An efficiency comparison of two content-based image retrieval systems, GIFT and PicSOM

Mika Rummukainen, Jorma Laaksonen, and Markus Koskela

Laboratory of Computer and Information Science,
Helsinki University of Technology,
P.O.BOX 5400, 02015 HUT, Finland
{mika.rummukainen,jorma.laaksonen,markus.koskela}@hut.fi

Abstract. Content-based image retrieval (CBIR) addresses the problem of assisting a user to retrieve images from unannotated databases, based on features that can be automatically derived from the images. Today, there exists several CBIR systems based on different methods. Only few attemps to benchmark these have been made, although the usefulness of benchmarking is undeniable in the development of different algorithms. In this paper we publish our benchmarking results of two CBIR systems with different implementation methods. The CBIR systems in question are GIFT (University of Geneva) and PicSOM (Helsinki University of Technology). The results clearly show that our PicSOM system, which we earlier have not been able to benchmark against other CBIR systems, comes off well in the comparison. Also, the results indicate that tests based on a single ground truth class are not enough for fair system comparisons.

1 Introduction

There has been a growing need for efficient image search engines in the WWW and other domains during the past few years. A number of content-based image retrieval (CBIR) systems have emerged but their performance has developed much slower than the performance of text search engines, such as Google or Altavista.

The human character is competitive by nature and thus a competition is a good way to quickly improve the quality of existing systems. The Benchathlon project (*http://www.benchathlon.net*) and IAPR's Technical Committee 12 (*http://sci.vu.edu.au/˜clement/TC-12/benchmark.htm*) both aim to hold an open competition for CBIR researchers where everyone can attend. In the Benchathlon project it has been planned to use the MRML communication language [1] (*http://www.mrml.net*) developed by the Computer Vision Group of the University of Geneva. The purpose of MRML is to divide a CBIR system into separate client and server parts and to create a standard method for communicating with different CBIR servers.

The PicSOM CBIR system [2], developed in the Laboratory of Computer and Information Science of the Helsinki University of Technology, can now communicate using MRML [3]. The PicSOM system also includes a benchmarking tool, which can be used for testing other MRML-based CBIR systems as well. Since the only publicly available MRML-based CBIR system was GIFT [4], we were able to run benchmarks between two systems. The results of our experiments are shown in this paper.

2 PicSOM

The PicSOM CBIR system is a framework for research on methods for content-based image retrieval (see [2] for a recent review; the PicSOM home page is at *http://www.cis.hut.fi/picsom*). The system is based on using several parallel Self-Organizing Maps (SOMs) [5] trained with separate statistical feature data. The SOM is an artificial neural network algorithm which defines an elastic, topology-preserving grid of points fitted to the high-dimensional input space. It attempts to represent all the available observations with an optimal accuracy by using a restricted set of models.

As a result of using multiple SOMs, the PicSOM system inherently uses multiple features for image retrieval and generally benefits from using all available features as it automatically neglects those working poorly. Features are usually comprised of statistical low-level visual data such as the MPEG-7 [6] content descriptors used in this work.

To reduce the complexity of training large SOMs, a special form of the algorithm, the Tree Structured Self-Organizing Map (TS-SOM) [7] is used. After training the SOMs with the TS-SOM algorithm, the map units are connected with the images of the database. This is done by locating the best-matching map unit (BMU) for each image. Furthermore, among the images sharing a common BMU, the best-matching one is used as a visual label for that unit.

2.1 Relevance Feedback with Self-Organizing Maps

The relevance feedback mechanism of PicSOM, implemented by using several parallel SOMs, is a crucial element of the retrieval engine [8]. The basic assumption is that images which are similar according to a specific visual feature are located near each other on the corresponding SOM surface. Therefore, we are motivated to spread the relevance information given by the user also to the neighboring map units of the user-rated images. This is done as follows. All relevant images are first given equal positive weight inversely proportional to the number of relevant images. Likewise, nonrelevant images receive negative weights that are inversely proportional to their total number. The overall sum of these relevance values is thus zero. For each SOM, the values are then mapped from the images to the corresponding BMUs where they are summed.

