



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Image Content Engine (ICE): A System for Fast Image Database Searches

J. M. Brase, D. W. Paglieroni, G. F. Weinert, C. W. Grant, A. S. Lopez, S. Nikolaev

March 23, 2005

SPIE Defense and Security Symposium
Orlando, FL, United States
March 28, 2005 through April 1, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Image Content Engine (ICE): A System for Fast Image Database Searches

James M. Brase, Douglas N. Poland, David W. Paglieroni, George F. Weinert,
Charles W. Grant, Aseneth S. Lopez and Sergei Nikolaev
Lawrence Livermore National Laboratory, P.O. Box 808, L-210, Livermore, CA 94551

ABSTRACT

The Image Content Engine (ICE) is being developed to provide cueing assistance to human image analysts faced with increasingly large and intractable amounts of image data. The ICE architecture includes user configurable feature extraction pipelines which produce intermediate feature vector and match surface files which can then be accessed by interactive relational queries. Application of the feature extraction algorithms to large collections of images may be extremely time consuming and is launched as a batch job on a Linux cluster. The query interface accesses only the intermediate files and returns candidate hits nearly instantaneously. Queries may be posed for individual objects or collections. The query interface prompts the user for feedback, and applies relevance feedback algorithms to revise the feature vector weighting and focus on relevant search results. Examples of feature extraction and both model-based and search-by-example queries are presented.

Keywords: broad area search, model matching, image search, image analysis

1. INTRODUCTION

New sensor and communication technologies provide scientists and analysts with a deluge of data in many forms: images, simulations, experimental measurements, communications intercepts, and textual data. Extracting knowledge from these masses of data requires the ability to discover subtle correlations in complex data sets – correlations that indicate relationships that lead to understanding

The *Image Content Engine (ICE)* is a set of software tools which allow an analyst to search a large set of images for a specific type of object. ICE provides two approaches for specifying the search: the target can be specified by a three-dimensional model or it can be specified by a set of example images. ICE provides image pipeline tools for extracting image features and matching object models, interactive relevance feedback tools for providing feedback on search performance, and tools for relational queries of the resulting detection databases.

The specific targets of ICE are very large image sets – terabytes in many cases. The feature extraction and object detection pipelines are implemented on parallel Linux clusters. A toolbox for mapping image processing pipelines to clusters and managing associated metadata with XML has been developed. The ICE system has been demonstrated on small clusters and is being scaled to larger systems for specific applications. In this paper we will describe the ICE system architecture and give examples of model-based target search and search-by-example applications.

2. THE ICE ARCHITECTURE

The top-level architecture of ICE is shown in Figure 1. We begin at the upper left with a set of images that is to be searched. The source of images can be either a real-time sensor stream or a pre-existing database. Image features and target matching calculations are performed in a set of image processing pipelines shown on the left side of the Figure. The tile feature pipeline computes a pre-defined set of features for each image tile – small (typically 128 pixels on a side) regularly-spaced overlapping rectangular areas. The object matching pipeline performs phase-sensitive image matching with a predefined user-selected set of three-dimensional object models. Extracted image features and target matching metrics are stored in a database.

The ICE search interfaces allow the user to interactively select target matches from a set of thumbnail images defined by peaks in the object matching metric. Search targets can also be specified by sets of example images. The query interface returns image tiles with feature distributions similar to the query examples. The use of both the model-based search and the query-by-example interfaces will be described in more detail in Sections 3 and 4.

Object matches that are confirmed by the user can be stored in an object database. These objects and their characteristics can be utilized by higher-level semantic graph-based knowledge management systems that correlate data from multiple sources.

A GIS interface is being developed for ICE which will allow the results of searches to be displayed as layers in a GIS. This capability will allow image information from multiple sources to be viewed in a common geospatial coordinate system and allow correlation between image objects detected by ICE and a priori map information. The search strategy in ICE will also be controllable by the GIS interface, allowing us to search specific map-specified areas. For example a search for a specific class of vehicles can be set up to follow known roads on a map.

