A World Wide Web Region-Based Image Search Engine

Kompatsiaris, Ioannis; Triantafyllou, Evangelia; Strintzis, Michael G.

A World Wide Web Region-Based Image Search Engine

I.Kompatsiaris, E.Triantafyllou, and M.G.Strintzis *

Informatics and Telematics Institute
1st km. Thermi - Panorama Road
57001 Thermi - Thessaloniki, Greece
Email: ikom@iti.gr

Abstract

In this paper the development of an intelligent image content-based search engine for the World Wide Web is presented. This system will offer a new form of media representation and access of content available in WWW. Information Web Crawlers continuously traverse the Internet and collect images that are subsequently indexed based on integrated feature vectors. As a basis for the indexing, the K-Means algorithm is used, modified so as to take into account the coherence of the regions. Based on the extracted regions, characteristic features are extracted using color, texture and shape/region boundary information. These features along with additional information such as the URL location and the date of index procedure are stored in a database. The user can access and search this indexed content through the Web with an advanced and user friendly interface. The output of the system is a set of links to the content available in the WWW, ranked according to their similarity to the image submitted by the user. Experimental results demonstrate the performance of the system.

1. Introduction

Thanks to the ubiquity of computer networks and the continuous increase in storage capacity, the amount of digital information that can be readily available and accessed by experts and lay-people alike, has increased dramatically and will keep on doing so at an accelerating pace. Information, and content as such, are rapidly becoming highly valuable commodities, the economic importance of which is beginning to rival that of more traditional resources. At the same time, there has been a tremendous growth of the Internet and all related services, especially of the World Wide Web (WWW). The Internet has become the most important source of information, content and knowledge. Ironically, there is a real big danger that this important and exciting development of digital information available over the Internet will fall victim to its own success. Part of the challenge stems from the sheer volume of data available, its ever-changing quality, and its wide range of formats. Indeed, the further elaboration of this evolution depends critically on the speed and reliability with which this information in all its available forms and formats can be retrieved and in that respect the size of the Internet has become an impediment rather than an asset, as it seriously hampers attempts to efficiently locate and collect relevant information.

In order to overcome this problem, search engines using text descriptors have been developed. A large number of catalogs and search engines index the plethora of documents on the World-Wide Web. For example, recent systems, such as Lycos, Alta Vista and Yahoo, index the documents by their textual content. These systems periodically scour the Web, record the text on each page and through processes of automated analysis and/or (semi-) automated classification, condense the Web into compact and searchable indexes. The user, by entering query terms and/or by selecting subjects, uses these search engines to more easily find the desired Web documents. Generally, the text-based Web search engines are evaluated on the basis of the size of the catalog, speed and effectiveness of search, and ease of use [1].

However, very few tools are currently available for searching for images and videos. This absence is particularly notable given the highly visual and graphical nature of the Web [2, 3]. Visual information is published both as embedded in Web documents and as stand-alone objects. The visual information takes the form of images, graphics, bitmaps, animations and videos. As with Web documents in general, the publication of visual information is highly volatile. New images and videos are added everyday and others are replaced or removed entirely. In order to
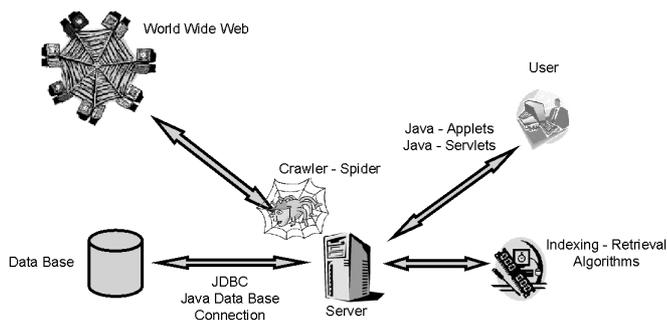
Figure 1. General System Architecture.

allow efficient search of the visual information, a highly efficient automated system is needed that regularly traverses the Web, detects visual information and processes it in such away to allow for efficient and effective search and retrieval.

