medoids [25]. We recommend the choice of K-medoids when it is desired an independent metric space. K-medoids only needs a dissimilarity matrix among elements while K-means requires an Euclidean-space dissimilarity metric.

K-means and K-medoids are two of the most popular partitional clustering methods [25]. K-means is intended for situations in which all variables are of the quantitative type, and the dissimilarities can be measured in the Euclidean space. However, Euclidean-distance based procedures lack robustness against outliers. In turn, K-medoids is a generalization of K-means in the sense that other metrics rather than Euclidean distance can be used to measure the dissimilarities among elements [2]. In spite of their limitations, both methods are largely used in the information retrieval literature [25].

Figure 3 shows an illustration of DAH-Cluster. At the beginning, we have E elements to be clustered. After the first iteration, we have the level C_1 of the hierarchy with k clusters $c_1 \dots c_k$. For each cluster c_i in C_1, we find the $\lfloor f \times k \rfloor$ closest clusters to c_i, create a new set of elements E_{c_i}, and perform the clustering again for each E_{c_i}, generating the next level in the hierarchy of clusters with nodes $c_{11} \dots c_{1k} \dots c_{k1} \dots c_{kk}$. We repeat this process while the number of elements in the cluster c_i is greater than the number of cluster k, i.e., while $|E_{c_i}| > k$.

Algorithm 1 DAH-Cluster

Require: The number of clusters k, the re-clustering factor $f \in [0, 1)$, the set of elements E, and a metric D;

procedure DAH-CLUSTER(k,f,E,D)
  $C \leftarrow$ CLUSTER(k, E, D)
  for each $c \in C$ **do**
    $C^* \leftarrow \lfloor f \times k \rfloor$ closest clusters of $c \in C$
    $E^* \leftarrow \{\}$
    for each $c^* \in C^*$ **do**
      $E^* \leftarrow E^* \cup c^*_{elements}$
    end for each
    if $|E^*| > k$ **then** ▷ $|\cdot|$ is the size of $\{\cdot\}$
      $c_{child} \leftarrow$ DAH-CLUSTER(k, f, E^*, D)
    end if
  end for each

end procedure

4.3 Convergence

Theorem 1. *DAH-Cluster method always converges in the number of clusters (width) and in the number of levels (depth).*

Proof. We have two possibilities: (1) the width convergence and (2) the depth convergence. The first is direct controlled by the number of clusters k. We use a clustering algorithm such as K-medoids and K-means which relies on the number of clusters k and always converges to the solution either by stability or by a fixed number of iterations. A proof can be found in [25]. The second is controlled by the factor of re-clustering f. Analyzing the hierarchy, we have

$$|c_i| < |E| \tag{5}$$

i.e., the size of each generated cluster is always less than $|E|$. Furthermore, we see that

$$\sum_{i=1}^{k} |c_i| = |E|. \tag{6}$$

The sum of the elements of all clusters in a tree branch is always equal to $|E|$. Therefore, we generate the next level of the hierarchy for the cluster c_i selecting $\lfloor f \times k \rfloor$ closest clusters to the cluster c_i. The number of selected clusters for a branch in the next level is always less than k given that $f \in [0, 1)$