The resulting sparse value fields on the SOM surfaces are low-pass filtered to produce qualification values for each SOM unit and its associated images. The low-pass filtering of the sparse value fields can be performed by convolving the field with a tapered window function. The exact shape of the window function is not significant, e.g. triangular or Gaussian windows can be used, but the length

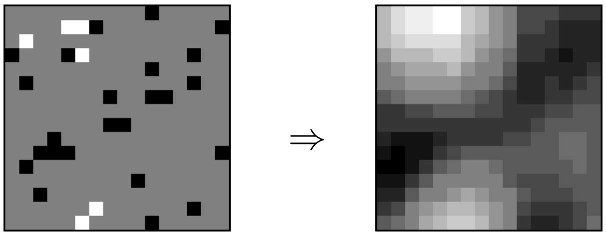

Fig. 1. An example of how a SOM surface is convolved with a low-pass filter. On the left, images selected and rejected by the user are shown with white and black marks, respectively. On the right, the convolution result, where relevance information is spread around the centers.

of the window is important for both retrieval performance and computational complexity [9]. Figure 1 illustrates how the positive and negative responses, displayed with white and black map units, respectively, are first mapped on a SOM surface and how the responses are expanded in the convolution. The total qualification value for each image is finally obtained by summing the corresponding responses on all used SOMs. Content descriptors that fail to coincide with the user's conceptions mix positive and negative user responses in nearby map units. Therefore, they produce lower qualification values than those descriptors that match the user's expectations and impression of image similarity. As a consequence, the different content descriptors and the SOMs formed from them do not need explicit weighting as the system automatically takes care of weighting their opinions.

3 GNU Image Finding Tool – GIFT

GIFT or GNU Image Finding Tool [4] is a free CBIR system released under GNU license. It was developed by the Computer Vision Group of University of Geneva (*http://vision.unige.ch*) and it is the first content-based image retrieval system that uses the MRML language [1] for its communications. The first version of GIFT was released in 1999 and was known as Viper [10]. GIFT is, however, still under development, and the latest released version is 0.1.9.

The purpose of the MRML language (*http://www.mrml.net*) is to separate the client part from the server part of the CBIR system and to create a standard method for communicating with different CBIR servers [11]. Thus, MRML sets the basis for easy benchmarking of CBIR systems. MRML is based on the XML language and it was developed by the authors of the GIFT system.

GIFT uses QBPE (Query by Pictorial Example) with user-relevance feedback. An MRML-based client is needed for connecting to GIFT and such a client is for example Charmer [12] (previously SnakeCharmer) also released under GNU license.

3.1 GIFT features

GIFT utilizes techniques common from textual information retrieval and uses a very large collection of binary-valued features. The number of features can be more than 80 000 and they are both local and global, simple color and texture features [10]. All the features are considered either present or not present in each of the images and each image typically has around thousand present features [11]. GIFT thus has a variable-length list of discrete features for every image.

3.2 Inverted file

The GIFT system includes a tool which is used for indexing new image collections. GIFT uses an inverted-file technique for its database indexing system. In the inverted-file database all existing features are listed [10, 4]. For each feature there is a list of all the images that contain this feature and the frequency of its occurrence in the collection. This inverted-file database system has been used and developed in text retrieval systems so the images can be accessed very efficiently [13].

3.3 GIFT algorithms

GIFT offers two built-in algorithms: Separate Normalization and Classical Inverse Document Frequency (CIDF). Both of them have been used in a black-box fashion in our experiments. The CIDF algorithm uses the following methods for weighting features:

$$\mathit{feature\ relevance}_j = \frac{1}{N}\sum_{i=1}^{N}(\mathit{tf}_{ij}R_i)\log^2(\frac{1}{\mathit{cf}_i}) \tag{1}$$

$$\mathit{image\ score}_{kq} = \sum_{j} \mathit{tf}_{kj}\, \mathit{feature\ relevance}_j \ \ , \tag{2}$$

where tf_{ij} is the term frequency of a feature in either a query or a result image, cf_i the collection frequency of a feature, q is a query with $i = 1, 2, .., N$ input images, k a result image, j the index of a feature and R the user-relevance of a query image between [-1,1] [11].

