E

TWC/17/5
ORIGINAL: English
DATE: May 28, 1999

UPOV

INTERNATIONAL UNION FOR THE PROTECTION OF NEW VARIETIES OF PLANTS
GENEVA

TECHNICAL WORKING PARTY ON AUTOMATION AND COMPUTER PROGRAMS

Seventeenth Session
Helsinki, June 29 to July 2, 1999

FLORES: A PICTORIAL DATABASE FOR ORNAMENTALS USING JAVA

Document prepared by experts from the Netherlands

FLORES: a pictorial database for ornamentals using JAVA

Gerie van der Heijden

SUMMARY

FLORES is an object-dependent system for image matching of ornamental varieties, where the feature extraction and matching depends on the type of object. It has user-driven segmentation tools in a cross-platform environment, using JAVA Applets. Furthermore the system allows a direct link with a relational database. The system has at this moment no means for combining similarities from different images of the same variety to determine an overall similarity. The final goal of FLORES is to serve as a digital reference collection, where varieties can be compared on basis of visual information.

INTRODUCTION

Most member states of UPOV make use of a relational database system for their variety descriptions. Pictorial information, which is especially indispensable for ornamentals, is often only stored in photobooks or the like. In some countries the pictures are stored in special image databases, e.g. in Germany and in the Netherlands. In Germany they have made a direct link between the image database and the administrative relational database. In the Netherlands such a link between picture and other variety information is not directly available, but the variety number is stored in both databases.

Even though the pictures are present in a database, the systems do not make use of the pictorial information: images are stored as binary large objects and are non-searchable items in the database. Searching has to be done by using keywords or UPOV characteristics/scores. It would be very helpful if varieties could be retrieved from the database in decreasing order similarity by the pictorial information itself.

Many attempts are being made to retrieve information from the database using the pictorial information. A nice example is the Virage [3] implementation in the Alta-Vista search engine: the AV Photo and Media Finder offers searching for visually similar images using fast complete image comparison techniques (see e.g. http://image.altavista.com/cgi-bin/avncgi). However, the similarity measures used are based on whole images and are not specific for any domain. Therefore the results can vary completely from your opinion of similarity. Therefore a more specific object (flower) based approach would be required. Some systems do allow more object specific information, like QBIC [1] and ARBIRS [2], but are still not useful in variety testing.

We propose here a system for finding the most similar varieties using the pictorial information as a helpful tool in Plant Breeders' Rights of ornamentals. The system should preferably fulfil the following demands:

1. The system should allow a new image of a candidate variety to be compared with images of a reference collections. The images may contain various backgrounds, and may contain different objects in one image, like leaves and flowers.
2. The system should be able to handle all crops in a generic way, but it should also be possible to implement feature extraction and matching specific for each crop/plant part.

Feature extraction and matching may have a general basis for flowers, but implementation details will depend on the crop, e.g. rose or tulip.

3. The system should also contain other variety information, like administrative data.
4. It should be possible/easy to share and exchange the information with different registration authorities in different countries.

Item 1 and 2 call for an object-specific approach: the system should extract different features and apply different matching algorithms, depending on the type of object (flower/leaf etc). Also, because of the many different kinds of objects, and the required details, the user should be able to control/inspect the objects in the picture. Therefore the system should provide means for automatically locating the objects, but still allow the crop expert to have control over the final objects used for matching. For combining the image information with other information, a simple unique key can suffice for a mapping between image/object and variety information in a relational database. This allows separate development of the relational database and the image database. Since the information should be exchanged by existing but different administrative systems in different countries (item 4), a good separation between image database and all kinds of relational databases is a prerequisite. Furthermore item 4 calls for secure use of an Internet/Intranet approach.

An image database system which has a object-specific approach is ARBIRS, described and developed by Gong [2]. ARBIRS has a standard module for image segmentation and texture regions. Although it does have the object specific approach, and is even capable of handling and querying multiple objects in one image, it does not distinguish between different types of objects. Therefore it only has one type of feature extraction and matching and does not allow for user-controlled segmentation.

QBIC has an object-module in the total system and also allows user-driven segmentation, but this is not possible in an Internet browser [1]. Furthermore, QBIC does not have the possibility to assign different features and matching functions for different kinds of objects.

We propose a system which allows handling of different kinds of objects, using interactive segmentation, across a network, based on JAVA. In the next section the set-up of system is given. In section 3 we discuss the future of the system.