The heavy computational loads in ICE reside in the image processing pipelines. To implement these pipelines efficiently we have developed a set of software tools for mapping arbitrary pipeline operations on Linux cluster-based computer architectures. Image blocks are allocated to sets of processors by a master control process. Image metadata is managed by passing XML descriptors with the image data blocks using the ISP toolkit. We are testing the scalability of the system using the MCR parallel system at LLNL. Most operations to date have been on small subsets of MCR's 989 nodes but the data-parallel approach should allow scaling to large parallel systems.

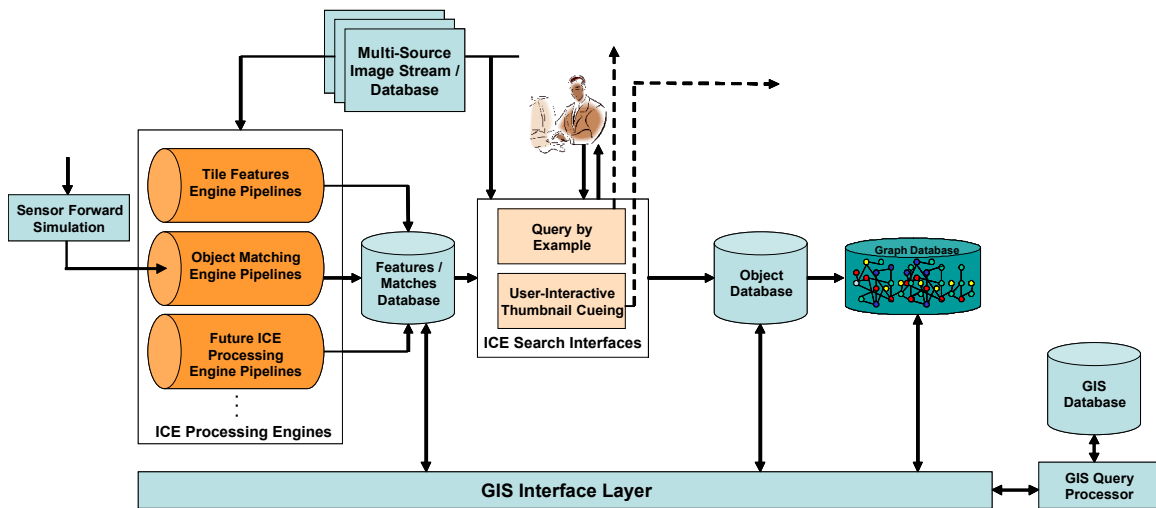


Figure 1: Architecture of the ICE image search system.

3. MODEL-BASED TARGET SEARCH

The principal function of ICE is to search large image sets for objects defined by a three-dimensional target model. The target model is given as a 3D graphics model file. It is projected into image space by a sensor model which captures the imaging geometry, illumination, and any other system aspects that influence the resulting image. Current models are relatively simple but interfaces to arbitrarily complex models are in place. The sensor model must adapt its performance to the quality of the image being searched. If the image is low resolution too many details in the model can degrade matching performance.

Matching the projected model to the image is performed by a phase-sensitive detection algorithm. We compute the local image gradient or image flow and match the phase angle off the gradient with that of the model. This method reduces sensitivity to illumination changes. The algorithm is described in detail in [1].

