

An Image Retrieval System Using Multispectral Random Field Models, Color, and Geometric Features

Orlando J. Hernandez
Electrical and Computer Engineering
The College of New Jersey
Ewing, New Jersey 08628-0718
hernande@tcnj.edu

Alireza Khotanzad
Electrical Engineering
Southern Methodist University
Dallas, Texas 75275-0338
kha@enr.smu.edu

Abstract

This paper describes a novel color texture-based image retrieval system for the query of an image database to find similar images to a target image. The retrieval process involves segmenting the image into regions of uniform color texture using an unsupervised histogram clustering approach that utilizes the combination of Multispectral Simultaneous Auto Regressive (MSAR) and color features. The color texture content, location, area and shape of the segmented regions are used to develop similarity measures describing the closeness of a query image to database images. These attributes are derived from the maximum fitting square and best fitting ellipse to each of the segmented regions. The proposed similarity measure combines all these attributes to rank the closeness of the images. The performance of the system is tested on two databases containing synthetic mosaics of natural textures and natural scenes, respectively.

1. Introduction

An ever-increasing usage of digital images and large volume image databases gives rise to the need for organizing them according to their content so they can be retrieved easily. Retrieval of image data based on pictorial content queries is an interesting and challenging problem actively worked on by the research community [1]. With the growth of multimedia computing and the spread of the Internet, more and more people have access to large databases and would have applications for such retrieval systems.

Texture information has been used in the past for browsing and retrieval of imagery, however the previously proposed approaches have either considered only gray level textures or pixels based color content and

not color texture [2]. The utilized segmentation algorithm in some approaches has also not been completely unsupervised [3].

2. Unsupervised segmentation

The first step in the proposed retrieval process is to segment the query image into regions of uniform color texture. It should be noted that since the ultimate goal is to retrieve images similar to the one presented to the system, it is necessary to find dominant regions of texture in the image but locating very exact boundaries of such regions is not as critical.

The segmentation algorithm used in this work relies on scanning the image with a sliding window and extracting "Color Content Color Texture" (C³T) features from each window [4], [5]. These features are then clustered using an unsupervised histogram-based algorithm. Mapping the identified clusters back into the image domain results in the desired segmentation.

The performance of the segmentation algorithm and the associated features is illustrated in **Fig. 1**. **Fig. 1** shows the segmentation results for several natural scene images. These natural scene images are available in [7]. Below each image the segmentation result is presented in the form of a gray-level image with pixels belonging to the same texture having the same gray level. In the next row, the boundaries of the segmented regions are shown as superimposed white lines. At the top of the figures, the size of the optimal window found by the algorithm is shown. It is observed that the algorithm performs quite well and is capable of localizing uniform color textures in each image.

In **Fig. 1**, we also compare the results of our approach with the image segmentation results achieved using the JSEG method described in [2]. The JSEG results were obtained from applying the images to the programs made

available by the JSEG authors at the Internet site <http://maya.ece.ucsb.edu/JSEG/>. The obtained region boundaries are superimposed on the original images. The

JSEG results are displayed in the last rows of Fig. 1. It can be seen that our segmentation results have a better match with perceptual boundaries in the images.