2 SYSTEM SETUP

The system consists of two main parts:

1. Client-site: takes care of picture loading, (interactive) segmentation, sending the object to the server for matching, and showing the resulting similar images;
2. Server-site: takes care of feature extraction, feature matching/similarity calculation, sorting by similarity and sending the results to the client.
3. The client program is a Java applet. An applet is a small program that can run in a web-browser across the network. The server program makes use of an existing image-library written in the C programming language [5]. A flowchart of the system, showing all components and communication order is shown in Fig. 1. The components of the systems are described in the next paragraphs.

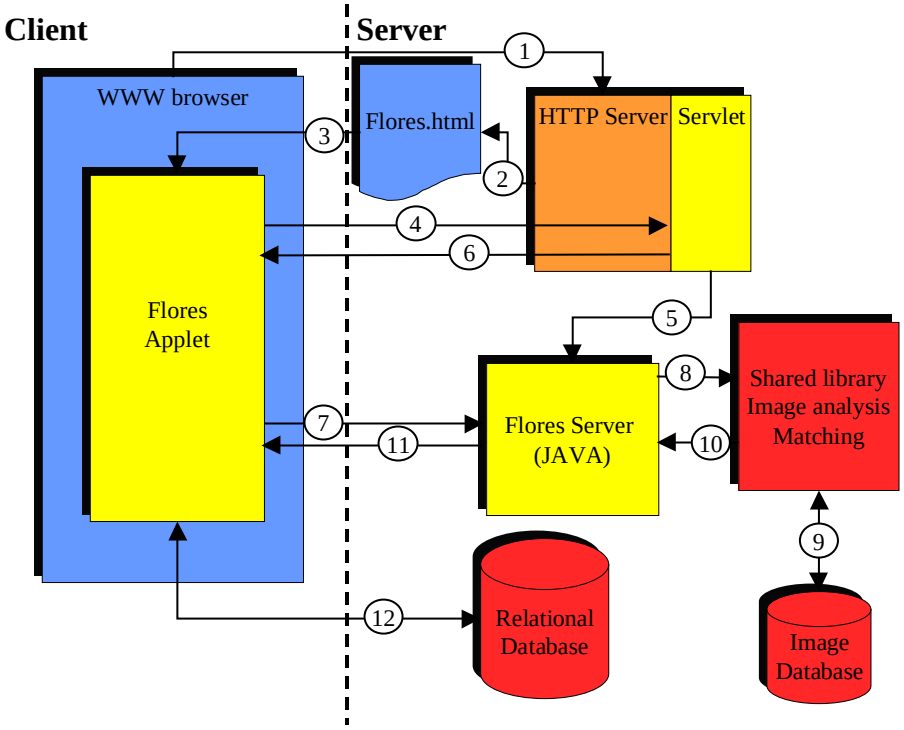

Fig. 1. A flow chart of the Flores system. The numbers indicate the flow order and are further explained in the text.

2.1 Client Site

The system starts when a user asks for the Flores.html document in his web-browser (nr 1 in Fig 1). The html-document, containing the applet is sent to the client (nr 2 and 3). The applet asks the user to load a local image (JPG of GIF at the moment). After an image is loaded, the user should select the part in the picture which is used for comparison.

Interactive Segmentation. An image can have more than one object, e.g. leafs, stem and flower(s). The applet has several tools to allow the user to indicate which part (object) of the image he/she wants to use.

The first tool is based on thresholding of the three colour histograms (for red, green and blue). The user can set upper and lower limits for the three colours and picture points (pixels) which fall outside the limits are de-selected. To make it easier for the user to set the correct limits, some good initial guesses are made by the system.

The result of changes made in the limits are directly shown in the image: de-selected points obtain a standard (user-selectable) colour, e.g. grey and selected pixels have their original colour.

Since it is difficult to obtain a correct selection with thresholding, more tools are available to make changes to the selection, e.g. to add or delete certain parts using the mouse, or to invert the mask. You can also select a single object by clicking on a pixel of the object.

Another segmentation tool is region growing which is comparable to the magic wand in several image processing programs like Adobe PhotoShop. A parameter can be set by the user to allow a smaller or higher colour distance between candidate pixels and initial pixel.

There is also a standard segmentation tool, which directly selects the correct object, if the object is recorded under fixed conditions.

After the segmentation the image contains a mask, which only allows showing of the selected object.

Database matching. After the segmentation the user can select different options for matching the image with the images in the database. The options include type of object (flower, plant, leaf, whole image) and type of matching (color histograms, combination of features). After choosing the options, the image is send to the server. Technically this is done in the following way: before the applet can send the image, it sends a message to a servlet via the http-port (nr 4 in Fig. 1). The servlet starts a socket and a server-program at that socket (nr 5 in Fig. 1) and returns the port number to the applet (nr 6 in Fig 1). Next, the applet sends the image information as a serialized object over the port to the server-program (nr 7 in Fig. 1).

