Content-Based Image Retrieval Using Hierarchical Temporal Memory

Bruce A. Bobier
Department of Computing and Information Science
University of Guelph
Guelph, ON, N1G 2W1
bbobier@uoguelph.ca

Michael Wirth
Department of Computing and Information Science
University of Guelph
Guelph, ON, N1G 2W1
mwirth@uoguelph.ca

ABSTRACT

Several querying interfaces for content-based image retrieval (CBIR) are reviewed and a new CBIR system is introduced that uses Hierarchical Temporal Memory for the automatic indexing of architectural images and provides a sketch-based and iconic index querying interface. Experimentation shows the system is robust for recognizing query images under varying amounts of noise, distortion, occlusion, blurring, and affine transformation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Retrieval models; I.2.6 [**Image Processing and Computer Vision**]: Connectionism and neural nets

General Terms

Experimentation, Performance, Theory

Keywords

Content-based Image Retrieval, Hierarchical Temporal Memory, Querying Interfaces

1. INTRODUCTION

The vast amount of digital visual information stored in online databases has increased the need for more efficient and effective means of its management and retrieval. Content-based image retrieval (CBIR) refers to the problem of searching for images in large databases using techniques from computer vision, artificial intelligence and pattern recognition [2]. Content-based approaches are needed for the management of visual information in cases where textual annotations for images are either incomplete or nonexistent. Content-based image retrieval (CBIR) has been used for indexing and retrieving numerous types of visual information including photographs, medical images, videos, line drawings, sketches, artwork, and 3D images [7]. In this article, the focus is placed on CBIR for the automatic indexing and retrieval of architectural elevation drawings. Elevation drawings are manually created with pen and paper and digitized as binary, grey scale or color images using a scanner or overhead camera (Figure 1). Presently, thousands of these drawings are stored in online databases, with even more known to exist in offline archives and collections. In the coming years when the offline collections are digitized and combined with online collections, the need for an effective means of managing and retrieving these images based on their contents will be of even greater importance.

Figure 1: Example of an elevation drawing.

Current image indexing schemes often rely on textual information such as the name, date, location, and architect of the represented building, which results in databases that index the circumstances of the image rather than the content. Color and texture-based indexing and retrieval schemes are similarly limited, as for most architectural design tasks, color and shading features are less relevant to the user than are form and spatial features [5].

This paper presents a new approach to CBIR, dubbed HTMCBIR, that uses the Numenta NuPic platform [4] to provide an intelligent system that aims to understand a query's semantics, rather than the low-level image features for indexing and retrieval using iconic indexes and a sketch-based interface. The system is trained on six categories of architectural elements and is shown to be robust to noise, distortion, occlusion, blurring, and affine transformations.

2. QUERYING INTERFACES

Formulating and specifying a query has been performed using numerous approaches, the most common being to describe images using textual keywords, which requires previous manual annotation of the image data. Keyword searches

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.

are further limited by their low scalability, poor ability to capture semantic information and that users prefer to retrieve images based on their content, rather than their associated keywords [3, 6].

Query-by-example (QBE) is another approach that involves the user supplying or selecting an image that is exemplary of the type of content they wish to retrieve. When an image is not provided, the user is presented with a pseudo-random selection of images of sufficiently diverse content or feature values, and upon selection of one or more images, the system retrieves the set of most closely related images. This process may be repeated to iteratively refine the results until the user is satisfied. Relevance feedback may also be provided by the user for each set of presented images, for each of which they assign positive or negative feedback as a means of refining their query. This approach requires considerable user interaction and often performs poorly at interpreting for what image attributes and features the user specified their feedback [8].

Query by shape involves the user constructing their query image by dragging geometric primitives onto a drawing canvas, although such approaches may not be suitable for the problem domain, and limit the user's ability to specify less structured or free-form queries.

A more versatile approach to querying by shape employs a sketch-based interface with which the user can form queries by drawing free-form objects or using simple geometric shapes. For the retrieval of architectural drawings, this approach is most suitable as it affords the most familiar interface for users and may also include predefined shapes, layouts and textures.

Iconic indexes are symbolic descriptors of image data or relationships, and are among the more suitable approaches to querying elevation drawing databases, as the hierarchical organization of architectural elements enables the user to iteratively refine their query using multiple iconic indexes (e.g. window - transom - Georgian-style transom).