$$\lfloor f \times k \rfloor < k. \tag{7}$$

Thus

$$\sum_{i=1}^{\lfloor f \times k \rfloor} |c_i| < |E| \tag{8}$$

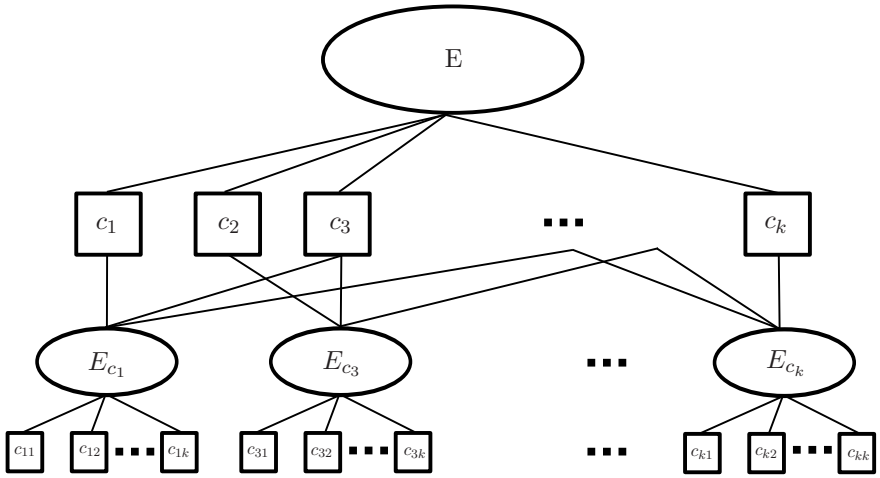

Figure 3: A representation of DAH-Cluster.

i.e., the sum of the elements for all selected clusters in the re-clustering stage for branch c_i is always less than the total number of elements to be clustered in this level E.

However, the number of elements to be clustered in the next level branch is given by

$$|E^*| = \sum_{i=1}^{\lfloor f \times k \rfloor} |c_i|. \tag{9}$$

which proves that $|E^*| < |E|$. □

5 Experiments

In this section, we present results for an application of DAH-Cluster for image retrieval tasks. Here we use the set of image descriptors described in Section 3. We compare our technique (DAH-Cluster) with direct or flat retrieval (with no clustering stage), hierarchical divisive (DHC), and partitional clustering (PC). We show that our method improves the efficiency of retrieval tasks.

The instantiation of DAH-Cluster for CBIR consists of two steps: (1) to build an offline hierarchical structure that better represents semantical relationships among images; (2) to use the structure in (1) to perform online retrievals.

5.1 Methodology

In this work, we have used the *query-by-example* (QBE) paradigm [26]. In QBE, we give an image as a visual example to the system and we query for images that are similar to the

given example. Clearly, the effectiveness of these systems is dependent on the properties of the example image.

In order to assess the system effectiveness, we divide an image database into two sets: training and testing. We perform 5-fold cross-validation in the evaluation process. We repeat this process 10 times and provide average results. We have used the simple and well-known algorithm of K-medoids in the clustering tasks [25].

In this paper, we have used two image databases described in the literature. The first database is a selection of the 1,624 images from Corel Photo Gallery and is the same as the reported in [23]. This database is highly heterogeneous and contains with different 50 categories. Figure 4 shows some examples of some selected categories.

Figure 4: Corel Photo Gallery database.

The second database is freely available[1]. We select 3,462 natural images from FreeFoto divided into nine classes. Figure 5 shows some examples of each category. *Sky and Clouds* category represents sunny and clear days. *Cummulonimbus Clouds* comprises images associated with heavy precipitation and thunderstorms. The other categories are self explanatory.

We use the *Precision* × *Recall* [26] metric to assess the retrieval effectiveness. *Precision* is the ratio of the number of relevant images retrieved to the total number of irrelevant and relevant images retrieved. *Recall* is the ratio of the number of relevant images retrieved to