3.4 Noted problems with GIFT

When we tried to perform the tests with GIFT we encountered a few problems. First was due to the high number of images (59 995) in our Corel test database, which somehow affected GIFT in such a way that GIFT was not able to create its final inverted-file database in a Pentium-based Linux system. First we assumed the problem occured because of the 2 GB memory limitation of the 32-bit system, but afterwards it seems that this was not the case. However, we had to port GIFT to Compaq GS160 Tru64 UNIX AlphaServer (equipped with 16 Alpha 1001 MHz CPUs and 16 GB of system memory) to complete the database indexing.

The other problems were the relatively low speed of GIFT both in the indexing and especially in the testing phases. The feature extraction for all the images in the database took about three days (70 hours) to complete. Thus, the time needed for the feature extraction of a single image was approximately 4 seconds. After the feature extraction had completed, the indexing phase including the creation of the inverted-file database took approximately 2 days (40 hours).

In the performed experiments the largest ground-truth class held 1115 images, and the most of the classes contained between 500 and 800 images. In the performance evaluation of GIFT, it managed to go through only approximately 20–25 queries per day, which made the running time for the most of the experiments 20–30 days and for the largest class more than 40 days. This is a really long time compared to the time required by PicSOM, which managed even the largest class within five hours.

3.5 PicSOM and GIFT differences

Between the implementations of PicSOM and GIFT there are several differences. The most important difference is found in the handling of the user-relevant images inside a single query. After the user has selected relevant images and asks for a new set of similar images, GIFT returns also already selected images and even those that have been marked as nonrelevant. Instead, PicSOM keeps a list of shown images and never returns these back to the user in the same query.

One might question the purpose of GIFT performing in such a way, since the prerequisite for reasonable results from performance measurements is that no same image is returned twice. Therefore, in the performed tests we had to change the number of new images asked from GIFT to the number of seen images plus the size of the query window so that GIFT certainly would return enough new images.

In GIFT, the total size of feature-files, 4.1 GB, and also the resulting size of the database, 780 MB, are both quite large compared to the 2.5GB size of the image database itself. With the URL to feature conversion file these take a total of 5 GB of system space which is twice the size of the original database. For PicSOM to store all features and indices it takes only about 1.1 GB in total.

4 Experiment settings

We used a database of 59 995 images from the Corel Gallery 1 000 000 product. The size of each image is either 384×256 or 256×384 pixels. The images have been grouped by Corel in thematic groups and also keywords are available. However, we have found these image groups rather inconsistent with the keywords. Therefore, we have created six manually-picked ground truth image sets with tighter membership criteria for experimenting with the PicSOM system [2, 14]. All the image sets were gathered by a single subject. The used sets and membership criteria were:

- **faces**, 1115 images (*a priori* probability 1.85%), where the main target of the image is a human head which has both eyes visible and the head has to fill at least 1/9 of the image area.
- **cars**, 864 images (1.44%), where the main target of the image is a car, and at least one side of the car has to be completely shown in the image and its body to fill at least 1/9 of the image area.
- **sunsets**, 663 images (1.11%), where the image contains a sunset with the sun clearly visible in the image.
- **horses**, 486 images (0.81%), where the main target of the image is one or more horses, shown completely in the image.
- **planes**, 292 images (0.49%), where all airplane images have been accepted.
- **traffic signs**, 123 images (0.21%), where the main target of the image is one or more official traffic signs, so commercial and other signs were rejected.

In PicSOM, we used a subset of MPEG-7 [6] descriptors as visual features. The used descriptors were *Scalable Color* (256), *Dominant Color* (6), *Color Structure* (256), *Color Layout* (12), *Edge Histogram* (80), *Homogeneous Texture* (62), and *Region Shape* (35). The dimensionalities of the descriptors are listed in parentheses. The MPEG-7 standard defines not only the descriptors but also special metrics that can be used with the descriptors when calculating the similarity between images. However, we use Euclidean metrics in comparing the descriptors because the training of the SOMs is based on minimizing a square-form error criterium. Only in the case of *Dominant Color* descriptor this has necessitated a slight modification in the use of the descriptor. MPEG-7's *Dominant Color* descriptor is variable-sized, i.e., the length of the descriptor varies depending on the count of dominant colors found. Because this could not be fit in the PicSOM system, we used only two most dominant colors or duplicated the most dominant color if only one was found.