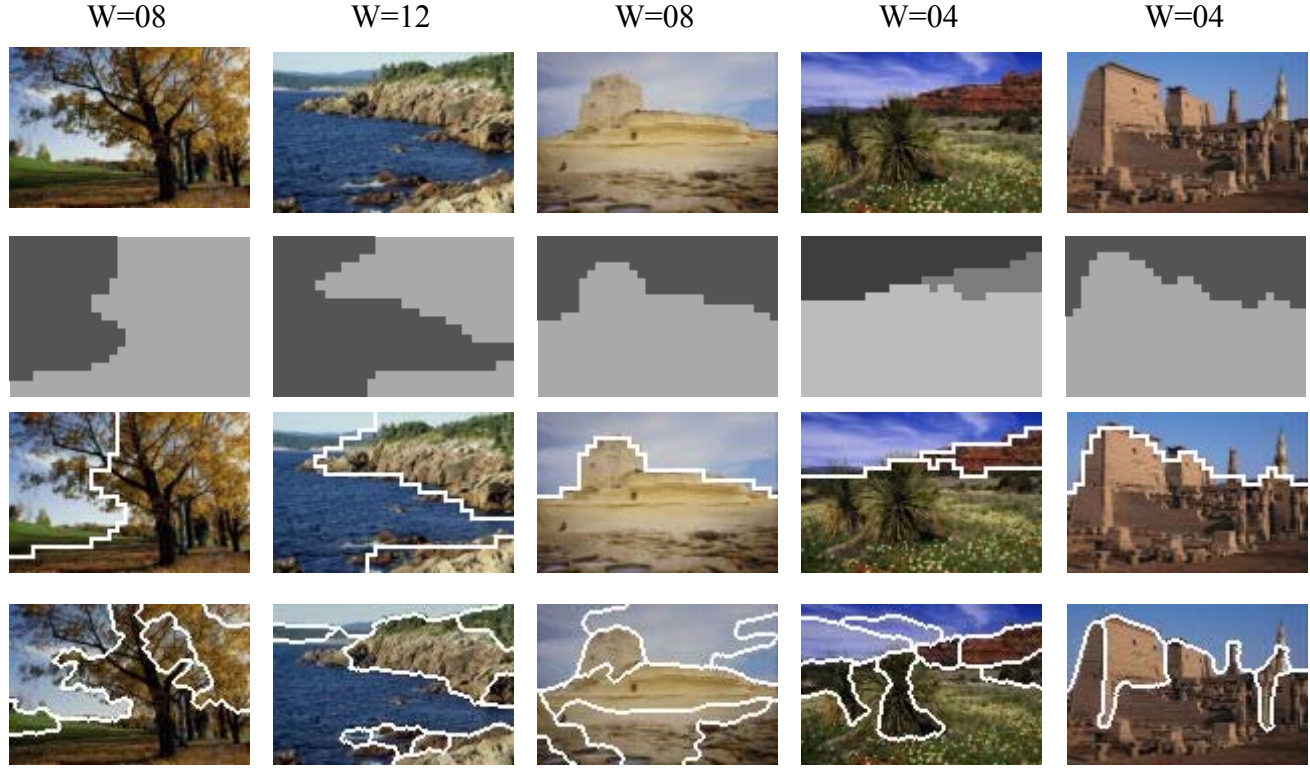


Fig. 1. Segmentation results for eight natural scene images, 1st row: Original image, 2nd row: Segmentation results, 3rd row: Texture boundaries corresponding to segmentation results, 4th row: Segmentation using JSEG method

3. Attributes used in similarity metrics

Once the images are segmented into regions of distinct color texture, similarity metrics need to be developed to measure how close two images are with respect to their color texture content. The attributes used for similarity computation are described in this section.

The foremost attribute is the type of color texture of each segmented region. To characterize this texture, the largest square that could be fitted to the region is found. This square will be referred to as the “Maximum Fitting Square” (MFS). The C³T features are then extracted from the MFS and used to characterize the color texture of the entire segmented region. The reason for using the MFS is that the shape of the segmented region could be irregular which will not be suitable for the MSAR model computation.

Several shape-related parameters are then extracted from the entire region. These are:

1. Centroid location, which is the mean of the row and column positions of the pixels contained in the region,

i.e. (\bar{x}, \bar{y}) . The centroid location conveys information about the overall position of the region.

2. Area (A) which is the total number of pixels contained by the region.
3. Overall shape as measured by the moments that define the best fitting ellipse to a region, i.e. the largest ellipse that could be fitted to the region. Such an ellipse could be found by computing three geometrical moments, μ_{xx} , μ_{yy} , and μ_{xy} where:

$$\mu_{xx} = \sum_{\text{OVER ROWS}} \sum_{\text{OVER COLUMNS}} \frac{(x - \bar{x})^2}{A}$$

$$\mu_{yy} = \sum_{\text{OVER ROWS}} \sum_{\text{OVER COLUMNS}} \frac{(y - \bar{y})^2}{A}$$

$$\mu_{xy} = \mu_{yx} = \sum_{\text{OVER ROWS}} \sum_{\text{OVER COLUMNS}} \frac{(x - \bar{x})(y - \bar{y})}{A}$$

These three shape-based measures along with the C^3T features extracted from the MFS are used in the similarity metric developed for the retrieval process.