[1] `http://www.freefoto.com/preview.jsp?id=15-19-1/`

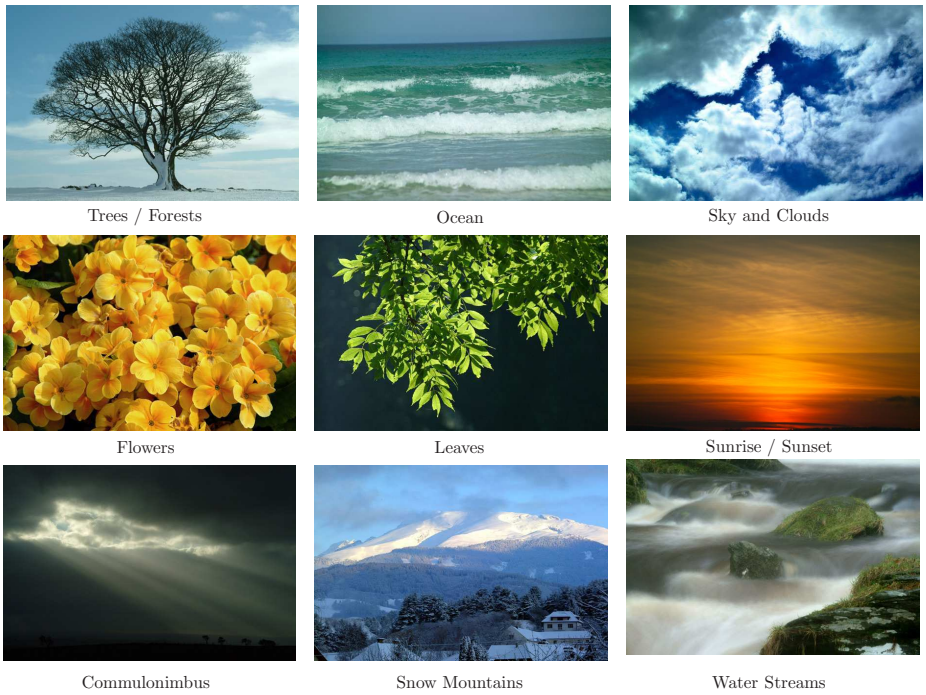

Figure 5: FreePhoto database.

the total number of relevant images in the database.

In the experiments, we have calculated the *Precision* × *Recall* after the top-30 images are retrieved. This represents the number of relevant images in the top-30 resulting images for each query. This value is an estimate of the number of retrieved images an user would accept to inspect in order to determine their relevance to her needs and it was first reported in [23].

5.2 Overall results

In this section, we present results for our method and provide comparison with state-of-the-art approaches. *PC* stands for *Partitional Clustering* (simple K-medoids), *HC* stands for *Hierarchical Clustering* and DAHC-f is *Divisive-Agglomerative Hierarchical Clustering* with factor of re-clustering f. In the experiments, we use values of k that are multiples of 5. We have chosen f values in order to yield integer values when calculating $f \times k$. Hence, we report results using $f \in \{0.05, 0.1, 0.15, 0.2\}$. High values of f leads to high overload in the offline creation of the hierarchical structure. We provide results for BIC, GCH, CCV, and Unser image descriptors. The dashed-lines is the direct flat retrieval using one of the descriptors.

Figures 6 and 7 show the efficiency and effectiveness results for Corel Photo Gallery and FreeFoto databases, respectively. Looking at the results, we see that DAH-Cluster greatly reduces the number of required operations (improves the efficiency) regardless the database

and the image descriptors. There is a trade-off between f and k in order to produce good results. The greater k, the better the precision. However, the greater k, the greater the number of required operations. In turn, the greater f, the better the re-clustering stages. Nevertheless, the greater f the greater the offline overload needed to create the hierarchical structure. We have found, experimentally, that $f = 0.2$ is a good trade-off for offline efficiency and online effectiveness. In each stage, $50 < k < 100$ is a good choice for k.

6 Conclusions

In this paper, we presented a new *Divisive - Agglomerative Hierarchical Clustering* approach for Image Retrieval (DAH-Cluster). It combines features from both divisive and agglomerative paradigms.

We provided several experiments showing that our technique reduces the number of required comparisons to perform a retrieval and still provides good effectiveness when compared to direct (flat) retrieval. The effectiveness small losses are acceptable in practical situations given the several orders of magnitude reduction in the number of required operations in each retrieval task.

DAH-Cluster relies on the choice of two factors: the number of of clusters k and the re-clustering factor f. We showed that high values of k improves effectiveness but leads to more required operations in order to perform a retrieval. In turn, we showed that there is a trade-off between f and k in order to produce good results. In general, if we increase k, we improve the effectiveness. However, high values of k leads to more required operations in order to perform a retrieval. In turn, high values of f improve the re-clustering stage and the online efficiency. However, the greater f the greater the overload in the offline creation of the hierarchical structure of the database. We have found, experimentally, that $f = 0.2$ is a good trade-off for low offline overload and online efficiency/effectiveness. In each stage, $50 < k < 100$ is a good choice for k. Finally, we also provided a formal proof of DAH-Cluster's convergence.

DAH-Cluster only requires the set of elements to be analyzed, and a metric measuring the dissimilarities among them. In this context, our future work include the application of our method for text retrieval and indexing using the state-of-the-art text descriptors in the literature. Furthermore, we intend to validate the method on a web-scale CBIR environment such as one containing several thousands of images.

