UCRL-SR-229741

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Image Content Engine (ICE)

J.M. Brase

April 5, 2007

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.

Auspices Statement

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

LDRD Auspices Statement

This work was performed under the auspices of the U.S. Department of Energy (DOE) by the University of California, Lawrence Livermore National Laboratory (LLNL) under Contract No. W-7405-Eng-48. The project 03-SI-003 was funded by the Laboratory Directed Research and Development Program at LLNL.

FYO6 LDRD Final Report
Image Content Engine (ICE)
03-SI-003
James M. Brase, Principal Investigator

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.

1. Introduction and Background

This report describes the work performed in the Image Content Engine (ICE) Strategic Initiative In Sections 1 and 2 we give an overview of the technical work and results obtained in the project. More detailed results are given in the attached papers Appendices 1- 6. Section 3 summarizes our exit strategy and on-going work resulting from ICE.

1.1 Defining the problem

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 national security and intelligence communities are facing a crisis in their inability to analyze massive volumes of remotely sensed imagery. Nearly every aspect of national security and intelligence progress is tied to the acquisition and interpretation of images. With growth in sensor and computing technologies, the capability to generate images is expanding exponentially, but the ability to extract understanding from images is not. There have been major investments over the last decade in new and more capable imaging sensors:

- A new generation of reconnaissance satellites (part of the Future Imagery Architecture) will expand the capacity to collect images by orders of magnitude.

- New platforms, such as UAV's, support collection of real-time video, which can potentially expand the volume of imagery by further orders of magnitude.

- New classes of imaging sensors, such as hyperspectral cameras and new synthetic aperture radars, provide data that is much more difficult for human analysts to interpret.

The R&D investment in extracting knowledge from massive volumes of incoming imagery has not kept pace with the R&D investment in new imaging sensors and sensor platforms. Human analysts are already overwhelmed - much of the imagery that is currently acquired never gets analyzed. This problem will get much worse as new sensor systems go into routine operation. A number of congressional studies have concluded that the existing R&D base in this area is inadequate:

> *"The Commission is convinced that woefully inadequate R&D holds hostage the future success of TPED (Tasking, Processing, Exploitation and Dissemination), the U.S. Imagery and Geospatial Service, and indeed of U.S. information superiority."*
>
> - *From a report of the Independent Commission on the National Imagery and Mapping Agency (NIMA), December 2000.*

Even if many more human analysts were available, the tools currently available would not provide sufficient assistance to allow them to assimilate massive volumes of image data in a time critical fashion. As the images increase in size and complexity, the tools needed to find patterns and relationships must extend to new types of images, analysis of spectral and temporal patterns (not just spatial patterns), and multiple disparate sources of information (these are used to help guide the analysis process).

The scientific community is facing the same set of challenges. New imaging systems like the Large Synoptic Survey Telescope (LSST) will produce massive data streams that must be processed in real-time and large-scale databases that will reach multiple petabytes in size. Tools that can handle image data at these rates and sizes do not exist today.

1.2 What is ICE ?

The mission of the *Image Content Engine (ICE)* Strategic Initiative (SI) is to develop a framework of software and underlying mathematical and physical models that will enable scientists and analysts to search massive volumes of complex imagery for content relevant to their missions and applications. Our specific objective is to develop tools for real-time analysis of streams of images and to search a large database images based on image content. The tools extract information from the raw image data and represent it in general index or catalog forms that allow direct queries. The top-level architecture of these systems is shown in Figure 1.1.