Fig. 2 illustrates this process for three natural scene images. In the figure, the original images are shown in the

first column followed by the computed MFS' superimposed on the segmented regions. The centroid locations, and the best fitting ellipses in images are shown in columns three and four, respectively.

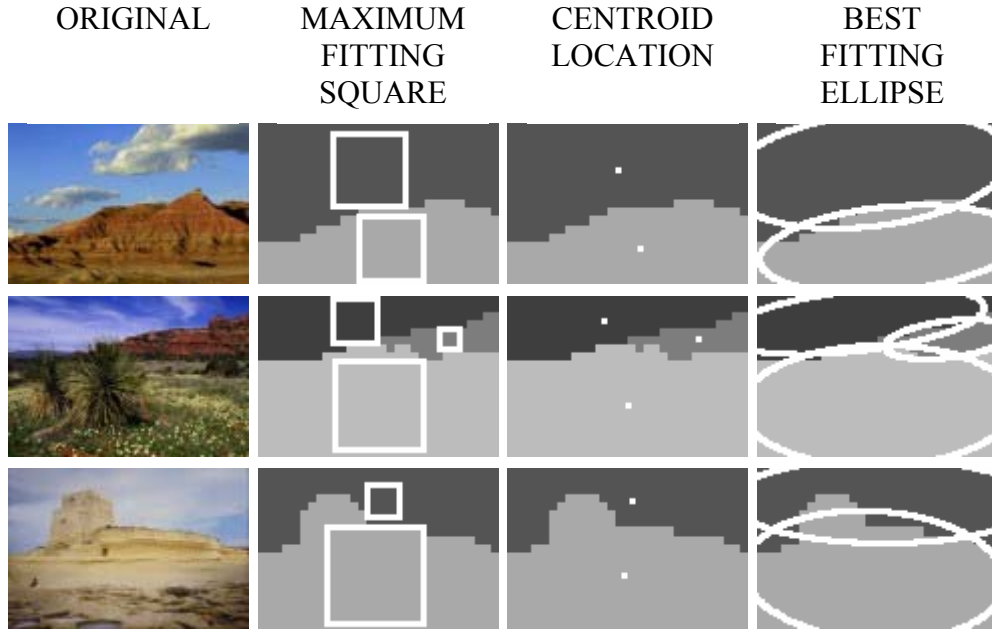


Fig. 2. Examples of texture and shape attribute computation for natural scene images

4. Search space reduction

The next step after extracting the texture and shape attributes is to reduce the number of database images that are to be considered in the retrieval process. This is done by finding those images that contain similar textures to that of the query image. This task is carried out by using the C^3T features extracted from the MFS. The C^3T features of the database images are first clustered using the clustering algorithm described earlier resulting in M clusters, C_i , $i = 1, 2, \dots, M$. Then, the C^3T features of each of the segmented regions of the query image are considered in this clustered 22-dimensional space. Let's denote the C^3T features of the k th segmented region of the query image as f_k . For each f_k , the closest cluster center $C_j(f_k)$ is found and all the database images that are associated with C_j cluster are tagged as images that need to be considered in the search process. After all f_k 's are considered, all the database images that are not tagged are removed from further consideration resulting in a reduced search space. In other words, only those database images that potentially have one or more similar color texture to those of the query image are retained. Note that this process also identifies the likely number of common

textures between the query image and each of the retained database images.

5. Region association

In the next phase, the retained database images are considered one at a time and an association is established between each region of the query image and one of the regions of the considered database image. This process is illustrated by an example.

Let's assume that the query image contains three segmented regions, Q_1 , Q_2 , and Q_3 and a database image has two regions denoted by R_1 and R_2 as shown in **Fig. 3**.

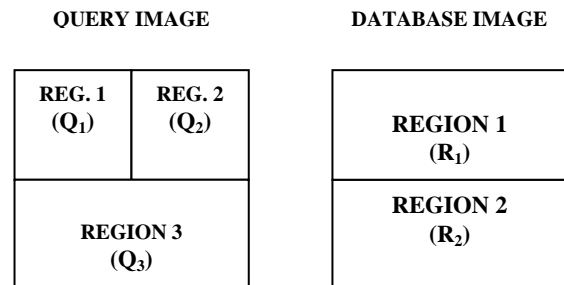


Fig. 3. An example of the Region Association process

The Euclidean distance between C^3T features of each possible Q_i, R_j pair ($d(Q_i, R_j)$) is computed resulting in the following six distances:

$$\{d(Q_1, R_1), d(Q_1, R_2), d(Q_2, R_1), d(Q_2, R_2), d(Q_3, R_1), d(Q_3, R_2)\}$$