In this paper the development of an intelligent image content-based search engine for the World Wide Web is presented. This system will offer a new form of media representation and access of content available in WWW. Information Web Crawlers continuously traverse the Internet and collect images that are subsequently indexed based on integrated feature vectors. These features along with additional information such as the URL location and the date of index procedure are stored in a database. The user can access and search this indexed content through the Web with an advanced and user friendly interface. The output of the system is a set of links to the content available in the WWW, ranked according to their similarity to the image submitted by the user.

The paper is organised as follows. In Section 2 an overview of the general system architecture is presented. In Section 3 a description of the Information Crawlers is given, while the indexing and retrieval algorithms are presented in Section 4. Experimental results are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. General System Architecture Overview

The overall system implementation is based on the use of the Java programming language. Java provides more efficient ways to develop network and especially WWW components and on the same time develop algorithms (indexing and retrieval) and communicate with databases. The overall system is split into two parts: (i) the off-line part and (ii) the on-line or user part.

In the off-line part, *Information Crawlers*, developed entirely in Java, continuously traverse the WWW, collect images and transfer them to the central *Server* for further processing (Fig. 1). Then the *image indexing* algorithms

process the image in order to extract descriptive features. The indexing procedure is based on a novel technique for the segmentation of images. As a basis for the indexing, the K-Means algorithm is used [4], modified so as to take into account the coherence of the regions. Based on the extracted regions, characteristic features are extracted using color, texture and shape/region boundary information. The algorithms for image processing were partially developed in C and linked with the rest of the system using Java Native Interface (JNI), and partially developed in Java. These features along with information of the images such as URL, date of process, size and a thumbnail are stored in the *database* using the Java Data Base Connection (JDBC) protocol. In our initial implementation, a MS-Access 2000 database was used, which supports the JDBC protocol and can also handle SQL queries. In this stage the full size initial image is discarded.

In the on-line part, a *user* connects to the system through a common Web Browser using the HTTP protocol. The user can then submit queries either by example images or by simple image information (size, date, initial location, etc). The query form is accessible to the user as a Java applet. The query then is processed by the server and the *retrieval* phase begins; the indexing procedure is repeated again for the submitted image and then the extracted features are matched against those stored in the database using an SQL query. The results containing the URL as well as the thumbnail of the similar images are transmitted to the user either using Java applets or by creating an HTML page. The results are ranked according to their similarity to the submitted image.

3. Information Crawler

The image collection process is conducted by an autonomous Web agent or Crawler. The agent traverses the Web by following the hyperlinks between documents. It detects images, retrieves and transfers them for processing to the system server. The extracted features are then added to the database. The overall collection process, illustrated in Fig. 2, is carried out using several distinct modules:

- The Traversal Crawler - assembles lists of candidate Web pages that may include images or hyperlinks to them.

- The Hyperlink Parser - extracts the URLs of the images.

- The Retrieval Crawler - retrieves and transfers the image to the system server for further processing.
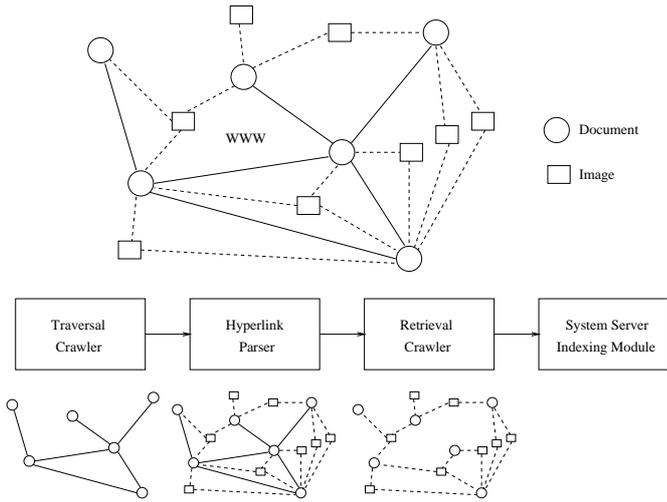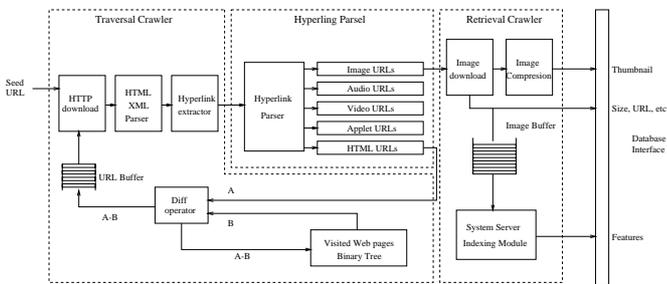
Figure 2. Image gathering process.

