Occurring in Chemical Reactions. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 137-140. (d) Bersohn, M. An Algorithm for Finding the Intersection of Molecular Structures. *J. Chem. Soc., Perkin Trans. 1* **1982**, 631-637. (e) Dubois, J.-E. Computer-Assisted Modelling of Reactions and Reactivity. *Pure Appl. Chem. 53*, 1317-1327. (f) Sicouri, G.; Sobel, Y.; Picchiottino, R.; Dubois, J.-E. Système DARC. Localisation des Variations sur l'Invariant d'une Reaction: Concept de Structure Transformante. *C. R. Acad. Sci., Ser. 2*, 523-528.

(29) (a) Carhart, R. E.; Smith, D. H.; Brown, H.; Djerassi, C. Applications of Artificial Intelligence to Chemical Inference. 17. An Approach to Computer-Assisted Elucidation of Molecular Structure. *J. Am. Chem. Soc.* **1975**, *97*, 5755-5762. (b) Sasaki, S.-I.; Abe, H.; Hirotie, Y.; Ishida, Y.; Kudo, Y.; Ochiais, S.; Saito, K.; Yamasaki, T. Chemics-E: A Computer Program System for Structure Elucidation of Organic Compounds. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 211-222. (c) Lipkus, A. H.; Munk, M. E. Combinatorial Problems in Computer-Assisted Structural Interpretation of C13 NMR Spectra. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 38-45. (d) Shelley, C. H.; Hays, T. R.; Roman, R. V.; Munk, M. E. An Approach to Automated Partial Structure Expansion. *Anal. Chim. Acta* **1978**, *103*, 121-132. (e) Carhart, R. E.; Smith, D. H.; Gray, N. H. B.; Nourse, J. G.; Djerassi, C. GENOA: A Computer Program for Structure Elucidation Utilizing Overlapping and Alternative Substructures. *J. Org. Chem.* **1981**, *16*, 1708-1718. (f) Dubois, J.-E.; Carabedian, M.; Dagane, I. Computer-Assisted Elucidation of Structures by Carbonates NMR. The DARC-EPIOS Method: Characterization of Ordered Substructures by Correlating the Chemical Shifts of Their Bonded Carbon Atoms. *Anal. Chim. Acta* **1984**, *158*, 217-233.

(30) Freeland, R. G.; Funk, S. A.; O'Korn, L. J.; Wilson, G. A. The CAS Registry System. II. Augmented Connectivity Molecular Formula. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 94-105.

(31) Ray, L. C.; Kirsh, R. A. Finding Chemical Records by Computer. *Science* **1957**, *126*, 814-818.

(32) Moers, C. N. *Ciphering Chemical Formulas.* Zatopleg System Zator Technical Bulletin 59; The Zator Co.: Boston, 1951.

(33) Minsky, M. A Framework for Representing Knowledge. In *The Psychology of Computer Vision*; Winston, P., Ed.; McGraw-Hill: New York, 1975.

(34) Quillian, M. R. Semantic Memory. In *Semantic Information Processing*; Minsky, M., Ed.; MIT Press: Cambridge, MA, 1968.

(35) Hodes, L. Clustering a Large Number of Compounds. 1. Establishing the Initial Method on an Initial Sample. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 66-71.

(36) (a) Feldmann, A.; Hodes, L. An Efficient Design for Chemical Structure Searching. I. The Screens. *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 147-152. (b) Feldmann, A.; Hodes, L. Substructure Search with Queries of Varying Specificity. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 125-128.

(37) Lynch, M. F. Screening Large Chemical Files. In *Chemical Information Systems*; Ash and Hyde, Eds.; Ellis Horwood: Chichester, U.K., 1975.

(38) Abe, H.; Okuyama, T.; Fujiwara, I.; Sasaki, S.-I. A Computer Program for Generation of Constitutionally Isomeric Structural Formula. *J. Chem. Inf. Comput. Sci.* **1984**, *24*, 220-229. Funatsu, K.; Migabayashi, N.; Sasaki, S.-I. Further Development of Structure Generation in the Automated Structure Elucidation Program CHEMICS. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 18-28.

(39) Lefkowitz, D. Substructure Search in the MCC System. *J. Chem. Doc.* **1968**, *8*, 166-173.

(40) Rossler, S.; Kolb, A. The GREMAS System, an Integral Part of the IDC System for Chemical Documentation. *J. Chem. Doc.* **1970**, *10*, 128-134.

(41) Dubois, J.-E.; Panaye, A.; Picchiottino, R.; Sicouri, G. Systeme DARC: Structure de l'Invariant d'une Reaction. *C. R. Acad. Sci. Ser. 2* **1985**, *295*, 1081-1086.

(42) Dubois, J.-E.; Attias, R. ITODYS Internal Report, Nov 1979.