For each feature, we trained a four-level TS-SOM with level sizes 4×4, 16×16, 64×64, and 256×256 units. The size of the bottommost TS-SOM level is selected so that it roughly equals the size of the database. In the training of the lower SOM levels, the search for the BMU has been restricted to the 10×10-sized neuron area below the BMU on the above level. Every image has been used 100 times for training each of the TS-SOM levels. Because the queries in the used experiment setting are always started with an image that belongs to the studied image class, the retrieval can be initiated in the neighborhoods of the reference image on the bottommost SOM levels (256×256=65536 map units) as they provide the most detailed resolution. After the training phase, the upper TS-SOM hierarchy is thus neglected. In spreading the responses of the sparse value fields, triangular windows of 8 map units in length were used.

In GIFT the algorithms used were both Separate Normalization and Classical Inverse Document Frequency (CIDF). Both algorithms were used with their default configurations, and the usage of GIFT was performed in a black-box style.

If the size of the database, N, is large enough, we can assume that there is an upper limit N_T of images ($N_T \ll N$) the user is willing to browse during a single retrieval session. In our test setting, each image in the studied class is given to the system one at a time as the initial reference image for category search. The system should then return images belonging to the same class, resulting in a leave-one-out type testing of the class. The system was set to return 20 images at each round, and with 20 rounds per query the total number of seen images was $N_T = 400$ images, i.e. 0.67% of the database size.

We chose to show the evolution of *precision* $\mathcal{P}(n)$ as a function of *recall* $\mathcal{R}(n)$ during the image retrieval process [15]. Precision and recall are intuitive performance measures that suite also for the case of non-exhaustive browsing. First, the initial values of $\mathcal{P}(n)$ display the initial accuracy of the system. Then, the intermediate values show how the relevance feedback mechanism is able to adapt to the class, and the final value $\mathcal{R}(N_T)$ – as well as $\mathcal{P}(N_T)$ – reflects the total number of relevant images found that far. With an effective relevance feedback mechanism, it is expected that $\mathcal{P}(n)$ first increases and then turns to decrease when a notable fraction of relevant images have already been shown.

In our experiments, we have normalized the precision value by dividing it with the *a priori* probability $\rho_{\mathcal{C}}$ of the class and call it therefore *relative precision* [2]. This makes the comparison of the recall–precision curves of different image classes somewhat commensurable and more convenient because relative precision values relate to the relative advantage the CBIR system produces over random browsing.

5 Results

Figure 2 displays the results of our experiments. Each of the six subfigures shows the recall–relative precision curves for one particular image class. Every subfigure contains three curves: the PicSOM algorithm, GIFT with the CIDF algorithm, and GIFT with the Separate Normalization algorithm.

The curves show that in three classes (faces, traffic signs, sunsets) PicSOM outperforms both GIFT algorithms clearly. In the faces class the recall values of the GIFT algorithms end at surprisingly low level. In the initial phase of the sunsets class the Separate Normalization algorithm is first better than PicSOM which then adapts better to the class and exceeds GIFT in precision.

In the remaining three classes (cars, planes, horses) GIFT performs better than PicSOM. Only in the last phase (after adaptation) of the planes class PicSOM catches up the Separate Normalization algorithm.

Between the two GIFT algorithms there is no clear winner, either. While the CIDF algorithm manages better in the traffic signs, planes and sunsets classes, the Separate Normalization algorithm is better in the other (cars, horses, faces) classes. In general, one might state that in most of the cases the precision of the CIDF algorithm improves during the iterative query, whereas the precision of the Separate Normalization algorithm either remains the same or begins to decline sooner.

Of the evaluated three algorithms, PicSOM outranks the others in three classes, Separate Normalization in two classes and CIDF in one class. Thus, in

