Correspondence Analysis and Hierarchical Indexing For Content-Based Image Retrieval

MILANESE, Ruggero, SQUIRE, David, PUN, Thierry

Abstract

This paper describes a two-stage statistical approach supporting content-based search in image databases. The first stage performs correspondence analysis, a factor analysis method transforming image attributes into a reduced-size, uncorrelated factor space. The second stage performs ascendant heirarchical classification, an iterative clustering method which constructs a heirarchical index structure for the images of the database. Experimental results supporting the applicability of both techniques to data sets of heterogeneous images are reported.

IEEE International Conference on Image Processing
Lausanne, Switzerland, 16-19 September 1996

CORRESPONDENCE ANALYSIS AND HIERARCHICAL INDEXING FOR CONTENT-BASED IMAGE RETRIEVAL

Ruggero Milanese, David Squire, Thierry Pun

Computer Science Department, University of Geneva,
24, rue Général-Dufour, 1211 Genève 4, Switzerland
E-mail: milanese@cui.unige.ch

ABSTRACT

This paper describes a two-stage statistical approach supporting content-based search in image databases. The first stage performs correspondence analysis, a factor analysis method transforming image attributes into a reduced-size, uncorrelated factor space. The second stage performs ascendant hierarchical classification, an iterative clustering method which constructs a hierarchical index structure for the images of the database. Experimental results supporting the applicability of both techniques to data sets of heterogeneous images are reported.

1. CONTENT-BASED IMAGE RETRIEVAL

The recent development of multimedia information systems has brought the need to manage large image databases, using efficient access techniques for archival and retrieval. Besides the use of catalog inventories based on textual descriptions, some systems have introduced the possibility to query the database through example images, or by specifying the desired image in terms of dominant colors [2] [3], texture [2] [8], curvilinear shape of objects [2] [7], and overall appearance of the whole image [8].

However, the practical application of the above approaches is often affected by several drawbacks. First, the choice of the image attributes appears arbitrary, and is often dependent on the specific application domain of the database. To overcome the task-dependency of such a choice, one may use a large set of general-purpose attributes, for instance color histograms, co-occurrency matrices, etc. However, this often results in very high-dimensional spaces, which are often highly correlated. Furthermore, the common use of the Euclidean distance for computing a similarity measure in such feature spaces appears unjustified.

Another weak point of most existing approaches is the inefficiency of their search operations at retrieval time. Given the large number of images and the high dimensionality of classical feature spaces, it is important to avoid exhaustive, linear-time direct comparisons. In other words, it is necessary to automatically map feature vectors into *indexes* which provide a more rapidly access to the relevant images in the database [5] [9]. Examples are hash tables, providing nearly constant access time [4], or hierarchical index structures, such as trees.

In this paper, we describe an approach for computing low-dimensional decorrelated indexes for an image database, and for organizing such indexes in a hierarchical tree structure allowing logarithmic-time binary search. In addition, these methods indicate the subset of pertinent image features and attributes that may be retained for indexing. Both techniques draw from methods developed for *exploratory statistics* [6], which offer well-founded, formal approaches for "exploring and explaining" the underlying structure of a data set.

2. CORRESPONDENCE ANALYSIS

Given a set of images $\mathbf{I}_1, ..., \mathbf{I}_N$ composing the database, a corresponding set of M-dimensional feature vectors $\mathbf{g}_i = [g_{i,1}, ..., g_{i,M}]$, $i \in [1 \ldots N]$ is first computed, and stored in a data table $\mathbf{G} = [g_{ij}]$ of dimension $N \times M$. The index creation stage consists of mapping these vectors onto an appropriate index space. We propose to derive such a mapping by applying a factor analysis method, called *correspondence analysis* (CA), to the data table $\mathbf{G}$. In a similar way to other factor analysis methods (e.g. principal component analysis, Karhunen-Loeve decomposition), a new orthogonal set of axes (the factor axes) is found, so as to maximize the sum of norms of the projected data onto the new axes. These axes can be ordered, according to a decreasing amount of data variance they account for. In this way, a lower-dimension decorrelated factor subspace may be identified, accounting for most of the variance in the data. In computational terms, this means that all subsequent data representation and comparison operations can be performed on a reduced dimensional space of size $L \leq \min\{M, N\}$.