(43) Dubois, J.-E.; Sobel, Y. DARC System for Documentation and Artificial Intelligence in Chemistry. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 326-333.

(44) The Markush Search System derived from the substructure search system comprises two kinds of structural screens: specific FRELs (extracted from the invariant part) and generic FRELs describing specific aspects of the invariant structure and generic aspects of the variable structural moieties. Lourdin, C. Traitements des Formules Generiques des Brevets dans le Systeme DARC. Thesis, University of Paris, 1976.

Automatic Processing of Graphics for Image Databases in Science

ROBERTO ROZAS* and HUGO FERNANDEZ

Department of Chemistry, University of Santiago de Chile, Casilla 5659, Santiago 2, Chile

Received April 4, 1989

Generation of a database for automatic characterization, storage, and retrieval of graphic scientific information is presented. The system makes use of a scanner that allows one to digitize any graphic information. The software developed for processing the digitized images generates unique descriptors for representing the images. The system also provides a cubic spline treatment for the stored descriptors to get polynomial coefficients which are used to retrieve any graphic correlation such as a spectrum or a $y = f(x)$ representation. The original image is also compacted for further utilization. The automation provided by this graphic or image database (IDB) makes the classification and retrieval treatment human independent.

INTRODUCTION

Alphanumeric databases offer great benefits in selective retrieval of specific records within huge amounts of information.[1-3] Knowledge communication in science, however, is normally alphanumeric and also graphic and sometimes is essentially graphic. This reality makes attractive the possibility of having a graphic or image database (IDB). The use of IDBs is important in science due to the need to study similar patterns or correlations found by previous researchers or even to establish new correlations. For instance, if we have a graphic pressure-volume representation, a NMR or an IR spectrum, we can retrieve it or retrieve its similar curves according to its shape.

The IDBs known in the literature have been implemented in a way in which the graphic information is characterized by alphanumeric attributes.[4,5] These attributes are externally incorporated to the database by human processing, not deduced by a program, even when several theoretical algorithms for automatic characterization of curves have been proposed.[6]

In this paper we present a system that automatically works out the scientific graphic information necessary for the storage and further selective retrieval of images. This is done with a scanner[7] and a specially designed program for image processing and for graphic representation and management. Representation of the graphic information is based on cubic splines,[8] whose coefficients allow for the comparison of the images in the retrieval process.

GENERAL DESCRIPTION OF THE SYSTEM

Let us suppose we want to put into the IDB a record (imaginary) constituted of alphanumeric and graphic information such as

Electroantennogram correlations for pheromone compounds
J. J. Spencer et al.
J. Volat. Sci. **1980**, *40*, 200-214

0095-2338/90/1630-0007$02.50/0 © 1990 American Chemical Society

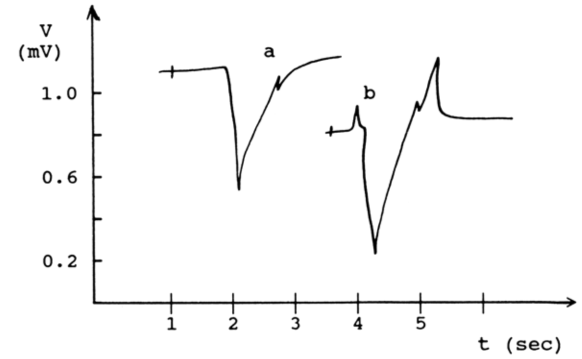

Figure 1. Typical electroantennograms for pheromones and mimics.

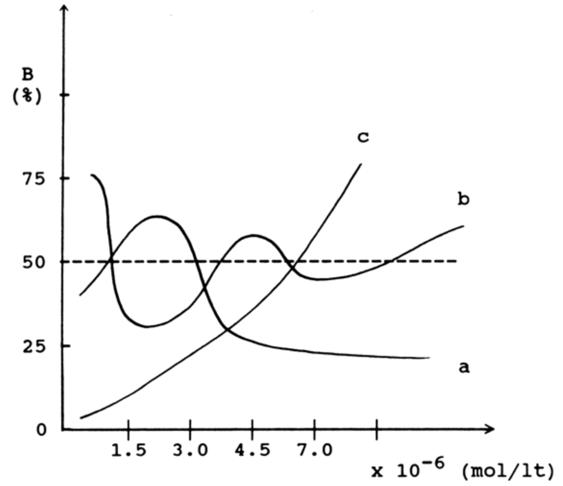

Figure 2. Typical correlations between electroantennograms and bioassays for pheromones as a function of their concentration.

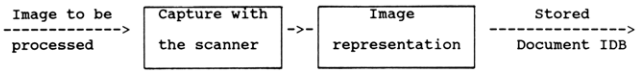

Figure 3. General diagram of graphic processing.