These six distances are then sorted from smallest to largest. Let's assume that this sorting results in the following order:

$$\{d(Q_3, R_1), d(Q_2, R_1), d(Q_3, R_2), d(Q_1, R_2), d(Q_1, R_1), d(Q_2, R_2)\}$$

An association between each Q_i and R_j is established by considering this sorted list and making the association according to the first occurrence of each Q_i in the list. In this example, the following associations are made:

$$Q_3 \leftrightarrow R_1 \quad Q_2 \leftrightarrow R_1 \quad Q_1 \leftrightarrow R_2$$

Upon completion of this step, the best potential match between each region of the query image and one of the regions of each of the database images is established.

6. Similarity computation

The similarity between the query image and a database image is computed using the region associations established in the previous phase. In this stage, the shape-based attributes are utilized to arrive at a final similarity measure. The Euclidean distances between the position, area, and shape attributes of each associated pair are computed. Continuing with the example from the previous section, the computed distances will be:

$$Q_3 \text{ associated with } R_1 : d(P_{Q_3}, P_{R_1}), d(A_{Q_3}, A_{R_1}),$$

$$d(S_{Q_3}, S_{R_1})$$

$$Q_2 \text{ associated with } R_1 : d(P_{Q_2}, P_{R_1}), d(A_{Q_2}, A_{R_1}),$$

$$d(S_{Q_2}, S_{R_1})$$

$$Q_1 \text{ associated with } R_2 : d(P_{Q_1}, P_{R_2}), d(A_{Q_1}, A_{R_2}),$$

$$d(S_{Q_1}, S_{R_2})$$

where:

$$d(P_{Q_i}, P_{R_j}) = \text{Euclidean distance between centroid}$$

locations of Q_i and R_j

$$d(A_{Q_i}, A_{R_j}) = \text{Euclidean distance between areas of } Q_i$$

and R_j

$$d(S_{Q_i}, S_{R_j}) = \text{Euclidean distance between 3-}$$

dimensional vector of $(\mu_{xx}, \mu_{yy}, \mu_{xy})$ of Q_i and R_j

Next, the position, area, and shape distances for all Q_i 's are added together to form total position, area, and shape difference measures denoted by PD, AD, and SD, respectively.

$$PD = d(P_{Q_3}, P_{R_1}) + d(P_{Q_2}, P_{R_1}) + d(P_{Q_1}, P_{R_2})$$

$$AD = d(A_{Q_3}, A_{R_1}) + d(A_{Q_2}, A_{R_1}) + d(A_{Q_1}, A_{R_2})$$

$$SD = d(S_{Q_3}, S_{R_1}) + d(S_{Q_2}, S_{R_1}) + d(S_{Q_1}, S_{R_2})$$

At this point a total texture difference measure based on the Euclidean distance of the C^3T features is also computed and denoted as TD.

$$TD = d(C^3T_{Q_3}, C^3T_{R_1}) + d(C^3T_{Q_2}, C^3T_{R_1}) +$$

$$d(C^3T_{Q_1}, C^3T_{R_2})$$

Finally a composite distance measure used as the overall similarity measure between the two images is computed. This measure denoted as S is:

$$S = \sqrt{PD^2 + AD^2 + SD^2 + TD^2}$$

7. Retrieval

The S measure reflects the degree of similarity between the query image and the considered database image. As such, the retrieval process involves sorting all the S measures computed between the query image and each of the database images in the reduced search space. The database images ordered from smallest to largest S will be the most similar to least similar to the query image, respectively.

The flow chart in **Fig. 4** describes how the overall system works. First, there is an off line process to characterize the database. In this process, each image in an image database is segmented, and characterized. This means that for each region in the segmented image, the area, centroid, moments, and C^3T parameters are extracted, and included into the index ASCII file that each image file in the database has. After the database has been characterized, the on line process consists of characterizing the query image like it is done for the images in the database, determining the "Reduced Search Space", and calculating the similarity metric S for each of the images from the database that end up being members of the "Reduced Search Space". These images are ranked according to the similarity metric, and copied from the database directory to a directory that contains the retrieval results; this is the output directory from the query operation.