Acknowledgments

The authors thank the financial support of Fapesp (Grants 05/52959-3 and 05/58103-3), CNPq (Grants 301278/2004, 311309/2006-2, and 477039/2006-5), and Microsoft EScience Project.

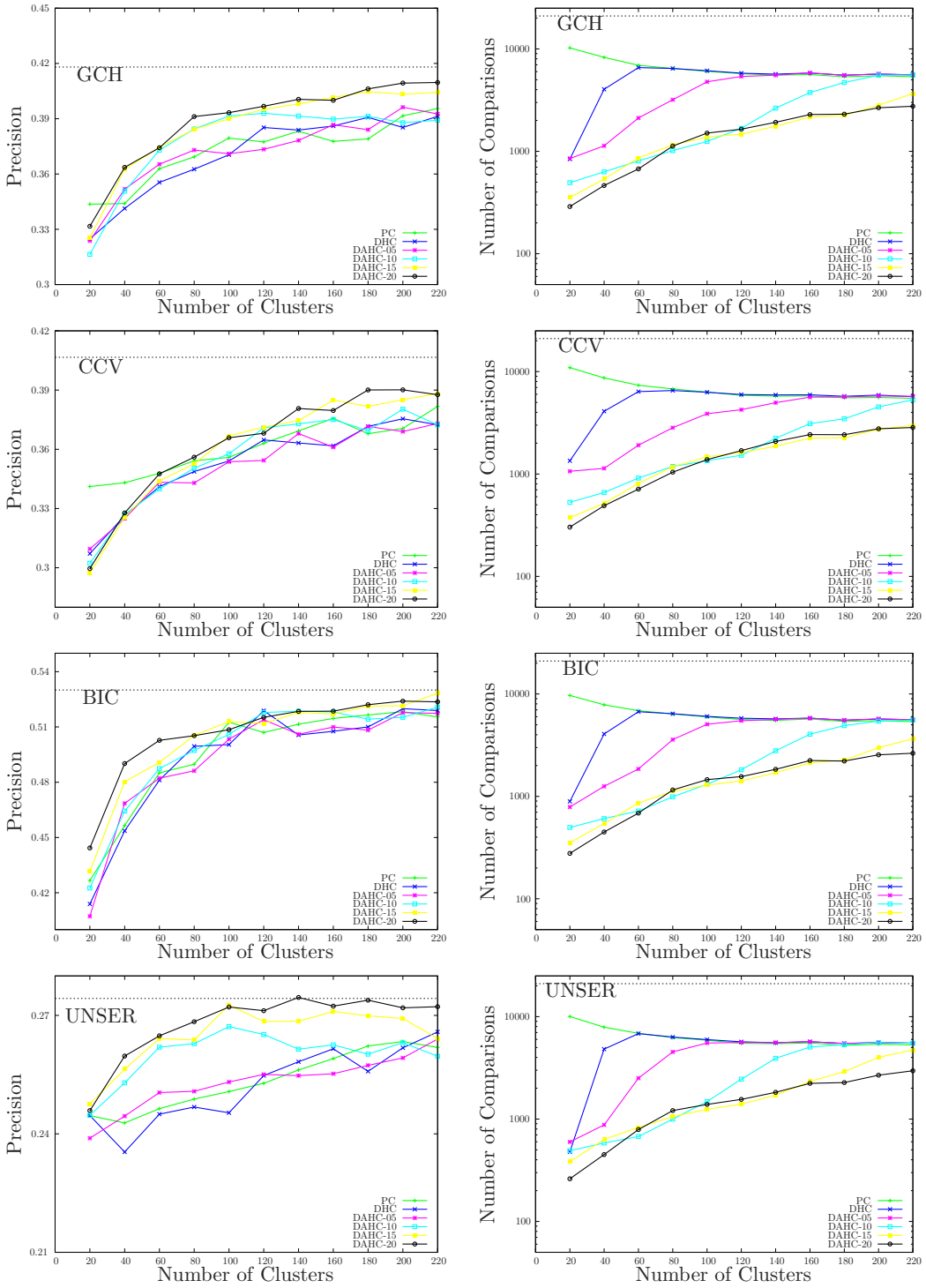

Figure 6: Effectiveness (left) and Efficiency (right) for Corel Photo Gallery. The results for GCH, CCV, BIC, and Unser image descriptors, respectively, are showed from top to bottom.