> Summary: Complex correlations represented in Figures 1 and 2 have been found for several pheromones....

The storage of the alphanumeric information is done in the usual way. In addition, the graphic information is presented to a scanner where a selection of the window to be processed is available. The images are digitized with this device (capture), and these images are treated for ASCII representation. In this step the image is submitted for automatic processing.

The system for managing the graphic information can be represented roughly in Figure 3.

The image representation is based on cubic splines.[8] For such a task, first, the image has to be delineated. Then, a contour recognition is made, and the delineated image is decomposed into fragments. Finally, some descriptors of these fragments allow for the representation of the image. In a pictorial way, if an image to be processed has graphic information like that of Figure 4a, the system delineates it to get a curve as in Figure 4b, which allows for the final alphanumeric description (Figure 4c).

The alphanumeric attributes generated by the system, which are the descriptors of the processed graphics, are stored together with the compacted nonprocessed digitized image in the IDB.

Retrieval of a specific graphic within the IDB is done by submitting it to the scanner to obtain the corresponding alphanumeric attributes, a process similar to that described for the input of an image into the IDB. Once this graphic is properly described, it is taken by the retrieval program to make a sequential indexed search within the IDB.

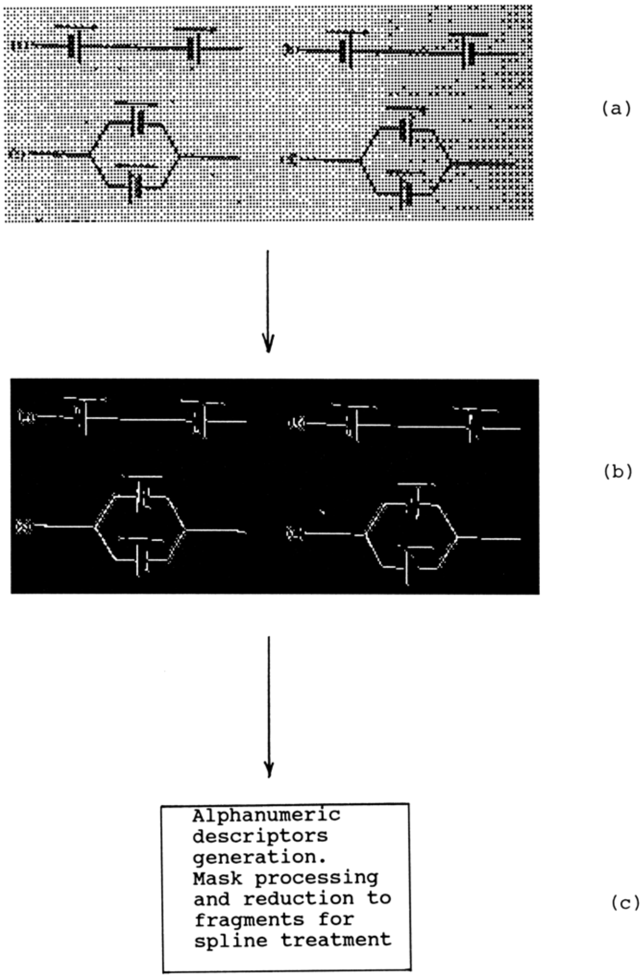
(a)
(b)
Alphanumeric descriptors generation. Mask processing and reduction to fragments for spline treatment
(c)

Figure 4. Diagram showing the global treatment of the graphic information.

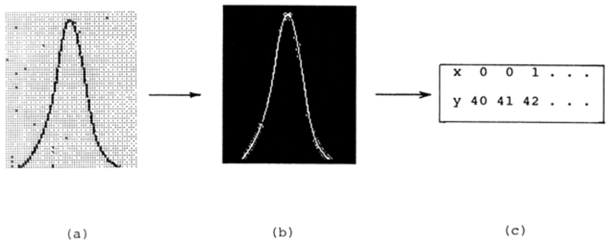

Figure 5. Steps involved in contour recognition: (a) original image; (b) delineated image; (c) bidimensional representation of the image.

COMPUTATIONAL DESCRIPTION

(a) Graphic Representation. We present below the whole treatment given to scientific graphic information. The image, captured with 16 gray-scale levels, from 0 (the darker level) to *h* (the lighter level), is transformed into a binary image with 0 and *h* only to capture its contour, which is represented by *h*. This process is shown in Figure 5a,b.