3. HIERARCHICAL TEMPORAL MEMORY

Hierarchical temporal memory (HTM) is a recent paradigm that models some of the structural and algorithmic properties of the human neocortex using elements of Bayesian networks such as the constant sharing of information between nodes and Belief Propagation. Although HTMs are similar to Bayesian networks, they differ in that HTMs have a clear parent/child relationship, are self-training and can more easily handle time-varying data [4]. An HTM is represented as a tree-shaped multi-level hierarchy of nodes, where information can flow in both vertical directions.

The nodes in Level 1 of the HTM receive sensory input directly from the image and use this data to construct a model of the HTM's environment by looking for spatiotemporal correlations in the input data points. Each level 1 node is supplied with sensory data from a small portion of the image, such that all of the input data is distributed equally across the lowest level nodes without overlap.

Figure 2 illustrates an HTM network with three levels, where the input is a 32x32 pixel image. At level 1, the 32x32 pixel input is received directly from the sensors and distributed across 64 nodes, with each node perceiving a 4x4 pixel area. At level 2, each node receives its input from the outputs of four level 1 nodes, which describe an 8x8 pixel area in the input image.

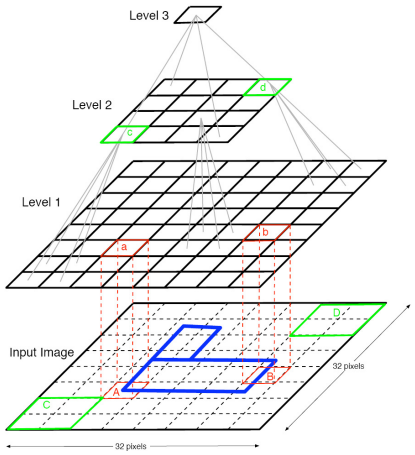

Figure 2: HTM with 3 levels (adopted from [4]).

Whereas the lowest level nodes receive their input from sensory data about the environment, nodes in higher levels of the hierarchy are provided with input comprised of their children's beliefs. Every node looks for spatial and temporal patterns, and the spatial patterns of higher level nodes are formed from the commonly recurring patterns of the beliefs reported by their children, while the temporal patterns are composed from the recurring changes of their children's beliefs.

Every node in an HTM network shares a common algorithm, regardless of its position in the hierarchy. When new data is submitted to the HTM, each node forms beliefs about the input data by creating two 2-column tables for each of its spatial and temporal beliefs. In each table, the left column enumerates the spatial or temporal patterns that the node has learned, and the right column reports the corresponding probability of the patterns occurring. During each cycle of input data being presented, each node performs two steps. First, the node assigns to each spatial quantization point the probability that the current input data matches this point. Second, the node searches for common sequences of quantization points, and represents each sequence as a variable. Over time, the node determines the probability that the current input belongs to each each sequence and assigns this probability to each variable. The probability that the input belongs to each sequence variable is added to a vector of probabilities and passed up the hierarchy to serve as input for the node's parent. Each node may also pass belief information down the hierarchy to its children, such as the spatial pattern it anticipates to encounter next, based on the temporal sequence that is believed to be currently occurring.

Nodes in lower levels deal with simple events that change quickly and occupy smaller spatial areas, while nodes in higher levels combine series of input patterns to form more stable groups over a greater range of data. Specifically, higher nodes sense more complex structures, namely patterns of patterns, which evolve less quickly and exist in larger spatial areas.

During each cycle, a node determines the distance between the input data and each quantization point. The causes discovered by lower level nodes are causes of low complexity,

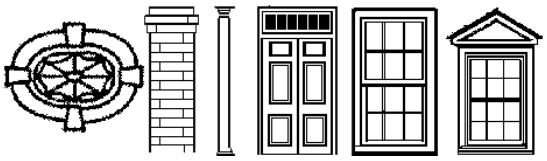

Figure 3: Iconic indexes of image categories.

Category	Training Images	Testing Images
chimney	37	7
column	14	3
door	21	4
dormer	13	3
roundWindow	19	3
window	37	7

Table 1: Number of training and testing images per object category for HTMCBIR.

such as edges, lines and corners, which can serve as components in higher level causes and allows for scalable and efficient memory usage, as the memory used to store low level causes is shared by the high level causes. However, a shortcoming of this paradigm is that a trained network has difficulty learning to recognize new objects that are not composed of previously learned sub-objects.