Figure 3. The Traversal and Retrieval crawlers.

3.1 Image Detection

In the first phase, the Traversal Crawler traverses the Web looking for images, as illustrated in Fig. 3. Starting from seed URLs, the Traversal Crawler follows a breadth-first search across the Web. It retrieves pages via Hypertext Transfer Protocol (HTTP) and passes the Hypertext Markup Language (HTML) code to the Hyperlink Parser. In turn, the Hyperlink Parser detects new URLs, encoded as HTML hyperlinks, and adds them back to the queue of Web pages to be retrieved by the Traversal Crawler. In this sense, the Traversal Crawler is similar to many of the conventional spiders or robots that follow hyperlinks in some fashion across the Web [5]. The Hyperlink Parser detects the hyperlinks in the Web documents and converts the relative URLs to absolute addresses. By examining the types of the hyperlinks and the filename extensions of the URLs, the Hyperlink Parser extracts the URLs of the images.

In the second phase, the list of image URLs from the Hyperlink Parser is passed to the Retrieval Crawler. The Retrieval Crawler retrieves the images and provides them as input to the indexing module. After the indexing procedure, the extracted features are added to the database. Another important function of the Retrieval Crawler is to extract attributes associated with the image such as URL, date of processing, size, width, height, file size, type of visual data, and so forth, and also generate a thumbnail icon, that sufficiently compacts and represents the visual information.

4. Image Indexing and Retrieval

4.1 Region Extraction

After the Information Crawler collects and transfers images to the server, image indexing algorithms process them in order to extract descriptive features. Based on the extracted regions, characteristic features are extracted using color, texture and shape/region boundary information. As a basis for the indexing, the K-Means algorithm is used. Clustering based on the K-Means algorithm is a widely used region segmentation method [6, 7] which, however tends to produce unconnected regions. This is due to the propensity of the classical K-Means algorithm to ignore spatial information about the intensity values in an image, since it only takes into account the global intensity or color information. In order to alleviate this problem, we propose the use of an extended K-Means algorithm: the K-Means-with-connectivity-constraint algorithm. In this algorithm the *spatial proximity* of each region is also taken into account by defining a new center for the K-Means algorithm and by integrating the K-Means with a component labeling procedure.

For the sake of easy reference we shall first describe the traditional K-Means algorithm (KM):

- Step 1 For every region $s_k, k = 1, \ldots, K$, random initial intensity values are chosen for the region intensity centers $\bar{I}_k$.

- Step 2 For every pixel $\mathbf{p} = (x, y)$, the difference is evaluated between $I(x, y)$, where $I(x, y)$ is the intensity value of the pixel and $\bar{I}_k, k = 1, \ldots, K$. If $|I(x, y) - \bar{I}_i| < |I(x, y) - \bar{I}_k|$ for all $k \neq i$, $\mathbf{p}(x, y)$ is assigned to region s_i.

- Step 3 Following the new subdivision, $\bar{I}_k$ is recalculated. If M_k elements are assigned to s_k then: $\bar{I}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} I(\mathbf{p}_m^k)$, where $\mathbf{p}_m^k, m = 1, \ldots, M_k$, are the pixels belonging to region s_k.

- Step 4 If the new $\bar{I}_k$ are equal with the old then stop, else goto *Step 2*.