The contour of the delineated image is stored as a bidimensional numeric structure (Figure 5c). This structure permits break point determination according to the method of cubic splines.[8] By this treatment the image is represented by fragments separated by break points, which can be seen in Figure 6. Each fragment is described by three pairs of coordinates (see Figure 6c). The first and the last pairs are the break points, and the second pair is the median point. Due to the approximations needed in the spline processing, these coordinates are worked out in the program as real numbers,

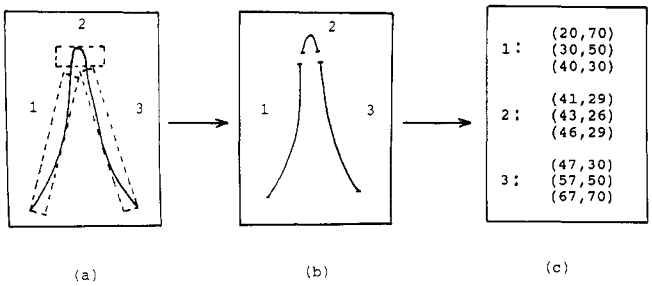

Figure 6. Treatment of a delineated image: (a), (b) visual representation of curve fragments; (c) coordinates in matrix format, of each fragment, stored for further spline coefficient determination.

but in the IDB they are stored as integer numbers.

Because any digitized curve is composed of four smaller continued lines that are vertical, horizontal, 45°, or 135°, an algorithm was developed for the theme of this work to obtain a delineated image (steps a and b of Figure 5). For this purpose, four templates were used to characterize each of the four component lines of an image (the negative image is used). Each template was applied to the eight neighbors of a pixel in the negative image, $f(x,y)$. For a pixel of the new image, $g(x,y)$, generated from $f(x,y)$ after application of the template, we have

for horizontal lines

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{aligned} g(x,y) &= 2(f(x,y-1) + f(x,y) + f(x,y+1)) - \text{neg} \\ \text{neg} &= f(x-1,y-1) + f(x-1,y) + f(x-1,y+1) + f(x+1,y-1) + f(x+1,y) + f(x+1,y+1) \end{aligned}$$

for vertical lines

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{aligned} g(x,y) &= 2(f(x-1,y) + f(x,y) + f(x+1,y)) - \text{neg} \\ \text{neg} &= f(x-1,y-1) + f(x,y-1) + f(x+1,y-1) + f(x-1,y+1) + f(x,y+1) + f(x+1,y+1) \end{aligned}$$

for 45° lines

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad \begin{aligned} g(x,y) &= 2(f(x+1,y-1) + f(x,y) + f(x-1,y+1)) - \text{neg} \\ \text{neg} &= f(x-1,y-1) + f(x-1,y) + f(x,y-1) + f(x,y+1) + f(x+1,y) + f(x+1,y+1) \end{aligned}$$

for 135° lines

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \quad \begin{aligned} g(x,y) &= 2(f(x-1,y-1) + f(x,y) + f(x+1,y+1)) - \text{neg} \\ \text{neg} &= f(x-1,y) + f(x-1,y+1) + f(x,y-1) + f(x,y+1) + f(x+1,y-1) + f(x+1,y) \end{aligned}$$

Each template is applied once to the original image to get an image with one type of enhanced line (e.g., horizontal); then a segmentation is applied to this image to get a binary image with one type of line. After this process is applied for each of the four templates, a Boolean OR operation is applied to all binary images to finally get the delineated image. This process is represented in Figure 7. The threshold value (9) shown in Figure 7 results from an empirical trial and error method for this kind of scientific image.

In this way coordinates (Figure 6c) are finally stored in the IDB. Once in the IDB, they are ready to be processed for retrieval through a spline coefficient determination. Such coefficients are calculated according to the equations given under Retrieval. The spline treatment is done to compare just the coefficients of the searched graphic with the coefficients of each image stored in the IDB. The described method is for representation of curves normally found in scientific graphics, as a spectrum or correlations between different parameters. If the image has solid shapes, a different delineating method[9] should be used.

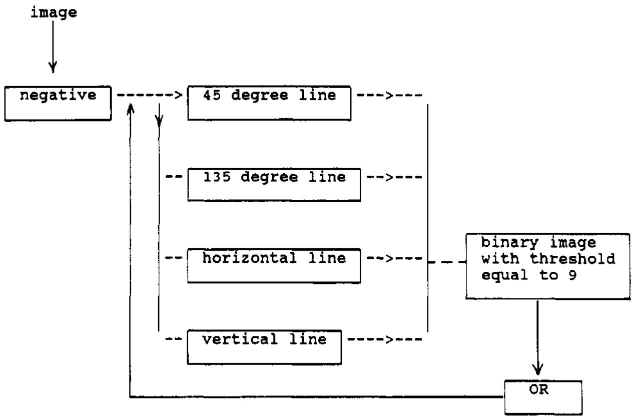

Figure 7. Block diagram of the delineation process.