4. CBIR USING HTM

In this section, the HTMCBIR system is introduced that uses Numenta's NuPic [4] as a basis to perform automatic indexing, querying and retrieval of architectural drawings. The network topology of the HTMCBIR system consists of four levels, where the nodes in the lowest level operate directly on the pixel data, and nodes in each subsequent higher level operate on the belief vectors produced by the previous level. The network is trained on a data set of 141 8-bit greyscale images that are divided into 6 categories (see Table 1). All of the training and testing images are of architectural elements that are commonly found in elevation drawings of houses and were cropped from actual drawings in the Library of Congress' Historic American Building Survey [1] and were converted to greyscale and resized. The original images were approximately 15,000 pixels square, and because of the drastic resizing, some of the images were manually edited to reconnect lines and restore information lost in downsampling.

Two methods of specifying a query are provided in HTMCBIR. The first falls under the iconic indexes paradigm, in which the user selects an iconic representation of an object category from the "Training Images" panel of the main window (Figure 4). The second method allows the user to sketch a line drawing query on the 128x128 drawing canvas. With both methods, once the query is entered into the drawing canvas, the user clicks the "Recognize Picture" button and the query is submitted to the system for recognition and classification. These query methods were selected as the majority of elevation drawings are grey scale or binary images, for which color- and texture-based approaches are not applicable. Further, they allow the user to visually specify their query in terms of object shape and spatial layout,

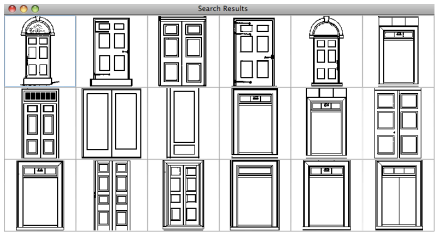

Figure 5: Retrieved images from the "door" category.

which provides a more usable and intuitive interface that is not influenced by the user's language or vocabulary.

To retrieve images based on the query image, HTMCBIR attempts to infer the category of the sketch based from the set of learned categories. The results of the recognition process are displayed in the main window as the three best matching object categories, with each possible category's degree of certainty being indicated by a bar chart (Figure 4). If the correct category is not included in this list, the user alters their query and has the system attempt to recognize it again. Once the intended category is included in the recognition results, the user clicks on the category's icon or label under the bar chart to retrieve all images of that category. This opens a secondary window containing thumbnail versions of each stored image in that category (Figure 5). Clicking on one of the thumbnails displays a zoomed-in version of the selected image and its location on disk.

5. QUERY RECOGNITION ACCURACY

To measure the performance of HTMCBIR for recognizing visual queries, the system was evaluated on seven criteria: recognition accuracy for an untouched testing corpus, accuracy with testing data distorted with lines, noise, occlusion, translation, blur, and scaling. The experiments use the same HTM network that is trained using 141 clean 128×128 pixel images from 6 categories. Here, "recognition accuracy" is operationally defined as the number of correctly classified query images divided by the total number of testing images.

The first experiment evaluates the HTMCBIR's ability to recognize hand drawn visual queries using a test set of 27 images taken from [1] and aims to measure the generalization ability of the system for a large corpus of testing data. A single run of this experiment was conducted, as static data sets are used for training and testing. Using the testing data set, a recognition accuracy of 92.6% was observed, wherein 25 of the query images were correctly recognized.

To test the system's robustness to noisy queries, random lines and noise were added to the testing data set. The random lines consist of pixel wide lines of random lengths drawn at random location. As the remaining experiments are non-deterministic, 50 runs were conducted for each. Using the 27 test images, mean recognition accuracies of 89.3% were observed when with two and four lines were added, and 88.6% with eight lines.

To add noise to the testing images, each pixel in the image was perturbed by a random amount with probability $p = \{15, 30, 90\}$, for which mean recognition accuracies of 89.3% were with each value of p.

The system was also tested for its ability to recognize queries under occlusion. First, a single black rectangle of

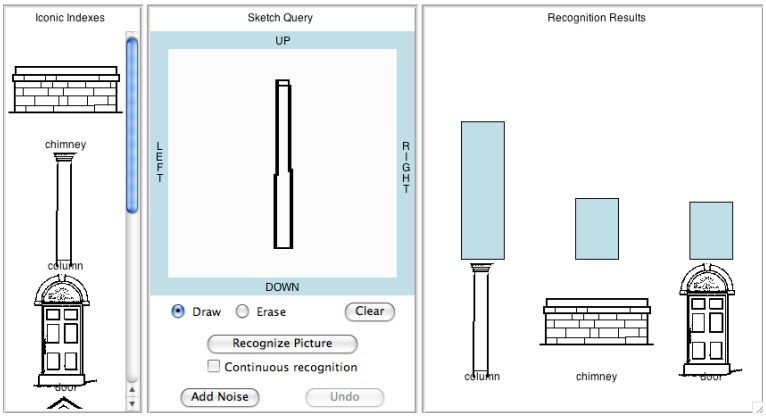

Figure 4: HTMCBIR application showing iconic indexes, sketch-based interface, and recognition results.