The results of the application of the above algorithm are improved using the K-Means with connectivity constraint (KMC) algorithm, which consists of the following steps:

- Step 1 The classical KM algorithm is performed for a small number of iterations. This result in K regions, with color centers $\bar{\mathbf{I}}_k$ defined as:

$$\bar{\mathbf{I}}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} \mathbf{I}(\mathbf{p}_m^k), \qquad (1)$$

  where $\mathbf{I}(\mathbf{p})$ are the color components of pixel $\mathbf{p}$ in the YUV color space, i.e. $\mathbf{I}(\mathbf{p}) = (I_Y(\mathbf{p}), I_U(\mathbf{p}), I_V(\mathbf{p}))$. Spatial centers $\bar{\mathbf{S}}_k = (\bar{S}_{k,X}, \bar{S}_{k,Y}), k = 1, \ldots, K$ for each region are defined as follows:

$$\bar{S}_{k,X,Y} = \frac{1}{M_k} \sum_{m=1}^{M_k} p_{m,X,Y}^k, \qquad (2)$$

  where $\mathbf{p}^k = (p_X^k, p_Y^k)$. The area of each region A_k is defined as $A_k = M_k$ and the mean area of all regions $\bar{A} = \frac{1}{K} \sum_{k=1}^{K} A_k$.

- Step 2 For every pixel $\mathbf{p} = (x, y)$ the color differences are evaluated between center and pixel colors as well as the distances between $\mathbf{p}$ and $\bar{\mathbf{S}}$. A generalized distance of a pixel $\mathbf{p}$ from a subobject s_k is defined as follows:

$$D(\mathbf{p}, k) = \frac{\lambda_1}{\sigma_I^2} \|\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}_k\| + \frac{\lambda_2}{\sigma_S^2} \bar{A} \frac{\|\mathbf{p} - \bar{\mathbf{S}}_k\|}{A_k} , \quad (3)$$

  where $\|\mathbf{p} - \bar{\mathbf{S}}_k\|$ is the Euclidean distance, σ_I^2, σ_S^2 are the standard deviations of color and spatial distance, respectively and λ_1, λ_2 are regularization parameters. Normalization of the spatial distance, $\|\mathbf{p} - \bar{\mathbf{S}}_k\|$ with the area of each subobject, $\frac{\bar{A}}{A_k}$ is necessary in order to allow the creation of large connected objects; otherwise, pixels with similar color and motion values with those of large object would be assigned to neighboring smaller regions. If $|D(\mathbf{p}, i)| < |D(\mathbf{p}, k)|$ for all $k \neq i$, $\mathbf{p} = (x, y)$ is assigned to region s_i.

- Step 3 Based on the above subdivision, an eight connectivity component labeling algorithm is applied. This algorithm finds all connected components and assigns a unique value to all pixels in the same component. Regions whose area remains below a predefined threshold are not labeled as separate regions. The component labeling algorithm produces L connected regions. For these connected regions, the color $\bar{\mathbf{I}}_l$ and spatial $\bar{\mathbf{S}}_l$ and motion centers $l = 1, \ldots, L$, are calculated using equations (1) and (2) respectively.

Figure 4. (a) Original image "Claire', (b) Result of the KMC algorithm (Step 2, first iteration). (c) Result of the component labeling algorithm (Step3). (d) The final segmentation after only four iterations.

- Step 4 If the difference between the new and the old centers $\bar{\mathbf{I}}_l$ and $\bar{\mathbf{S}}_l$ is below a threshold, then stop, else goto Step 2 with $K = L$ using the new color and spatial centers.

Through the use of this algorithm the ambiguity in the selection of number K of regions, which is another shortcoming of the K-Means algorithm, is also resolved. Starting from any K, the component labeling algorithm produces or rejects regions according to their compactness. In this way K automatically adjusted during the segmentation procedure.

In Fig. 4 an example of the segmentation procedure is shown. In Fig. 4a the original image of the videoconference image "Claire" of size 176×144 is shown. In Fig. 4b the result of the first iteration of the KMC algorithm is shown (result of Step 2 of the KMC algorithm, for the first iteration). In Fig. 4c the result of the component labeling algorithm is shown (Step 3 of the algorithm). The initial number of subobject was set to $K = 5$ and the component labeling algorithm produced $L = 6$ regions. In Fig. 4d the final segmentation after only four iterations is shown.

4.2 Region Descriptors

We store a simple description of each region's color, texture and spatial characteristics. For each extracted region k, the color centers $\bar{\mathbf{I}}_k$ as estimated during the segmentation procedure are stored. For each image region we also store the mean texture descriptors (i.e., anisotropy, orientation, contrast).