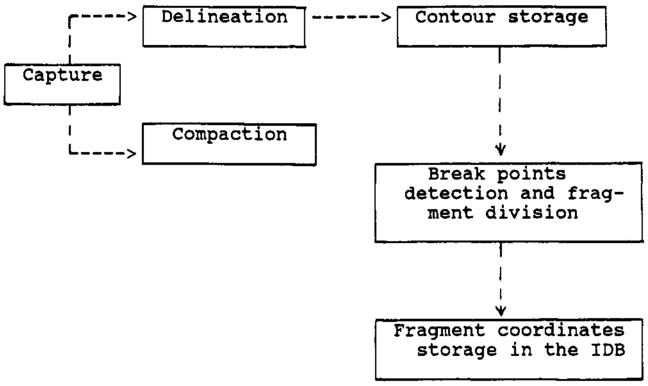
Delineation
Contour storage
Capture
Compaction
Break points detection and fragment division
Fragment coordinates storage in the IDB

Figure 8. Diagram of the whole storage process applied to an image.

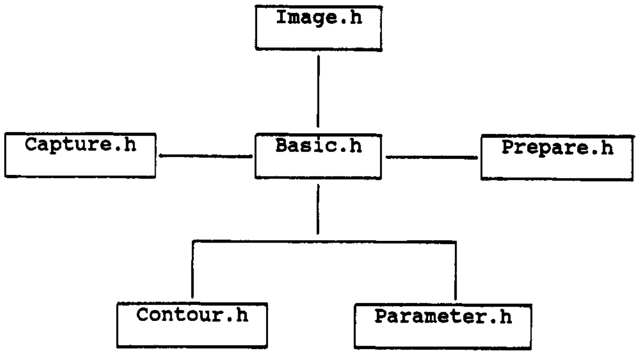

Figure 9. Libraries of the IDB.

Simultaneously with the input of this derived information, the original captured image is submitted to compaction because it is necessary to have the real image with minimum storage. For this reason, the compaction method provided by the VAX Ultrix system is applied. The whole process, applied to an image and described in previous steps, can be represented as in Figure 8.

For the detection of the break points a mask[8] is applied to the curve, defined by

$$\frac{(xn - x0)yj - (yn - y0)xj + (x0yn - xny0)}{[(xn - x0)^2 + (yn - y0)^2]^{1/2}} \leq \text{MAX}$$

where $x0$, xn, $y0$, and yn are the coordinates of the extreme points of the fragments submitted to the mask, (xj,yj) is any point of such fragment, and MAX is the numerical value of the mask width related to the range of precision imposed to each curve fragment (12.6 in this work). If the expression is true, the process continues; if not, (xj,yj) is a break point.

(b) Database Organization. The library distribution of programs and subroutines for the IDB is represented in Figure 9. Capture.h has subroutines that take an image from the digitized file and convert this file to a matrix format. Image.h allows one to make transformations (smoothing, equalizing, gradient, etc.) to the images already in the matrix format. Prepare.h transforms an image from matrix format with the gray-scale levels into a delineated image. Contour.h has the routines in charge of further processing of the delineated image. Parameter.h contains only definitions of parameters used in other libraries for image processing. Basic.h is the library where the routines for management of the database are located; it also allows one to coordinate functions and routines implemented in other libraries.

The entities of the IDB are shown below as files with their respective sizes in bytes:

data.tex (input by the user except the first parameter)		
code of the image		01
name of the image		20
title		20
author		15
page		02
descriptor.tex (generated by the system)		
code of the image		01
coordinates of each fragment		
coord x	3 times	02
coord y		02
original points		02
new points		02

The files of this IDB are sequentially indexed by the access code of the image. Each file has fixed-length records. The access to a record of a file is done through the fseek function of the C language.[10] Other functions for accessing a file are putrec_d and putrec_i, which store a record in the file of descriptors and image data, respectively. The opposite objective is done through getrec_d and getrec_i. The parameter pointer, number of register, and name of the variables are passed to these four functions. For instance

getrec_i(im,nam,tit,aut,&pag,&x0,&xf,&y0,&yf,fp)

gets the complete record of the file representing the fundamental data of an image. Here im is the code of the image; nam, tit, aut, and pag are the name, title, author, and page of the image respectively; x0, xf, y0, and yf are the extreme coordinate values considered.

File linkage is of logic type; for instance, data.tex and descriptor.tex are linked by the access code field of the image. Linkage between a record of the data.tex file and the file that has the compacted image is done through the code of the record of data and the name of the file containing the image. For instance, code number = 26 means that the content of the image is in file I26.C; this is done with functions provided by the system.