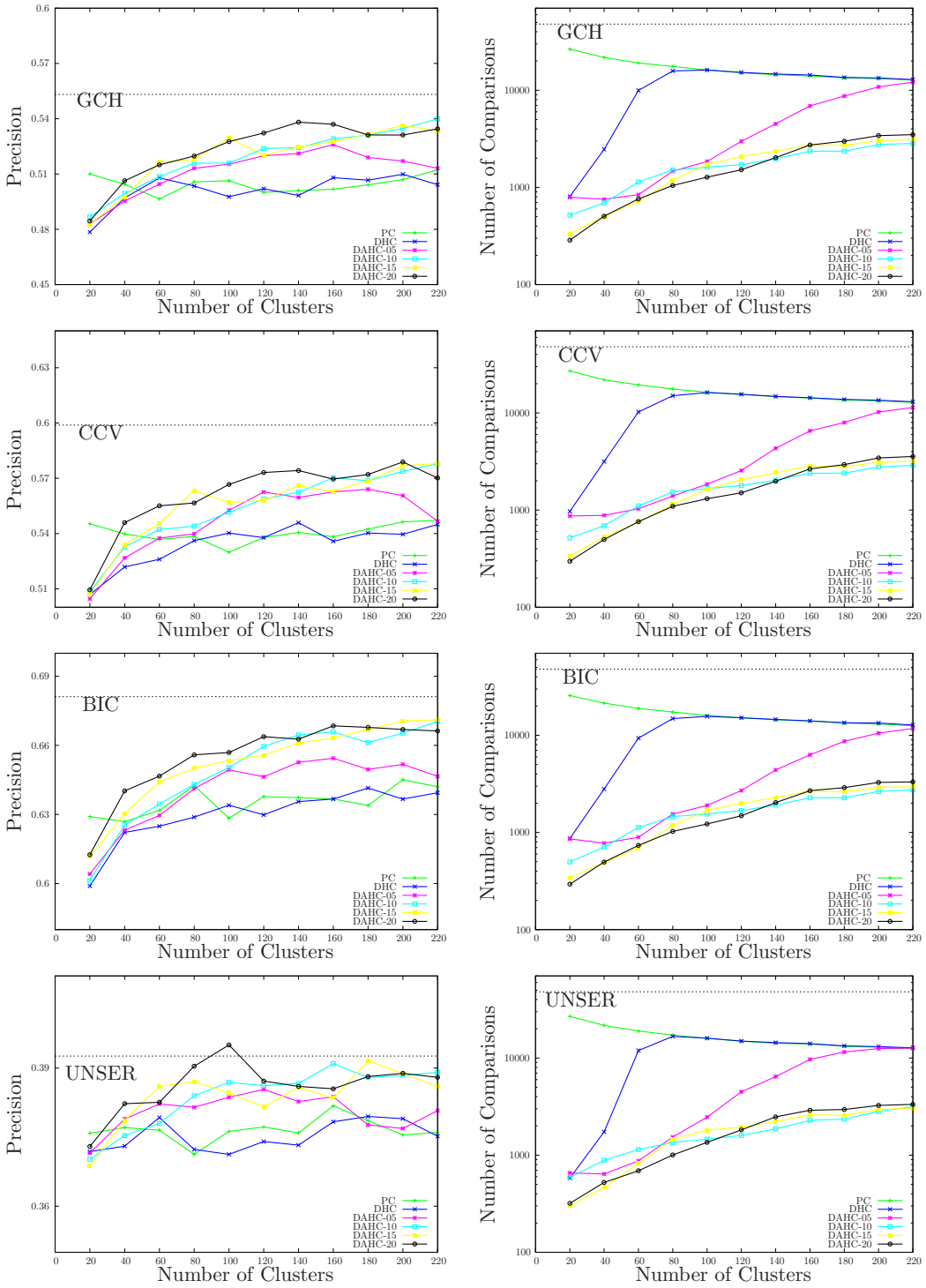

Figure 7: Effectiveness (left) and Efficiency (right) for FreeFoto. The results for GCH, CCV, BIC, and Unser image descriptors, respectively, are showed from top to bottom.