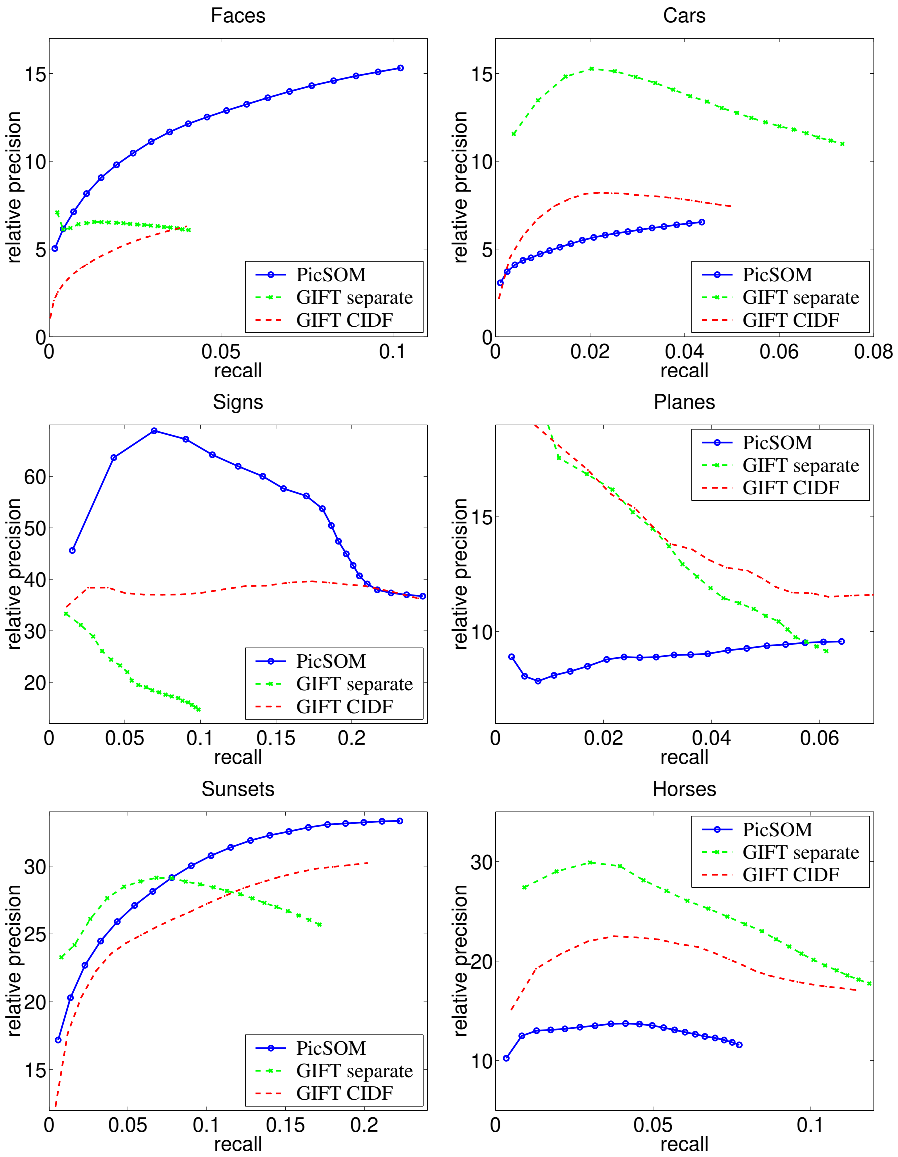

Fig. 2. Recall–relative precision curves for the three algorithms and six image classes.

the light of these results, there is no clear winner between the three compared algorithms, but we can at least tentatively say that the performance of PicSOM is on the same level as that of GIFT's.

The image classes can also be analyzed by the performance of the best-performing algorithm. In the faces and sunsets classes the precision of retrieval is increasing all the time, whereas in the other classes it decreases. In the traffic signs and cars classes the decrease of precision starts only after some query rounds, while in the horses and planes classes the decrease starts immediately.

6 Conclusions and future plans

In this paper, we have presented a comparison of three content-based image retrieval techniques from two independent CBIR systems. Based on our results from experiments with six manually-selected image classes, we find both PicSOM and GIFT very competitive CBIR systems. Although PicSOM outperforms both of the two GIFT algorithms in three out of the six classes, there is still work to do to improve the efficiency of PicSOM in the other three classes. The public availability of the GIFT system has thus made it possible for us to run this kind of a benchmark and to find out the weaknesses of our own system.

The results of our experiments also show how important it is to include enough variety for comparisons in benchmarking. If the comparison results had been based on a single class only, they would not have revealed all the information we now have. We even could have declared any one of the algorithms as a clear winner with a purposeful selection of the image class. However, since our results are, in fact, based on a single type of comparison only, we do not claim to have thoroughly benchmarked the two CBIR systems. Instead, we note that for a good benchmark, several different types of tests are needed, which all should measure the efficiencies of the systems from different perspectives.