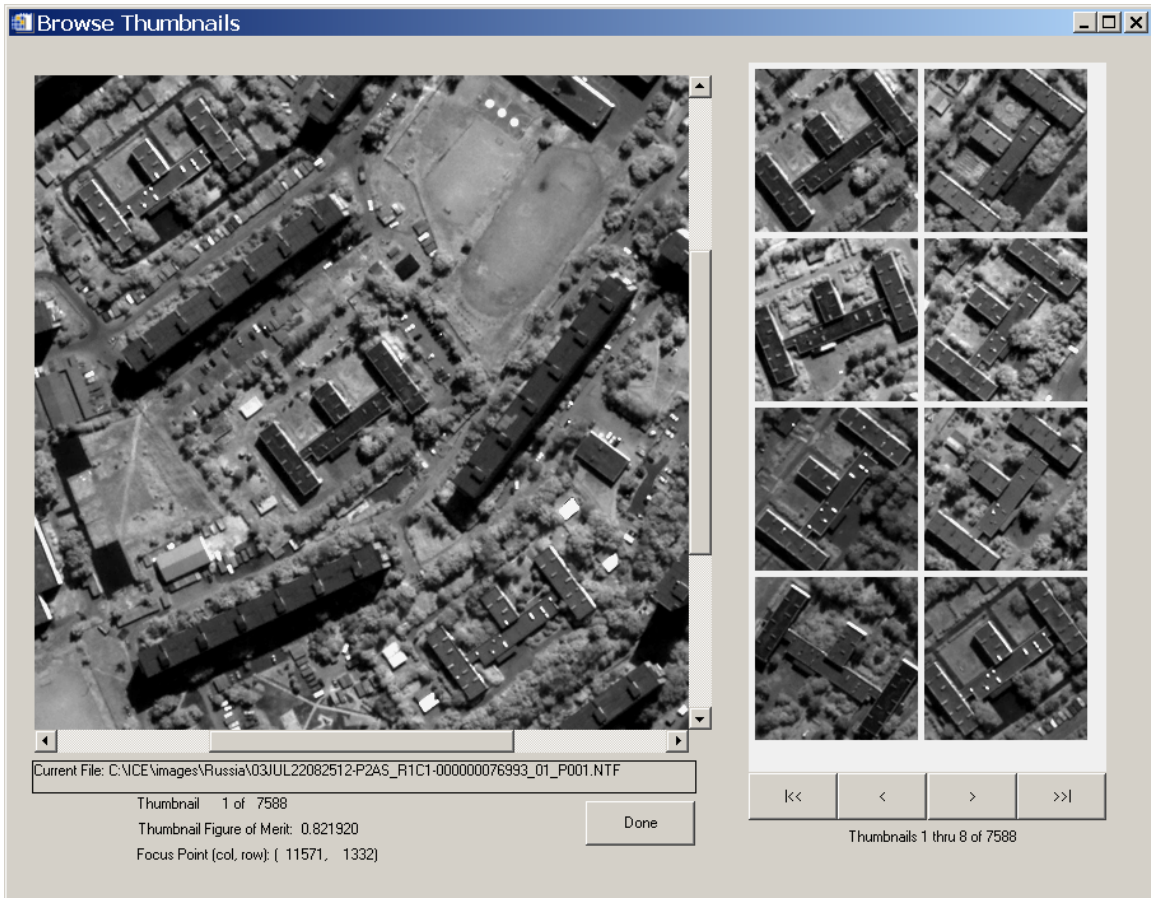


Figure 2: The user interface for selecting detection thumbnails from the model-based search process (Images courtesy of Digital Globe).

Figure 2 shows an example of the user interface for model-based search. In this case the model described the distinctive building shown in the center of the image. The thumbnails on the right show multiple detections of this class of building in the image database. The thumbnail positions are determined by locating peaks in the matching metric surfaces. Clicking on any of the thumbnails brings up a wider view of that specific image in the left window. The user can scroll down through the set of returned thumbnails and mark those that are acceptable detections.

4. SEARCH BASED ON EXAMPLES

An alternative approach to specifying a search is through the use of example images [2,3,4]. Each image tile – an arbitrary rectangular set of pixels – is described by a hierarchical set of image features, the feature vector. If we choose an example region and compute its feature vector, we can then search the index of feature vectors for the full image database and return those that are in some sense closest to the query feature vector.