References

[1] R. C. Veltkamp and M. Tanase, "Content-Based Image Retrieval Systems: A Survey," Department of Computing Science, Utrecht University, Tech. Rep. UU-CS-2000-34, 2000.

[2] P. Berkhin, "Survey of clustering data mining techniques," *Accrue Software*, vol. 10, pp. 92–1460, 2002.

[3] A. K. Jain, M. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[4] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern Recognition*, vol. 35, pp. 945–965, 2002.

[5] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic Annotation, Indexing, and Retrieval," *Journal of Web Semantics*, vol. 2, no. 1, pp. 49–79, 2004.

[6] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical Clustering Algorithms for Document Datasets," *DMKD*, vol. 10, no. 2, pp. 141–168, 2005.

[7] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, "Hierarchical clustering of WWW image search results using visual, textual and link information," in *12th ACM MM*, 2004, pp. 952–959.

[8] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, "Incremental hierarchical clustering of text documents," in *16th CIKM*, 2006, pp. 357–366.

[9] R. Campos and G. Dias, "Automatic Hierarchical Clustering of Web Pages," in *28$_{th}$ ACM SIGIR*, 2005, pp. 83–85.

[10] P. Ferragina and A. Gulli, "A personalized search engine based on web-snippet hierarchical clustering," in *WWW'05*, 2005, pp. 801–810.

[11] R. Baeza-Yates, B. Bustos, E. Chávez, N. Herrera, and G. Navarro, *Clustering and Information Retrieval*. Kluwer Academic Publishers, 2003, pp. 1–34.

[12] J. Seo and B. Shneiderman, "Interactive Exploration of Multidimensional Microarray Data: Scatterplot Ordering, Gene Ontology Browser, and Profile Search," Ph.D. dissertation, Univ. of Maryland, College Park, 2003.

[13] A. G. Matthew Cooper, Jonathan Foote and L. Wilcox, "Temporal event clustering for digital photo collections," *ACM TOMCCAP*, vol. 1, no. 3, pp. 269–288, 2005.

[14] M.-L. Shyu, S.-C. Chen, M. Chen, and C. Zhang, "A unified framework for image database clustering and content-based retrieval," in *ACM Intl. Workshop on Multimedia Databases*, 2004, pp. 19–27.

[15] K. Heller and Z. Ghahramani, "Bayesian Hierarchical Clustering," in *22nd ICML*, 2005, pp. 297–304.

[16] S. Antani, R. Long, and G. Thoma, "Content-Based Image Retrieval for Large Biomedical Image Archives," in *11th WCMI*, 2004, pp. 829–833.

[17] C. Thies, A. Malik, D. Keysers, M. Kohnen, B. Fischer, and T. M. Lehmann, "Hierarchical feature clustering for content-based retrieval in medical image databases," in *SPIE*, 2003, pp. 598–608.

[18] R. Stehling, M. Nascimento, and A. Falcão, "An Adaptive and Efficient Clustering-Based Approach for CBIR in Image Databases," in *ACM Intl. Symposium on Database Engineering & Application*, 2001, pp. 356–365.

[19] S. Bhatia, "Hierarchical Clustering for Image Databases," in *IEEE EIT*, 2005, pp. 6–12.

[20] D. Kinoshenko, V. Mashtalir, E. Yegorova, and V. Vinarsky, "Hierarchical Partitions for Content Image Retrieval from Large-Scale Database," in *4th MLDM*, 2005, pp. 445–456.

[21] M. J. Swain and B. H. Ballard, "Color indexing," *IJCV*, vol. 7, no. 1, pp. 11–32, 1991.

[22] Greg Pass and Ramin Zabih and Justin Miller, "Comparing images using color coherence vectors," in *ACM Multimedia*, 1996, pp. 65–73.

[23] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *11th CIKM*, 2002, pp. 102–109.

[24] M. Unser, "Sum and difference histograms for texture classification," *TPAMI*, vol. 8, no. 1, pp. 118–125, 1986.

[25] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2001.

[26] R. S. Torres and A. X. Falcão, "Content-based image retrieval: Theory and applications," *RITA*, vol. 13, no. 2, pp. 161–185, 2006.