The geometric descriptors of the region are simply the spatial center $\bar{\mathbf{S}}_k$ and covariance or scatter matrix $\mathbf{C}_k$ of the region. The centroid $\bar{\mathbf{S}}_k$ provides a notion of position, while the scatter matrix provides an elementary shape description. In the querying process discussed in Section 4.3, centroid separations are expressed using Euclidean distance. The determination of the distance between scatter matrices, which is slightly more complicated, is based on the three quantities
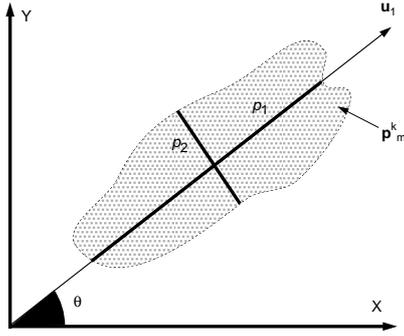
4

Figure 5. Extraction of shape descriptors.

$[\det(S)]^{1/2} = \sqrt{\rho_1 \rho_2}$, $1 - \rho_1/\rho_2$ and θ (ρ_1 and ρ_2 are the eigenvalues and θ the argument of the principal eigenvector of $\mathbf{C}_k$). These three quantities represent approximate area, eccentricity and orientation.

Specifically, if $\mathbf{p}_m^k = [p_{m,X}^k, p_{m,Y}^k]^T$, $m = 1, \ldots, M_k$ are the pixels belonging to region k with coordinates $p_{m,X}^k$, $p_{m,Y}^k$ then the covariance (or scatter) matrix of region k is

$$\mathbf{C}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} (\mathbf{p}_m^k - \bar{\mathbf{S}}_\mathbf{k})(\mathbf{p}_m^k - \bar{\mathbf{S}}_\mathbf{k})^T .$$

Let ρ_i, $\mathbf{u}_i$, $i = 1, 2$ be its eigenvalues and eigenvectors: $\mathbf{C}_k \mathbf{u}_i = \rho_i \mathbf{u}_i$ with $\mathbf{u}_i^T \mathbf{u}_i = 1$, $\mathbf{u}_i^T \mathbf{u}_j = 0$, $i \neq j$ and $\rho_1 \geq \rho_2$. As is known from Principal Component Analysis (PCA), the principal eigenvector $\mathbf{u}_1$ defines the orientation of the region and $\mathbf{u}_2$ is perpendicular to $\mathbf{u}_1$. The two eigenvalues provide an approximate measure of the two dominant directions of the shape (Fig. 5).

4.3 Image Retrieval by Querying

In our system, similarly to that in [8], the user composes a query by submitting an image to the segmentation/feature extraction algorithm in order to see its segmented representation, selecting the regions to match, and finally specifying the relative importance of the region features. Once a query is specified, we score each database image based on how closely it satisfies the query. The score μ_i for each query is calculated as follows:

1. Find the feature vector f_i for the desired region s_i. This vector consists of the stored color, position, and shape descriptors (Section 4.2).

2. For each region s_j in the database image:

   (a) Find the feature vector f_j for s_j.

   (b) Find the Mahalanobis distance between f_i and f_j using the diagonal covariance matrix (feature weights) Σ set by the user: $d_{ij} = [(f_i - f_j)^T \Sigma^{-1}(f_i - f_j)]^{1/2}$.

   (c) Measure the similarity between f_i and f_j using $\mu_{ij} = e^{\frac{d_{ij}}{2}}$. This score is 1 if the regions are identical in all relevant features; it decreases as the match becomes less perfect.

3. Take $\mu_i = \max_j \mu_{ij}$.

We then rank the images according to the overall score and return the best matches, along with their related information such as URL, date, size, etc.

5. Experimental Results

The proposed system has been used for image collection, segmentation into regions and region feature extraction from real Web sites and for test images. Results of images segmented into regions are shown in Fig. 6 [1]. Features are then extracted from the regions and stored in the database. The segmented regions correspond to semantic objects, allowing efficient indexing and retrieval.

In Fig. 7 a query using the user-friendly WWW interface is shown. The user starts a query by selecting the snow region (white color). The system returns a set of images also containing snow regions and another one containing a region of white houses. The majority of results appear to have high semantic relevance with the submitted image.