The distinctive feature of CA with respect to other classical factor analysis methods lies in a particular normalization applied to the data table $\mathbf{G}$ before the transformation. This normalization yields an *observation matrix* $\mathbf{P}$. It can be shown that the Euclidean metrics defined over points in the transformed space $\mathbf{G}'$ is equivalent to the χ^2 distance on the corresponding points in the original space $\mathbf{G}$. Below is a detailed description of the normalization and transformation operations. Let us first denote by g the scalar:

$$g = \sum_{i, j} g_{ij} \quad .$$

Let us also define two diagonal weight matrices $\mathbf{D}_\lambda$ and $\mathbf{D}_\gamma$, as follows:

$$[\mathbf{D}_\lambda]_{ii} = \left(\sum_{j=1}^{M} \mathbf{G}_{ij}\right)^{-1/2}, \quad [\mathbf{D}_\gamma]_{jj} = \left(\sum_{i=1}^{N} \mathbf{G}_{ij}\right)^{-1/2}$$

The data table $\mathbf{G}$ is first normalized into the observation matrix $\mathbf{P} = \mathbf{D}_\lambda \mathbf{G} \mathbf{D}_\gamma$. The next step is the projection of $\mathbf{G}$ into the factor space, given by the transformation $\mathbf{G}' = g\mathbf{D}_\lambda \mathbf{P}\mathbf{E}$, where the matrix $\mathbf{E} = [\mathbf{e}_1, ..., \mathbf{e}_M]$ contains the eigenvectors of the covariance matrix $\mathbf{C} = \mathbf{P}^T\mathbf{P}$.

The "normalization" of $\mathbf{G}$ into $\mathbf{P}$ by means of the two matrices $\mathbf{D}_\lambda$ and $\mathbf{D}_\gamma$ allows both images and features (rows and columns of $\mathbf{G}$) to be projected onto a *common* factorial space, where they play a similar role and can be analyzed simultaneously. For instance, it is possible to discover which particular attribute is closest to a cluster of images, hence which are the important parameters characterizing such images. Conversely, it is possible to identify the images most typical of a given attribute. Correspondence analysis also provides tools for analyzing the statistical significance of the projected vectors in the factorial space. For instance, it is possible to select the attributes whose absolute contribution to each factorial axis is the largest. Further details on such analysis tools can be found in [6] [10].

3. HIERARCHICAL INDEXING

Given an image database, and the associated feature vectors $\mathbf{g}_1, ..., \mathbf{g}_N$, the above procedure describes how to compute their coordinates $\mathbf{x}_1, ..., \mathbf{x}_N$ in a reduced L-dimensional factorial space. In order to perform efficient content-based retrieval in the database, it is necessary to construct a hierarchical index representation which, given a query image described by a vector $\mathbf{x}$, directly points to classes of similar images. This can be done by clustering the transformed feature vectors using some metric. Below we describe an iterative clustering algorithm called *ascendant hierarchical classification* (AHC) [1].

The clustering process in AHC is incremental. First, pairs of vectors $\mathbf{x}_h, \mathbf{x}_k$ that are closest in the factorial space are merged. Here, the Euclidean norm L_2 is used, which corresponds to the χ^2 distance on the original space [6]. An image whose vector $\mathbf{x}_h$ is isolated from all the other ones, i.e. $\min_k \|\mathbf{x}_h - \mathbf{x}_k\| > T$, where T is a distance threshold, is kept as an individual cluster. In the next step, a newly-formed cluster can be merged with another cluster. Rather than using distances between cluster centroids as the merging criterion, a variance criterion is employed, in particular the *within-class variance*. A cluster $C_i = \{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, ...\}$ (possibly composed of a single image) is thus merged with the cluster C_j which minimizes the internal variance of the new merged class $C_i \cup C_j$. By repeatedly performing this clustering procedure, a binary tree is constructed with approximately $2N$ nodes and depth $\log N$. Each node stores information on its center of gravity, its variance, and a pointer to the individual pictures forming the cluster.