Showing the results. The applet receives the results of the matching (nr 11 in Fig. 1). The results are a set of 25 records, ordered by decreasing similarity. Each record the variety registration number of the object, the similarity, and a link to the URL-address of the image. The variety registration number can be used for a link with the REX database (nr 12 in Fig. 1). The 25 images are downloaded in thumbnail format and shown in a new gallery window. By clicking on a thumbnail image, a full size image is downloaded and shown in a new window, which can subsequently be processed (segmented/matched) as described above.

2.2 Server Site

After the server has received the image, the image is split up in an RGB image (the full original image) and a binary mask image. Subsequently the image library of Scil-Image [5] is called through the Java Native Interface (nr 8 in Fig. 1), using the color image, the binary image and the matching option as parameters.

There are at the moment two options for matching: histogram based and feature based. The histogram option is a basic option, available for all objects. The feature option is object specific.

Histogram based matching. This option is fast and easily calculated. The results achieved are limited for accurate matching. It only uses basic colour information. It allows that pictures are made under varying lighting conditions, since the light intensity is discarded in the matching. The normalised red and green histogram form the basis for this similarity measure. See [6] for a more detailed description of this measure.

This measure is very fast to calculate, and can be used over the complete database without any restriction on type of object, i.e. matching based on histogram similarity is done over all objects.

Features based matching. Depending on the type of object, which is indicated by the user, various features are calculated, such as area, perimeter, length/width ratio, shape factor ($perimeter^2/(4.\pi.area)$), modus of red, green and blue (i.e. the value of the histogram bin with the highest count in the histogram) and the standard deviation of red, green and blue. Note that intensity is used in the matching process, so standardised recordings are necessary for optimal performance of the system.

After the features are calculated, they are weighted and used in linear combinations, where the weights and coefficient for the linear combinations are based on a linear discriminant analysis. This allows use of many features, reduced to a small sub set of most discriminating features. The new set of recombined and weighted features is used for searching nearest neighbours.

The most similar objects in the feature space are found using the SR-Tree algorithm [4]. The SR (sphere/rectangle) tree is an index structure which allows a very fast search of k-nearest neighbours in a high dimensional feature space. For more details see [2,4].

The feature based method makes use of assumptions of the kind of object: for each different kind of object a specific feature set is chosen, optimised weights are calculated and a separate search tree is used. Therefore finding most similar images using this method is done only within the specified type of object.

The user-interface of FLORES and search results are shown in Figure 2.