Our approach to computing a feature vector is shown in Figure 3. The image is represented by a hierarchical set of feature maps. At the first level we compute a 5-scale Laplacian pyramid [5] representing local image contrast at each scale. A thresholding saliency operator selects features in the maps to be passed to the higher levels. The saliency

operator retains prominent features and eliminates low-valued features. At the second level we compute sets of orientation features using Gabor kernel operators [5]. These filters produce maps representing local orientation in the image tile at each scale. The third level in the hierarchy computes local pairwise correlations between orientations. For example, a corner is represented by the coincidence of two orientations at near ninety degree angular separation. The multiple scales are retained throughout the hierarchy.

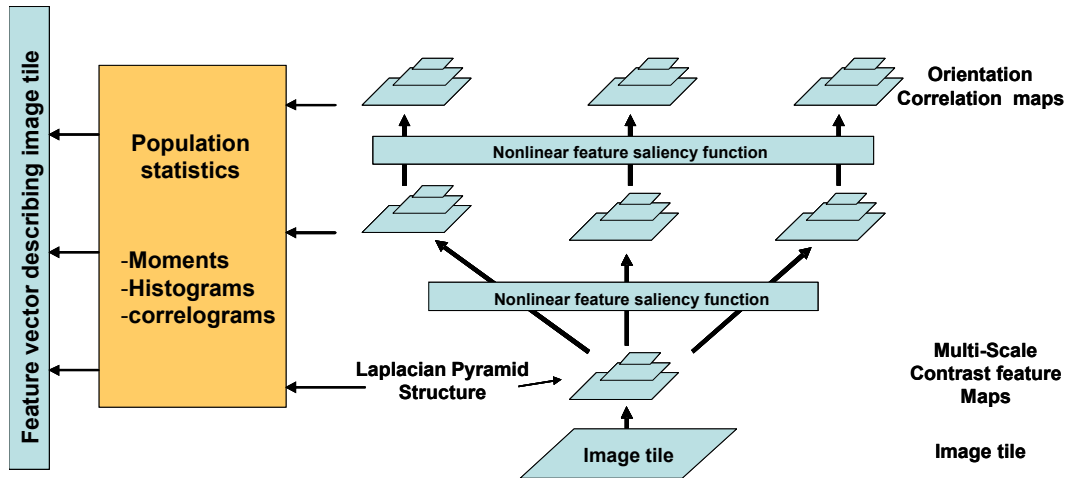


Figure 3: Hierarchical image representation used for image searches. Image feature vector components are given by population statistics of the feature maps.

At the completion of this process the original image tile is represented by a set of feature maps with varying scales and orientations. We then compute the population statistics of each feature map, the mean, standard deviation, min, and max, and insert these values into a feature vector. The assembled set of populations statistics then serves as the signature for the tile. In a typical case we use 5 scales, 7 angles, and 13 angular separations or 105 feature maps. Given four statistics for each feature map we have a 420 dimensional feature vector.