(c) Retrieval. Selective retrieval of specific images is based on matching of fragments of the pattern searched and those of the stored images in the IDB. A comparison for all the fragments is carried out by calculation of the spline coefficients[11] for each fragment using their stored coordinates. Once this is done for all the fragments, the final matching is done between the coefficients of the fragments; this allows one to decide if an image is equal or not to another one. The coefficients used for comparison of the fragments are

$$C1x = -\tfrac{3}{2}x_1 + 2x_2 - \tfrac{1}{2}x_3$$

$$C2x = \tfrac{1}{2}x_1 - x_2 + \tfrac{1}{2}x_3$$

$$C1y = -\tfrac{3}{2}y_1 + 2y_2 - \tfrac{1}{2}y_3$$

$$C2y = \tfrac{1}{2}y_1 - y_2 + \tfrac{1}{2}y_3$$

where (xi,yi) are the coordinate points of each fragment. Then if two fragments have their four coefficients equals, such fragments are equals.

Table I. Run-Time of the Program Subroutines for an Image of 227 × 194 Pixels

routine	action	time, s
readimage( )	change image to matrix format	50
delineate( )	delineate an image	48
genercont( )	read contour	04
pack( )	segment contour into fragments	128
	compaction	15
	comparison	02
equaliz( )	equalize an image	01
smooth( )	smooth an image	06
gradient( )	get the gradient of an image	08
negative( )	get the negative of an image	01
	uncompact	15
binary( )	segment an image	01

(d) Statistics of the Software. The amount of memory necessary for data storage for each image is as follows: record length of the data.tex file, 66 bytes; record length of the descriptor.tex file, 16 bytes; median size of a compacted image, 10 kbytes; mean number of descriptors, 110. To store n images we will have

$$\begin{aligned} M(n) &= 66n + 110 \times 16n + 10240n \text{ bytes} \\ &= 12066n \text{ bytes} \\ &= 11.783n \text{ kb} \end{aligned}$$

Therefore, to store 255 images, we need 11.783 × 255 = 3004.717 kb, or 2.9 Mb. In a more general description, for a mean descriptor number d and an average size I (kb) of the image we will have

$$M(n) = 66n + nd16 + 1024In \text{ (bytes)}$$

The run time of the input function is determined by two components, one for creation of the input environment (25 s) and the other for compaction and descriptor extraction, which is image size dependent. So the input time T_i is given by

$$T_i = 25 + T_{\text{comp}} + T_{\text{descr}} \text{ (s)}$$

The run time of the subroutines involved in a specific image processing is given in Table I.

The retrieval process is divided into three steps: (a) creation of the retrieval environment (28 s); (b) pattern codification; (c) search in the IDB. Therefore, the retrieval time T_r will be

$$T_r = 28 + T_{\text{codif}} + T_{\text{search}} \text{ (s)}$$

The system was developed for a microVax II, under the Ultrix operating system, but the capture step was done with a PC-AT microcomputer, under MS-DOS, to which the scanner HP-Scanjet was linked. Communication between the PC and the microVax was done through Kermit. The software was done entirely in C language (Ultrix C compiler). Currently the system is being converted to turbo_C to run it on a PC-AT under DOS, but here the dimensions of the matrix to be processed are limited (main memory), i.e., for two matrices, the allowed maximum number of elements is 173 × 173 for each matrix. This means that in a PC-AT it is possible to process two images of a maximum size of 2.3 × 2.3 in. at a resolution of 75 × 75 dpi.

As was previously established, the process is based on fragment comparison. Such a comparison is done by the identity of the coefficients of its fragments. Because the image captured has some original noise, and due to the distortion introduced by the whole image processing, realization of such a theoretical identity is not absolute. Therefore, it is convenient to introduce a range of coincidence for the coefficients sub-

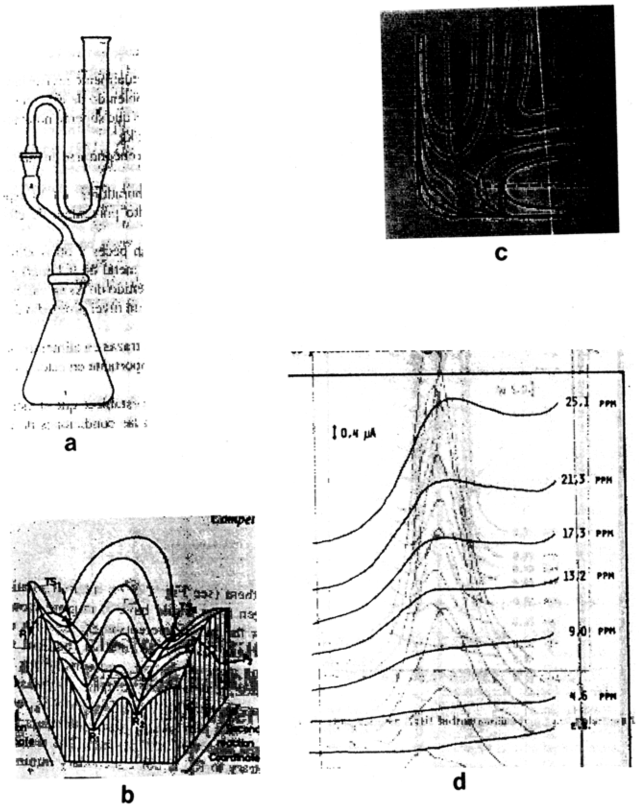

Figure 10. Images in the database.