6. Conclusions and Future Work

In this paper the development of an intelligent image content-based search engine for the World Wide Web is presented. Information Web Crawlers continuously traverse the Internet and collect images that are subsequently indexed based on integrated feature vectors. As a basis for the indexing, the K-Means algorithm is used, modified so as to take into account the coherence of the regions. Based on the extracted regions, characteristic features are extracted using color, texture and shape/region boundary information. These features along with additional information such as the URL location and the date of index procedure are stored in a database. The user can access and search this indexed content through the Web. The output of the system is a set of links to the content available in the WWW, ranked according to their similarity to the image submitted by the user.

In future work, we will emphasize on other visual dimensions such as texture. We also plan to extend our technique so as to handle video indexing by extracting motion features.

---

[1] For more results: http://uranus.ee.auth.gr/Istorama

5

Figure 6. Images segmented into regions.

Figure 7. Example of a query using the WWW interface.

References

[1] G. S. Jung and V. N. Gudivada, "Autonomous tools for information discovery in the world-wide web," Tech. Rep. CS-95-01, School of Electrical Engineering and Computer Science, Ohio University, Athens, OH, 1995.

[2] S. Sclaro, "World wide web image search engines," in *NSF Workshop on Visual Information Management*, Cambridge, MA, June 1995.

[3] J. R. Smith and S.-F. Chang, "Visually Searching the Web for Content," *IEEE Multimedia Magazine*, vol. 4, no. 3, pp. 12–20, Summer 1997.

[4] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences," *IEEE Trans. on Circuits and Systems for Video Technology*, (to appear) 2000.
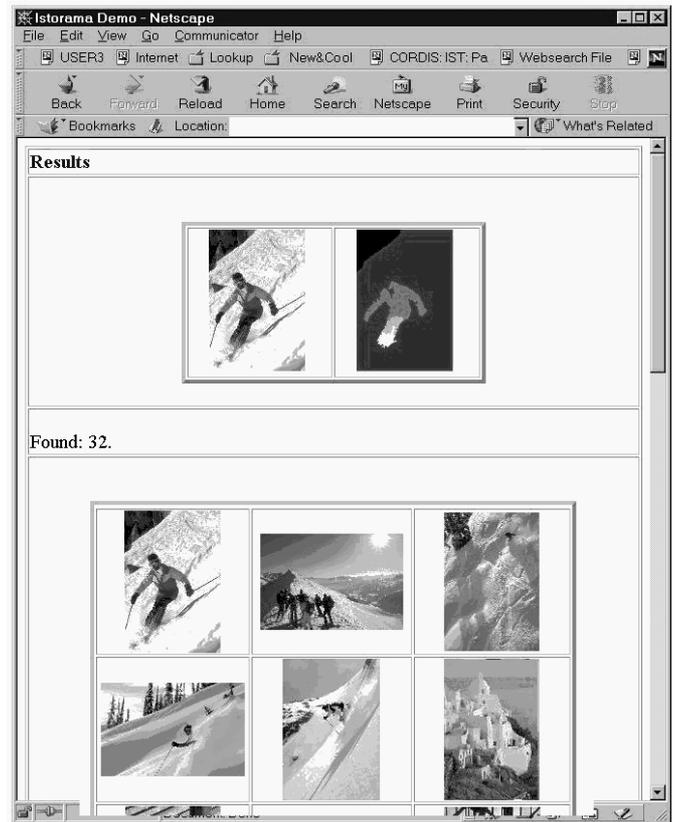
[5] M. Koster, "Robots in the web: threat or treat?," *ConneXions*, vol. 9, no. 4, April 1995.

[6] S. Z. Selim and M. A. Ismail, "K-means-type algorithms," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 6, no. 1, pp. 81–87, Jan. 1984.

[7] I. Kompatsiaris, G. Mantzaras, and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects in Color Image Sequences," in *2000 IEEE International Symposium on Circuits and Systems (ISCAS 2000)*, Geneva, Switzerland, May 2000.

[8] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-based image querying," in *CVPR '97 Workshop on Content-Based Access of Image and Video Libraries*, 1997.