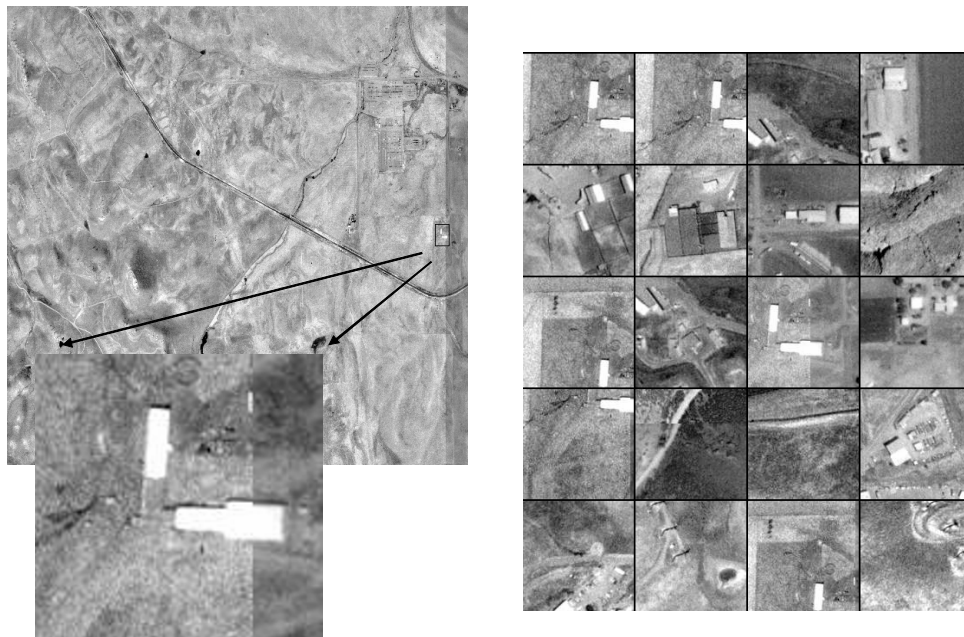


Figure 4: Query by example results for a search for buildings in the Altamont Pass area of California (Images courtesy of Terraserver USA).

5. QUERIES AND RELEVANCE FEEDBACK

The matching algorithms described in Sections 3 and 4 produce 2-dimensional match surfaces that represent the similarity of the query image to the target object at each location (either a pixel or a tile) in the target image. Each matching algorithm produces a scalar similarity value at each location by taking a normalized weighted sum of a vector of scalar similarity values defined by the matching algorithm. For example, in the ICE model-based phase-sensitive matching algorithm, at a given query image pixel location and target model orientation, the similarity is arrived at by summing the cosine-squared of the phase differences over all projected model pixels. In query-by-example, a weighted sum of the previously described feature vectors defines the similarity.

In model-based search, we improve the performance of the matcher by examining the shape of the local maxima peaks in the match surface. For example, searches for high aspect ratio rectilinear objects tend to produce strong local maxima in areas such as plowed fields that have repeated linear structures. These repeated structures will be reproduced in the match surface. If isolated instances are expected, as is usually the case, these repeated structures can be rejected so that the true matches float to the top. Likewise, if multiple instances are required (e.g., a certain vehicle is only interesting when multiples are present), then isolated peaks in the match surface can be rejected.

The current relevance feedback approach utilizes the exponential weight updating algorithm described in [6], in which new weight calculations are based upon ratios of individual feature performance to the overall feature vector performance. It is found that this algorithm is relatively effective when the percentage of relevant hits is low, as is often the case in broad area search problems.

6. CONCLUSION AND FUTURE DIRECTIONS

We have given a brief overview of the ICE system being developed at LLNL to support broad area search in large image databases. The system has proven to be an effective tool for search based on high-fidelity three-dimensional target models. Its capability to do searches based on sets of example images has demonstrated effective performance in initial test applications.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory, under contract No. W-7405-Eng-48.

REFERENCES

1. D. Paglieroni, W. Eppler and D. Poland, "Phase Sensitive Cueing for 3D Objects in Overhead Images", *Sensor Data Exploitation and Target Recognition*, Proc. SPIE Defense & Security Symposium, 5809, Orlando, Florida, March 2005.
2. S. Antani, R. Kasturi, and R. Jain. "A survey on the use of pattern recognition methods for abstraction, indexing, and retrieval of images and video", *Pattern Recognition* 35, 945, 2002.
3. Y. Rui, T. Huang, and S. Chang. "Image Retrieval: Current Techniques, Promising Direction, and Open Issues", *J. of Visual Comm. and Image Representation* 10, 39, 1999.
4. N. Chang and K.S. Fu. "Query-by-Pictorial-Example", *IEEE Tr. Software Engineering* SE-6, 519, Nov. 1980.
5. H. Greenspan, R. Goodman, P. Perona, S. Rakshit, and C. Anderson. "Overcomplete steerable pyramid filters and rotation invariance". *ICCV*, 222, 1994.
6. Z. Jin, I. King and X. Li, "Content-Based Image Retrieval by Relevance Feedback", *Proceedings to the Fourth International Conference on Visual Information Systems (VISUAL2000)*, Lyon, France, November 2000.