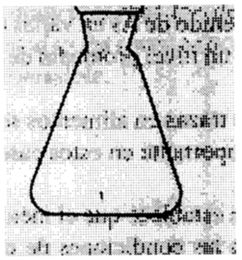

Figure 11. Image to be searched with no coordinate specifications.

mitted to comparison. To that aim the C1 coefficients for those fragments under comparison are allowed to vary within 0.0–0.5, and the C2 coefficients are allowed to vary within 0.0–1.0. So

$$\text{if abs}(\text{C1\_imagl} - \text{C1\_imag2}) \leq 0.5$$

and

$$\text{if abs}(\text{C2\_imagl} - \text{C2\_imag2}) \leq 1.0$$

the coefficients are considered as equals. This is the range of variation for identity, both in C_x and in C_y, considered by the system.

SAMPLE SEQUENCES

We present now two examples of how the system works, using Figures 10–12. These figures appear with some mottled background because they have been taken directly from a photocopy of journals with 16 gray-scale levels. This shows the ability of the software developed to cope with degraded photos. With better photographs like Figures 1 and 2, the system also works properly. For an IDB with 100 images like those shown in Figure 10, with their respective alphanumeric attributes input by the user and those generated by the system itself, we can ask for all the images containing a figure like the one shown in Figure 11. The system, in just a couple of minutes, will show the image of Figure 10a, with all the information associated with it. If the IDB has other images containing Figure 11, the system will also recover them.

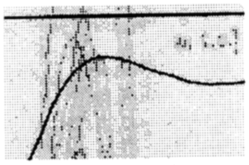

Figure 12. Image to be searched with coordinate specifications.

If we ask for images having the correlation of Figure 12, the system will show all images containing it, such as those of Figure 10d. If we additionally introduce alphanumeric restrictions such as coordinates

$$x\text{: } (0.5\text{–}30) \times 10^{-5}\ \text{A}$$

$$y\text{: } 0\text{–}3.5\ \text{V}$$

then the system will find images like those of Figure 10d but having coordinates within the range imposed to the search. This means that other images, which after comparison by the method described under Retrieval, even containing Figure 12, will not be accessed if they do not fit the alphanumeric conditions.

The system also works properly with intersecting curves and with segmented lines as those described by the points TS_1 and P_1, respectively, in Figure 10b.

DISCUSSION AND CONCLUSIONS

One of the characteristics of the IDB described above is that it permits retrieval of any previously stored graphic correlation, independent of the relative position of the pattern we are looking for in the images of the IDB. Thus, if we are looking for a pressure–volume correlation, we will find it wherever it exists: in the middle, upper, or lower parts of the image or at the right or left of the image. The only restriction is that the image containing the pattern we are looking for must not be rotated (cf. Figure 4).

The reliability of the retrieval of any image is between 0.5 and 5% error. Thus, if the pattern we are looking for is complex, we can get some images not containing the real pattern; also, we can lose some images that are not found within the IDB even though they contain the pattern. If the pattern we are looking for is less sophisticated, like a spectrum or a usual correlation, the error is not higher than 0.5%. For instance, for 31 images contained in the IDB and 372 comparisons (i.e., 12 different patterns for each of the 31 images) the system gave two errors, representing a 0.54% error.

The IDB developed here was implemented with 16 gray-scale levels, and it allows one to work with scientific representations having colored plotted figures, not just images in black and white. A further refinement of this IDB is the use of alphanumeric data introduced with the image such as coordinates accompanying a correlation within the image. For instance, we can look for a certain shape of a curve, but this shape can be found in images of different content; the coordinates accompanying the wanted pattern can further limit the search, and the values of these coordinates can make the search more specific.

Within this context the IDB developed here promises to be of utility such as that showed by alphanumeric databases having molecular graphic information.[1] Also, the IDB has an automatic generator of alphanumeric descriptors not explicitly included in the original graphic input information. This characteristic is also similar to other alphanumeric databases.[2] Additionally, this IDB permits one to work directly with graphic information in an automatic way that, to date, was not widely known in scientific information management. These characteristics make our IDB processing (classification, keywords, etc.) "human independent".

As long as technology continues to develop, the processing time at input and retrieval can be optimized, even when the actual time of retrieval is short enough: 2–4 min for a specific pattern within 100 images. Also, technology will permit one to work with a bigger storage capacity. Software development such as image compaction and image processing also will make it possible to work with the more sophisticated IDBs that are necessary in science. Further work in this direction is under way in our laboratory, where the reliability or accuracy of the IDB is being taken into account.

