SPIRE: A Progressive Content-Based Spatial Image Retrieval Engine

Chung-Sheng Li, Lawrence D. Bergman, Yuan-Chi Chang, Vittorio Castelli, John R. Smith

Data Management Department, IBM T. J. Watson Research Center, P. O. Box 704, Yorktown Heights, NY 10598

(914) {784-6661,784-7946,784-7327,945-2396,784-7320}

{csli,bergmanl,yuanchi,vittorio,jsmith}@us.ibm.com

ABSTRACT

In this demo, we will show the implementation of a content-based SPatial Image Retrieval Engine (SPIRE) for multimodal unstructured data. This architecture provides a framework for retrieving multi-modal data including image, image sequence, time series and parametric data from large archives. Dramatic speedup (from a factor of 4 to 35) has been achieved for many search operations such as template matching, texture feature extraction. This framework has been applied and validated in solar flares and petroleum exploration in which spatial and spatial-temporal phenomena are located.

Keywords

content-based retrieval, digital library, multimedia database.

1. INTRODUCTION

Despite the tremendous progress in the content-based retrieval systems during the recent decade, there still exists a number of outstanding issues: (1) Almost all of the content-based search engines are based on similarity matches on features extracted from a single data modality such as images or video. There still lacks a framework for performing similarity search across multiple modalities simultaneously. Combining fuzzy information from multiple systems has been investigated in the IBM Garlic project. However, only fuzzy conjunction or disjunction among attributes of the same data object has been considered. A more general framework that allows combining fuzzy information through spatial and temporal rules is still needed. (2) Scalability of the content-based search on a combination of raw data, features, semantics, and parametric data is yet to be investigated.

2. FRAMEWORK

In the SPIRE framework, a simple search object can be specified at different abstraction levels - raw data, feature, semantics, metadata, or a combination of the above. A composite search object can then be constructed from multiple simple objects with spatial, temporal, or Boolean relationships. During query processing time, the composite object query is transformed and processed sequentially using an algorithm based on dynamic programming. This algorithm eliminates the possibility of false dismissal of candidate combinations without having to perform an exhaustive search. The scalability of the proposed scheme is achieved by adopting a progressive framework, in which a hierarchical scheme is used to decorelates and reorganize the information contained in the images at all of the abstraction levels. Consequently, the search operators can be applied on a much smaller portions of the data and progressively refine the search results. This technique achieves a significant speedup as compared to more conventional implementations. The speedup factor for template matching (the search operator at the raw pixel level) and classification (the search operator at the semantic level) is more than 20 times. A 400% to 800% speedup has also been achieved for texture extraction and matching (the search operator at the feature level).

3. PROTOTYPE

Queries are constructed and parsed syntactically at the java-based client using a drag-and-drop interface in conjunction with an image navigation browser. This interface also allows the definition of new objects and features. After being specified and syntactically analyzed, the query is then sent to the server. A query usually involves both metadata search and potentially multi-modal search on both images and time series. Based on the progressive framework and the SPROC algorithm, the search engine works in conjunction with the query parser to parse the query string into a script consisting of a set of SQL statements and content-based search operators. The IBM DB2 database engine performs search on the metadata, while the image content search is performed by the content search operators scheduled and sequenced by the search engine. Usually the metadata search serves as a pruning mechanism, to achieve maximum reduction of the search space during the first stages of the query execution process. The content-based search engine also accesses the feature descriptors as well as the raw pixels of an image region in order to compute the final results. The results of the search are rendered by the visualization engine.

In the demo, we will show a reservoir modeling scenario that often arises in the oil/gas exploration and production environments. A simple object for *laminated sandstone* based on the texture feature is defined through the drag-and-drop java interface with both examples and counter examples. The query results from a dataset consisting of 1000 feet of resistivity measurement of borehole and well log data are then displayed. This scenario allows the capture of geologist's knowledge of definitions of various geological structures, and thus facilitate interpretation of vast amount of oil/gas exploration data.

ACM SIGMOD 2000 5/00 Dallas, TX, USA