It is possible to use the above binary tree as an a hierarchical index structure pointing to the images, by labelling each node with a variable-length binary code $(b_1 b_2 ... b_n)$, where $n \le \log N$ identifies the depth of the node in the tree, and the value of $b_m = \{0, 1\}$ encodes, respectively, a left or right branch from the ancestor node at level m. Once the tree is constructed for the whole database, this index structure can be used to rapidly extract the images that are most similar to a query image. The retrieval procedure is as follows. First, a feature vector $\mathbf{g}$ is computed from the query image and transformed into the factorial space, yielding a compact vector $\mathbf{x}$. The search then starts from the root node, and computes the distance of $\mathbf{x}$ from the centers of gravity of the two clusters represented by the children nodes (0) and (1). The cluster with the closest center of gravity (e.g. (1)) is selected, and the search continues down to its children (10), (11). This process is repeated until the number of images represented by a node corresponds to the number F of desired responses from the database. The pointers to the individual images stored with the selected node are then used to access and display the retrieved images to the user. In case of uniform distribution of the image database in the factorial space, the classification tree is balanced, in which case the search has $\log N/F$ complexity.

4. EXPERIMENTAL RESULTS

The methods described above have been applied on a database of images of various type and content, currently containing $N = 54$ gray-level images (cf. Figure 2). For demonstration purposes, and in consideration of the unconstrained nature of the images, the choice of the attributes computed from each image was based on the following criteria: use of standard algorithms, ease of computation, low dimensionality, and invariance to translation, rotation, and scale changes. In practice, these attributes were obtained by 1st- and 2nd-order statistical moments, computed on the raw *image intensity*, as well as on three classes of geometric primitives: *line segments* and *circular arcs* (approximating the image contours), and *uniform regions* (obtained through region growing). The corresponding attributes were the following: average μ_G (attribute no. 1) and standard deviation σ_G (2) of image intensity; average μ_{SL} (3) and standard deviation σ_{SL} (4) of segment lengths; average μ_{SO} (5) and standard deviation σ_{SO} (6) of segment orientations; average μ_{AL} (7) and standard deviation σ_{AL} (8) of circular arc lengths; average μ_{AR} (9) and standard deviation σ_{AR} (10) of circular arc radii; average μ_R (11) of the standard deviation of region intensities (measure of regions homogeneity); standard deviation σ_R (12) of the averages of region intensities (measure of image homogeneity). Attribute 5 was found to be unsuitable, and was excluded from the following analysis.

Figure 1 shows the projection of the $N = 54$ images and $M = 11$ attributes into the factorial plane spawned by the first two factorial axes, which explain 67.0% of the total variance of the data. Although more factorial axes should be used for a finer analysis, the display of these two axes already allows one to understand the properties of the current data set. Inspection shows distinct image clusters (e.g. of images no. 13, 23, 35, 38), isolated elements (e.g. image 5), and large clouds. The distances in factorial space between these images correspond well to their subjective differences, and will be reflected in the hierarchy of classes yielded by AHC. It can also be seen that some attributes appear significant, i.e. well separated (e.g. μ_G, σ_G, σ_{SL}, denoted respectively by 1, 2, 4), whereas other ones are highly correlated (e.g. σ_{SO}, μ_{AL}, σ_{AL}, σ_{AR}, respectively 6, 7, 8, 10). Some attributes of the latter group may thus be dropped to reduce the computational cost. The dominant factors in explaining the first factorial axis are found to be, in the order, μ_G, σ_R, σ_{SL}, σ_G. Another type of reasoning can be done on the relationships between images and attributes. The proximity of attribute σ_G (2) to images 10 and 48, for instance, means that these images are very representative of attribute σ_G, or conversely, that characteristic σ_G is a key factor in positioning images 10 and 48 in this factorial plane.

Figure 2 presents some of the results that can be obtained by the AHC algorithm described above for hierarchical indexing. In this experiment, the number L of factors used for computing the distance and moments was limited to 4, which explains 90.6% of the total variance. This provides a binary tree that can be cut at various levels to display the corresponding images. The clusters shown in Figure 2 correspond to those at level 3 (i.e. whose bit-length is 3). For two large classes, also the subclasses obtained at one or two further levels are displayed. The dominant factors in explaining these classes are those mentioned above. For instance, it is clear that the 8 classes are mainly ordered according to the first axis, i.e. the average gray level. Furthermore, the classes in the middle (codes: 100, 101, 110, 11100) contain high-contrast images, suggesting that the second factor axis σ_G was significant in determining them. Geometric features also play an important role, as shown by the segregation between images containing geometrically simple, large-scale features (e.g. classes 000 to 011) and complex small-scale ones (e.g. classes 11110, 11111). Some classes contain images without clear relationships from the perceptual point of view (e.g. classes 11100, 11101); this can be explained in terms of the globality of the attributes (average values and standard deviations), and of the small dimension of the reduced factorial space. However, a number of classes can be identified, which do group perceptually-similar pictures, for instance class 001 (paperclips), class 1000 (interior scenes), class 1001 (stamps), class 101 (playing cards), and classes 11110 and 11111 (biological images).