Actually, we would have wanted to run the experiments with 50 rounds per query so that instead of $N_T = 400$ the total number of seen images would have been $N_T = 1000$ as in earlier experiments with PicSOM. However, running GIFT turned out be so time consuming that we had to settle for a lower number of rounds per query. In spite of that, the results distinguish the algorithms adequately and a larger number of rounds per query could only have given more distinctive results in the cases where the performances of two algorithms were close to each other.

We have recently been experimenting with a new modification of the PicSOM algorithm. This novel development incorporates the use of automatically segmented images in the system. Our very preliminary results have shown that the precision of the system can be considerably increased by this method, especially in the cases where the original PicSOM method is performing poorly. It will be very exciting to run a comparison between this approach and the two GIFT algorithms.

Acknowledgment

This work was supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *New information processing principles*, the latter being part of the Finnish Centre of Excellence Programme.

References

1. Müller, W., Pecenovic, Z., Müller, H., Marchand-Maillet, S., Pun, T., Squire, D.M., Vries, A.P.D., Giess, C.: MRML: An extensible communication protocol for interoperability and benchmarking of multimedia information retrieval systems. In: Proceedings of SPIE Photonics East – Voice, Video, and Data Communications, Boston, MA, USA (2000)
2. Laaksonen, J., Koskela, M., Oja, E.: PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing **13** (2002) 841–853
3. Rummukainen, M.: Implementing multimedia retrieval markup language for image retrieval systems' comparison. Master's thesis, Helsinki University of Technology, Otaniemi, Espoo, Finland (2003)
4. Müller, H., David McG. Squire, W.M., Pun., T.: Efficient access methods for content-based image retrieval with inverted files. In: Proceedings of Multimedia Storage and Archiving Systems IV (VV02), Boston, MA, USA (1999)
5. Kohonen, T.: Self-Organizing Maps. Third edn. Volume 30 of Springer Series in Information Sciences. Springer-Verlag (2001)
6. MPEG: MPEG-7 overview (version 8.0) (2002) ISO/IEC JTC1/SC29/WG11.
7. Koikkalainen, P., Oja, E.: Self-organizing hierarchical feature maps. In: Proceedings of International Joint Conference on Neural Networks. Volume II., San Diego, CA (1990) 279–284
8. Laaksonen, J., Koskela, M., Laakso, S., Oja, E.: Self-organizing maps as a relevance feedback technique in content-based image retrieval. Pattern Analysis & Applications **4** (2001) 140–152
9. Koskela, M., Laaksonen, J., Oja, E.: Implementing relevance feedback as convolutions of local neighborhoods on self-organizing maps. In: Proceedings of International Conference on Artificial Neural Networks, Madrid, Spain (2002) 981–986
10. Squire, D.M., Müller, W., Müller, H., Pun, T.: Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. Technical Report 98.04, Computer Vision Group, Computing Centre, University of Geneva, rue Gnral Dufour, 24, CH-1211 Genve, Switzerland (1998)
11. Müller, H., Müller, W., Squire, D.M., Pecenovic, Z., Marchand-Maillet, S., Pun, T.: An open framework for distributed multimedia retrieval. Technical Report 00.03, Computer Vision Group, Computing Group, University of Geneva, rue Gnral Dufour, 24, CH-1211 Genve, Switzerland (2000)
12. Pecenovic, Z.: (Charmer CBIR client, `http://viper.unige.ch/demo/demo.html`) Visited 27.03.03.
13. Squire, D., Müller, W., Müller, H., Pun, T.: Content-based query of image databases: inspirations from text retrieval. Pattern Recognition Letters **21** (2000) 1193–1198
14. Koskela, M., Laaksonen, J., Oja, E.: MPEG-7 descriptors in content-based image retrieval with PicSOM system. In: Proceedings of 5th International Conference on Visual Information System, HsinChu, Taiwan (2002) 247–258
15. Müller, H., Müller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Performance evaluation in content-based image retrieval: overview and proposals. Pattern Recognition Letters **22** (2001) 593–601