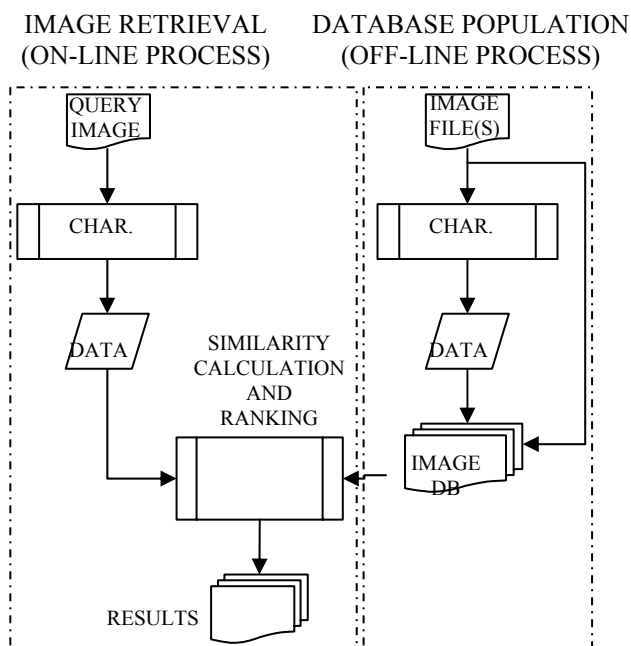
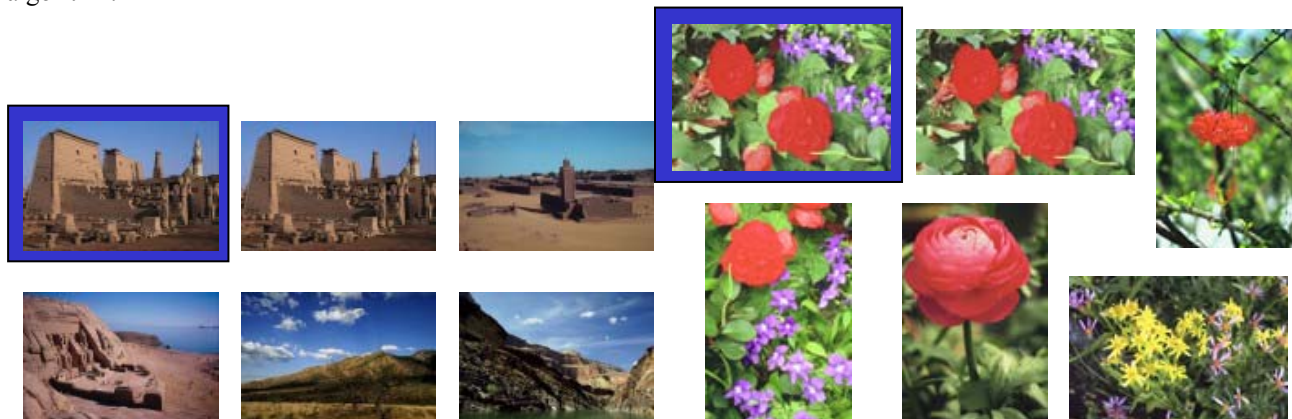


Fig. 4. Overall System Block Diagram

A series of experiments is carried out by taking an image from the considered database [6] [7] and using it as the query image. In all the experiments the database image that is the same as the query image is found as the most similar image to the query image.

In **Fig. 5**, examples of the performance for the Natural Scenes database are shown. The images with the blue border are the query image. The five most similar images retrieved from the database are shown in the order of similarity from left to right and top to bottom. Note that the exact match of the query image is ranked as the closest retrieved image. The other retrieved images all contain similar scenery as the query image. This is an important affirmation of the validity of the proposed algorithm.



It should be noted that the segmentation of database images can be done off-line. The online retrieval process involving the segmentation of the query image and computation and ranking of the corresponding S measures takes around 10 seconds on a 2.27 GHz Intel Pentium class machine with 512 MBytes of random access memory (RAM).