REFERENCES

[1] E. Diday, G. Govaert, Y. Lechevallier and J. Sidi (1981). Clustering in pattern recognition. In: J.-C. Simon, R. Haralick, Eds., *Digital Image Processing*, D. Reidel Publ. / Kluwer Inc., 19-58.

[2] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker (1995). Query by image and video content: the QBIC system. *IEEE Computer*, September, 23-32.

[3] Y. Gong and M. Sakauchi (1995). Detection of regions matching specified chromatic features. *Comp. Vision and Image Understanding*, 61, 2, March, 263-269.

[4] R. H. Gueting (1994). An introduction to spatial database systems. *V.L.D.B. Journal*, Special Issue on Spatial Databases Systems, H.-J. Schek, Ed., 3, 4, 357-399.

[5] V.N. Gudivada and V.V. Raghavan, Eds (1995). Special Issue: Finding the right image: Content-based image retrieval systems, *IEEE Computer*, September.

[6] M. Jambu (1991). *Exploratory and Multivariate Data Analysis.* Statistical Modeling and Decision Science Series, Academic Press.

[7] T. Kato (1992). Data-base architecture for context-based image retrieval. *Proc. SPIE, 1662: Image storage and retrieval systems*, 112-123.

[8] A. Pentland, R.W. Picard, and S. Sclaroff (1994). Photobook: tools for content-based manipulation of image databases. (Storage and Retrieval for Image and Video Databases II, San Jose, CA, USA, 7-8 Feb. 1994). *Proceedings of the SPIE - The International Society for Optical Engineering*, 2185, 34-47.

[9] T. Pun and R. Milanese (1995). Computer vision and multimedia information systems. In: M. Sakauchi and R. Jain, Eds., *Proc. International Workshop on Multimedia Information Systems and Hypermedia*, Tokyo, Japan, March 23-24, 29-37.

[10] T. Pun, D. Hochstrasser and C. Pellegrini (1988). Correspondence analysis and hierarchical classification of complex images: application to two-dimensional gel electrophoretograms. In: J.L. Lacoume, A. Chehikian, N. Martin and J. Malbos (eds), *Signal Processing IV, Theories and Applications*, North-Holland.

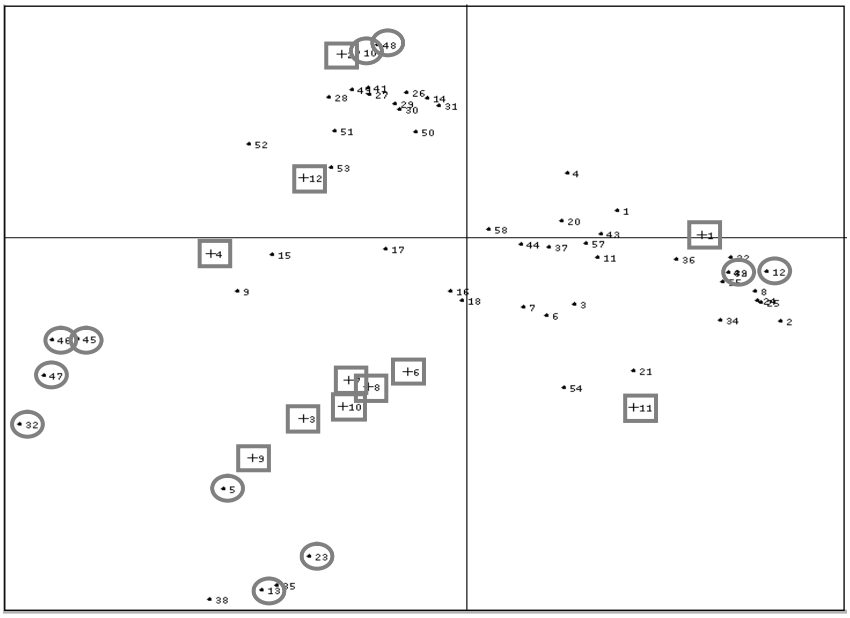

Figure 1: Factor subspace spawned by the first and second axes. Images are indicated by a dot (.) and numbered from 1 to 54; circles correspond to images which contribute the most to the factor axes. Attributes are indicated by a cross (+) surrounded by a grey square; they are numbered from 1 to 12 corresponding respectively to μ_G, σ_G, μ_{SL}, σ_{SL}, μ_{SO}, σ_{SO}, μ_{AL}, σ_{AL}, μ_{AR}, σ_{AR}, μ_R, σ_R.