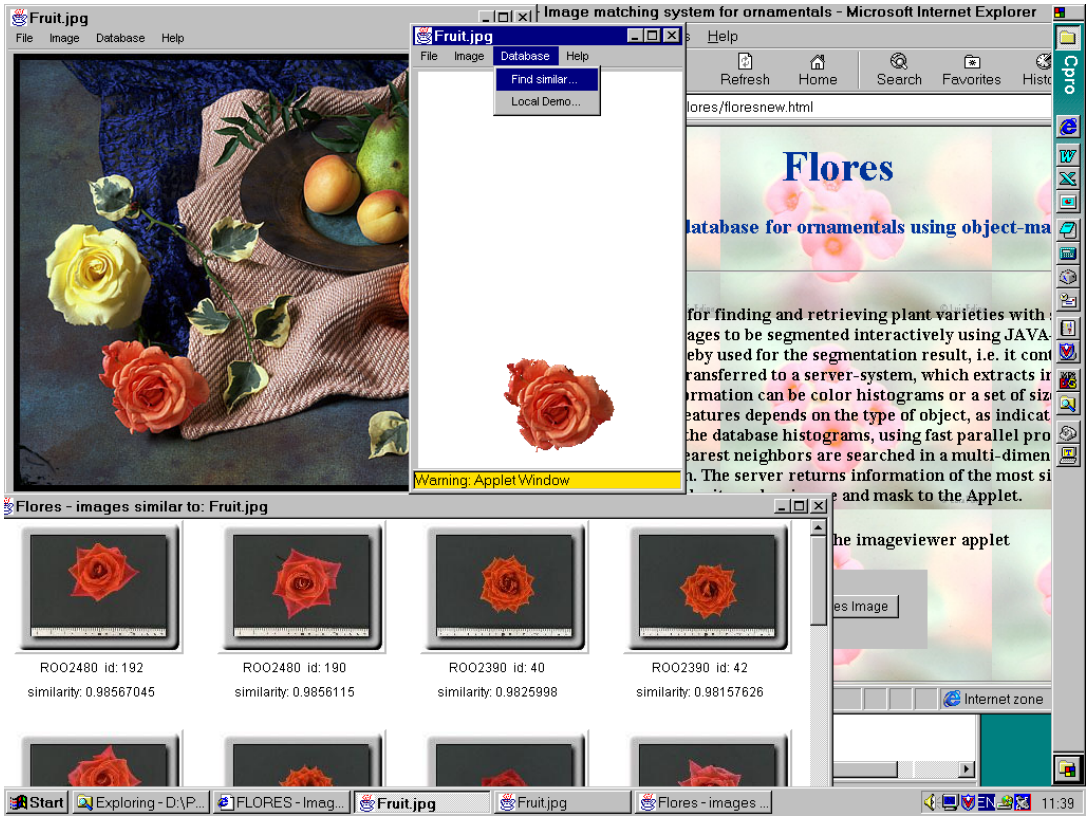

Fig. 2. The user interface of FLORES. Top left, an image can be loaded by the user, containing for example several objects. Top middle: the system allows the user to mask the image with the exception of the object of interest. The masked image can be used for finding the most similar images in the database. Bottom: After searching the database, FLORES pops up a new window with the most similar images. The variety number under the thumbnail picture can be used as a link to the administrative database.

3 RESULTS AND DISCUSSION

At the moment the (prototype) database contains images of flowers of eighty rose varieties. At least three flowers per variety are available. The images are recorded under controlled conditions, so within the database all pictures can be compared.

By allowing user-driven segmentation in the applet, the user can quickly select the correct object, masking the remaining part of the image. In this way a crucial step is set for optimal matching. If the system would perform the segmentation fully automatically, this could lead to incorrect segmentation of the object and the reliability of the system would be seriously distorted. The advantage of this approach is that we combine the expertise of human and computer. We use the human brain for the semantics (determining kind of object, controlling segmentation), and the computer for the quantitative comparison. The information that the user already has (basic knowledge of kind of object) is supplied to the system and the system focuses on detailed quantitative information, e.g. detailed shape, colour and texture information. By showing the resulting most similar images to the user, the user is still in control of the final decision.

Since multiple images of different flowers are available for each variety, the ranking of the most similar of the other images of the **same** variety can be used as measure of retrieval accuracy. For this purpose, the database was limited to 70 varieties represented by three flowers each, yielding a total of 210 images. Each image was compared with the remaining 209 images. In a random situation, the expected ranking of the most similar image of the same variety would be 70 (using Monte Carlo simulation). For the search tree, this value was 2.03. In 71% of the cases, the most similar other image of the same variety had ranking 1. We expect that these values can be improved by further optimisation of the feature set.

The system is not capable of combining several images to obtain an overall similarity for a variety. At first glance this seems a simple matter of using weighing coefficients to form a linear combination of the individual object similarities, even though it may be difficult to find the right coefficients. This is indeed true if all similarities are available, like in the histogram matching. However, the main problem is in using search trees. Then only the k_1 nearest images for one view of a variety are retrieved, which is a different set from the k_2 nearest images for another view of the variety. Combining the sets of several views may lead to an incorrect ranking of overall similarity and even to a (nearly) empty set. A practical solution is to use a threshold for each view, such that a variety with a view having a similarity below the threshold, can never be regarded as a similar variety, even if all other views are (nearly) identical. This would extend the search considerably and probably reduce the advantages of the SR-tree unacceptably.

Currently, the system is being optimised for rose flowers, enhancing the feature-extraction and matching process using the SR-tree. In the future the system will be extended to other flower types and will provide a direct link to a relational database containing the variety information corresponding with the image(-object).

Since it uses a platform independent interface (Java), the image library is programmed in ANSI-C and the connection with the relational administrative database is only by a simple key (the variety number), FLORES is easily transportable to any platform and can be linked to nearly every database in any country. Therefore it offers possibilities as a standard tool in variety testing for finding the most similar variety using pictures. The final goal is to serve as a digital reference collection. It will be developed further and other member states are kindly invited to discuss the system.

References

1. Flickner, M, et al. Query by image content: the QBIC-system. IEEE Computer (1995) 23-32
2. Gong, Y. Intelligent image databases. Toward advanced image retrieval. Kluwer Academic Publishers. (1998).
3. Gupta, A., Jain, R. Visual Information Retrieval. Comm. of the ACM 40(5) (1997) 70-79.
4. Katayama, N. Satoh, S. The SR-tree: an index structure for high-dimensional nearest neighbor queries. Proceedings of the ACM SIGMOD, Tucson Arizona (1997).
5. Scil-Image user manual. Version 1.4. TNO Institute of Applied Physics, Delft, NL (1997).
6. Van der Heijden, G.W.A.M., G. Polder & J.W. van Eck. FLORES: a JAVA based image database for ornamentals. VISUAL'99. Amsterdam 2-4 June 1999.

[End of document]