A performance evaluation exercise is conducted for this system. This consists of evaluating the “wall clock” run time for the database characterization and the query processes. This run time data is collected for the three different test databases, and on two different hardware compute platforms. The results are presented in **Table 1**. The hardware (**HW**) column describes the platform that is used to run the evaluations. Platform “A” refers to a 1.70 GHz Intel Pentium class machine with 128 Mbytes of RAM. Platform “B” refers to a 2.27 GHz Intel Pentium class machine with 512 Mbytes of RAM.

After studying the data in the table below, the following observations can be made:

- For a specific compute platform, the time it takes to characterize a database is linear with the number of images in the database.
- The time that it takes to characterize an image is linear with the area of the image.
- The time it takes to perform a query does not depend on the size of the database, but on the size and area of the query image.

Both the characterization of multiple images in a database, and the extraction of C³T features for the windows across the image being segmented for characterization, are independent processes. Therefore, this system lends itself to being implemented on parallel processing platforms, where the execution speedup achieved will be linear with the number of parallel processors employed.

Retrieved Image	S Value	Retrieved Image	S Value
Top Middle	0.0000	Top Middle	0.0000
Top Right	13.2759	Top Right	2.3322
Bottom Left	13.4045	Bottom Left	3.8578
Bottom Middle	13.4643	Bottom Middle	7.4531
Bottom Right	13.6548	Bottom Right	7.7316

Fig. 5. Examples of retrieval results for queries of the Natural Scenes database

Table 1. Performance Analysis Results.

PROC.	DB	IMAGES	SIZE	AREA	HW	TIME (min.)	PER IMAGE
OFF-LINE	TEXT. I	30	128 X 128	16,384	A	52	2
OFF-LINE	TEXT. I	30	128 X 128	16,384	B	25	0.841
OFF-LINE	TEXT. II	102	128 X 128	16,384	A	174	2
OFF-LINE	TEXT. II	102	128 X 128	16,384	B	82	0.804
OFF-LINE	SCENES	51	120 X 80	9,600	A	49	0.959
OFF-LINE	SCENES	51	120 X 80	9,600	B	21	0.407
QUERY	TEXT. I	30	128 X 128	16,384	A	2	0.055
QUERY	TEXT. I	30	128 X 128	16,384	B	0.529	0.003
QUERY	TEXT. II	102	128 X 128	16,384	A	2	0.021
QUERY	TEXT. II	102	128 X 128	16,384	B	0.705	0.007
QUERY	SCENES	51	120 X 80	9,600	A	0.822	0.016
QUERY	SCENES	51	120 X 80	9,600	B	0.062	0.007

8. Conclusions

In this work, a color texture-based approach to image retrieval from a large database is developed. Features derived from the Multispectral Autoregressive (MSAR) random field model and the RGB color space characterize the color texture content of the image. These features are used in conjunction with an unsupervised clustering-based segmentation algorithm to segment the images into regions of uniform color texture. Similarity measures based on the color texture content, shape, size, and position of the segmented regions are developed to measure the closeness of a query image to the database image. The effectiveness of the approach has been demonstrated using two different databases containing synthetic mosaics of natural textures and natural scenes.

9. References

[1] Y. Rui and T. S. Huang, Image Retrieval; Current Techniques, Promising Directions, and Open Issues, Journal of

Visual Communications and Image Representation, vol. 10, 1999, pp. 39-62.

[2] Y. Deng and B. S. Manjunath, Unsupervised Segmentation of Color-Texture Regions in Images and Video, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 8, 2001, pp. 800-810.

[3] J. R. Smith and S-F. Chang, Integrated Spatial and Feature Image Query, Multimedia Systems – ACM - © Springer-Verlag 1999, vol. 7, no. 2, 1999, pp. 129-140.

[4] O.J. Hernandez and A. Khotanzad, "Color Image Segmentation Using Multispectral Random Field Texture Model & Color Content Features", *Journal of Computer Science and Technology*, October 2004.

[5] O.J. Hernandez and A. Khotanzad, "Color Image Retrieval Using Multispectral Random Field Texture Model & Color Content Features", *Pattern Recognition Journal*, Volume 36, Issue 8, August 2003.

[6] Vision and Modeling Group, MIT Media Laboratory, Vision Texture (VisTex) database, <http://www-white.media.mit.edu/vismod/>, 1995.

[7] Corel Corporation, Professional Photos CD-ROM Sampler – SERIES 200000, 1994.