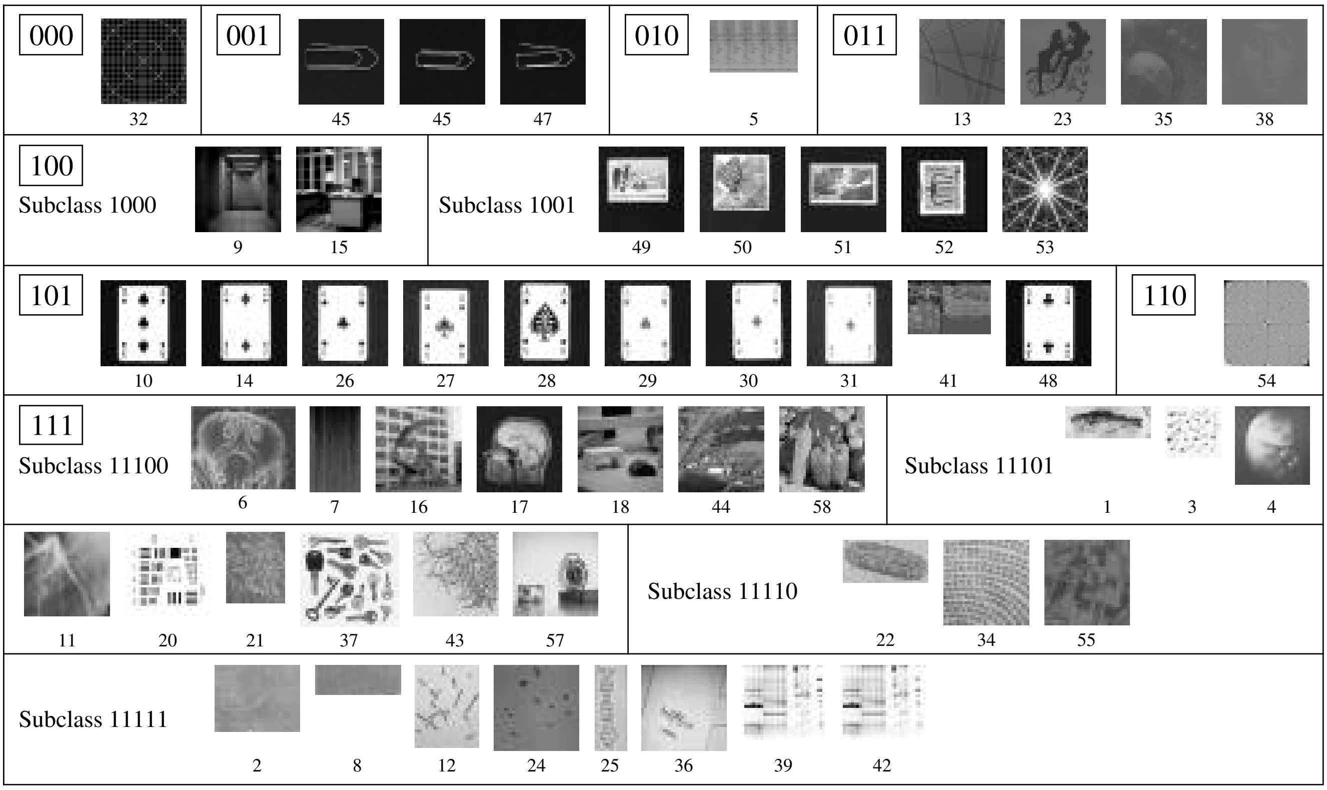

Figure 2: Image classes at the 3rd level of the indexing tree produced by the AHC algorithm, using $L = 4$ factor axes. The codes below each image correspond to the numbers indicated in Figure 1. Some dark images have been slightly enhanced for print out.