random dimensions between 13x13 and 26x26 was added to each testing image, for which the system correctly recognized 87.8% of the queries on average over 50 runs. For the second and third occlusion testing variations, four and eight black rectangles of random dimensions between 13x13 and 51x51 were added to each test image, resulting in mean recognition accuracies of 38.4% and 17.4% respectively. Note that with four and eight occlusions, more than 50% of the test images were often occluded.

To test the system's ability to recognize queries under translation, the testing images were translated horizontally and vertically by a random amount in the range of [-10,10], [-19,19], and [-38,38] pixels, for which mean accuracies of 74.2%, 66.4%, and 45.3% were observed respectively.

The system's ability to recognize blurred images was also tested, wherein each testing image was blurred using the Python Imaging Library's `ImageFilter.BLUR` function. When the images were blurred once, a recognition accuracy of 85.7% was observed, and when repeatedly blurred a random number of times in the range [2, 3], and [4,8], recognition accuracies of 63.7% and 55.9% were observed.

The scale invariance for query recognition was also tested by scaling the images a random amount between 80-120%, 60-140% and 30-170% of their original size, resulting in recognition accuracies of 81.4%, 70.4% and 57.6% respectively.

6. CONCLUSION

This article has reviewed prominent querying interfaces for CBIR and introduced a new approach for automatically indexing architectural drawings and retrieving them with a usable interface. For indexing, many current approaches fail to capture the semantic information contained in the images by using image features and attributes that greatly differ from those used by the human vision system for object recognition and clustering of related images. Hierarchical Temporal Memory is a biomimetic approach to the vision problem, that in the context of CBIR, aims to "bridge the semantic gap" by translating the low-level features to high-level concepts that are more easily understood by the user and allow them to specify queries using their own terminology. The HTMCBIR system presented here, which extends Numenta's NuPic implementation, has been shown to provide promising results for indexing and recognizing small greyscale images. The querying interface of HTMCBIR allows the user to quickly and easily specify a query, and the query image recognition algorithm was shown to be robust to spatial noise, occlusions, blurring, and affine transformations despite having been trained on only clean, undistorted images.

With additional training data and parameter tuning, it is believed that the HTMCBIR system is capable of achieving a higher recognition accuracy. The purpose of this implementation is a proof of concept to show that HTM is sufficiently flexible to provide efficient and accurate indexing of line drawings. Future work may investigate the scalability of HTM and evaluate its suitability for processing large data sets (e.g. the Library of Congress' Historic American Building Survey has approximately 30,000 elevations drawings that are ∼15,000 pixels square [1]).

7. REFERENCES

[1] Historic American Buildings Survey (HABS). http://memory.loc.gov/ammem/collections/habs_haer, 2007.

[2] Vittorio Castelli and Lawrence D. Bergman. *Image Databases: Search and Retrieval of Digital Imagery*. Wiley-Interscience, 1st edition, December 2001.

[3] Ritendra Datta, Jia Li, and James Z. Wang. Content-based image retrieval: approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM workshop on Multimedia information retrieval*, pages 253–262, New York, NY, 2005. ACM.

[4] Dileep George and Bobby Jaros. The HTM learning algorithm. Numenta Inc. Whitepaper, 2007.

[5] Mark D. Gross and Ellen Yi-Luen Do. Diagram query and image retrieval in design. In *ICIP '95: Proceedings of the 1995 International Conference on Image Processing*, volume 2, pages 2308–2313, Washington, DC, USA, 1995. IEEE Computer Society.

[6] Surya Nepal and M. V. Ramakrishna. Query processing issues in image(multimedia) databases. In *Proceedings of the 15th International Conference on Data Engineering*, pages 22–29, Washington, DC, 1999.

[7] R. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical report, Utrecht University, the Netherlands, 2002.

[8] Xiang S. Zhou and Thomas S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, April 2003.