It is not difficult to realize that future chemical databases will allow one to work with alphanumeric, molecular, and graphic information in an easy way such as this paper has shown. Also, we can extrapolate this statement for science in general—the treatment of integrated alphanumeric and graphic information will be the normal way of managing scientific knowledge.

ACKNOWLEDGMENTS

We thank the ANDES Foundation, Grant C-10390/1988, for the Microvax II. We also thank Prof. M. L. Contreras for discussions and FONDECYT and the University of Santiago for financial support.

REFERENCES AND NOTES

(1) Contreras, M. L.; Deliz, M.; Galaz, A.; Rozas, R.; Sepulveda, N. A Microcomputer-Based System for Chemical Information and Molecular Structure Search. *J. Chem. Inf. Comput. Sci.* **1986**, *26*, 105–108.

(2) Contreras, M. L.; Deliz, M.; Rozas, R. Personal Microcomputer Based System of Chemical Information with Topological Structure Data Elaboration. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 163–167.

(3) Rumble, J. H., Jr.; Lide, D. R., Jr. Chemical and Spectral Databases: A Look into the Future. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 231–235.

(4) Prasad, B. E.; Gupta, A.; Toong, H.-M. D.; Madnik, S. E. A Microcomputer-Based Image Database Management System. *IEEE Trans. Ind. Electron.* **1987**, IE-34, 83–88.

(5) Felician, L. Image Base Management System: a Promising Tool in the Large Office System Environment. *DATA BASE* **1987/88** (Fall/Winter), 29–36.

(6) Werman, M.; Wu, A. Y.; Melter, R. A. Recognition and Characterization of Digitized Curves. *Pattern Recognit. Lett.* (*Netherlands*), **1987**, 5, 207–213.

(7) HP-Scanjet, Scanning Gallery Users Guide; Hewlett-Packard: Sunnyvale, CA, 1987.

(8) Lozover, O.; Preiss, K. Automatic Construction of a Cubic B-Spline Representation for a General Curve. *Comput. Graphics* **1983**, *2*, 149–153.

(9) Gonzalez, R. C.; Wintz, P. *Digital Image Processing*; Addison-Wesley: Reading, MA, 1977.

(10) Kerningham, B.; Ritchie, D. *The C Programming Language*; Prentice-Hall: Englewood Cliffs, NJ, 1978.

(11) Turbo Pascal, Reference Manual, Version 3.0; Borland International, 1985.

Vertex Indices of Molecular Graphs in Structure–Activity Relationships: A Study of the Convulsant–Anticonvulsant Activity of Barbiturates and the Carcinogenicity of Unsubstituted Polycyclic Aromatic Hydrocarbons

G. KLOPMAN* and C. RAYCHAUDHURY

Department of Chemistry, Case Western Reserve University, Cleveland, Ohio 44106

Received April 21, 1989

A new methodology is proposed whereby *local* distance based vertex indices are used in structure–activity studies. It is also shown that it is possible to reconstruct chemical graphs for those indices found to be relevant to activity. This is essential if the results of structure–activity analysis by methods utilizing graph indices are to be useful in the design of new active molecular entities. The methodology is illustrated by applications to the study of the convulsant–anticonvulsant activity of barbiturates and the carcinogenic activity of unsubstituted polycyclic aromatic hydrocarbons.

INTRODUCTION

Explaining biological activities of chemical compounds in terms of molecular topology has gained substantial attention in recent years.[1-18] The objective of all such studies is to explore the role of the connectedness of atoms in the expression of the biological activities of molecules. Franke et al.[8] have discussed the necessity of considering topological aspects of the chemical structures to explain their biological functions.

The connectedness, or the topology of the molecules, is conveniently expressed in two ways. One is in terms of molecular fragments or substructures,[1,2,5-8] and the other is in the form of molecular graphs and the indices derived therefrom.[3,4,10-18] While substructural analyses are designed mainly to identify the potential structural components that could be responsible for some biological activity, graph-theoretical methods are mainly used to relate the structural characteristics of chemical compounds to their biological activities. These structural characteristics include branching patterns, bonding types, cyclicity, etc.

Clearly, the substructural approaches help medicinal chemists to analyze the relationship of the molecular fragments to the biological activities of chemical compounds. However, a possible shortcoming of this approach, as well as most other approaches using discrete descriptors, is that biologically relevant substructures may be ignored if they are not present in the training data set. It appears that, topologically (in the sense of connectivity), flexible structural descriptors might play some important role in these situations. Hence, graph theory[19,20] seems to be a prime choice to cope with such problems.

The structural formulas of chemical compounds are essentially molecular graphs whose vertices and edges represent, respectively, the atoms and their connecting chemical bonds in the molecules. This straightforward representation of chemical structures has enabled chemists for decades to take advantage of this branch of mathematics to solve some relevant

* Address correspondence to this author.