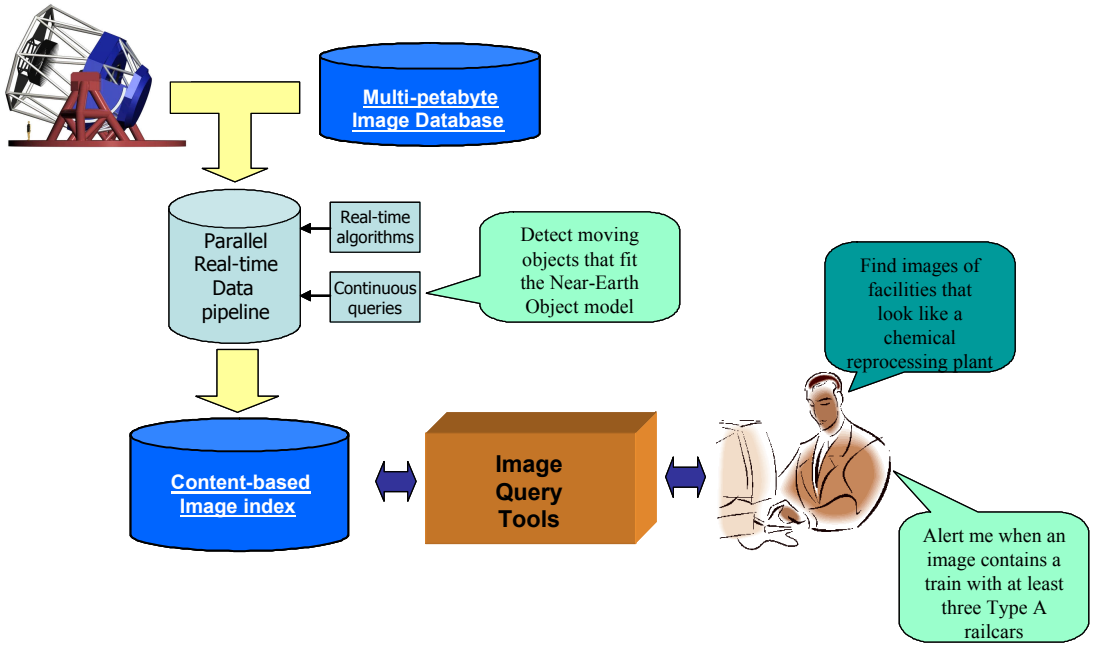

Figure 1.1 Top-level architecture of ICE systems. We are developing tools for real-time analysis of image streams for large-scale sensors and for analysis of large image databases.

The basic approach in ICE is to build a hierarchical model of image content that describes images at three levels: features, objects, and graphs that define spatial relationships. This set of image models forms an index of the image database or stream that can be efficiently searched by a scientist or analyst for content relevant to their query. ICE will deliver the algorithms and software to build these models. We will demonstrate them on large-scale parallel computer systems (e.g. MCR) to establish their scaling to very large data sets. We will deliver query tools that low us to search the image models at all three levels and we will characterize the query performance in several LLNL programmatic and science applications.

To make these capabilities more concrete, imagine a query like "Find all the facilities in Country A that have are consistent with our spatial model of a chemical reprocessing plant." A query like this would currently require human analysts to manually retrieve images of buildings at suspect sites and visually cross-correlate with a facility model. This search could take months for a full country. The goal of the ICE SI is to handle a query like this automatically — defining the relevant image features, bringing in data from other sources, and presenting the analyst with guidance on where to look to make final determinations.

ICE is enabled by the tremendous progress made in the last ten years in computational image models and image analysis algorithms. Emerging understanding of human visual processes, new knowledge management techniques based on large semantic graphs, efficient new approaches to machine learning, and new capabilities to acquire and analyze image sequences will be brought together in ICE. The ability to use multiple levels of knowledge representation in an image is an important new aspect of ICE. This approach allows us to use partial semantic descriptions for a scene but also to drop back to a more image-based model when we cannot segment and detect objects with confidence. In this approach, perfect image segmentation and object detection is not required for useful image retrieval capability.

2 Research Activities

The main components of an ICE system are shown in Figure 2.1. They start with a source of images that can be either a real-time source of an image stream (e.g. the LSST camera) or a pre-existing database of images. The heart of the system is a hierarchical set of image models that are computed from the set of images in a real-time image analysis pipeline. The models effectively serve as a high level index of the content of the set of images. Finally, interactive object detection and query-by-example modules can extract information from the set of image models to respond to specific queries or detection models. The overall architecture and implementation is described in more detail in the first attached paper [1].

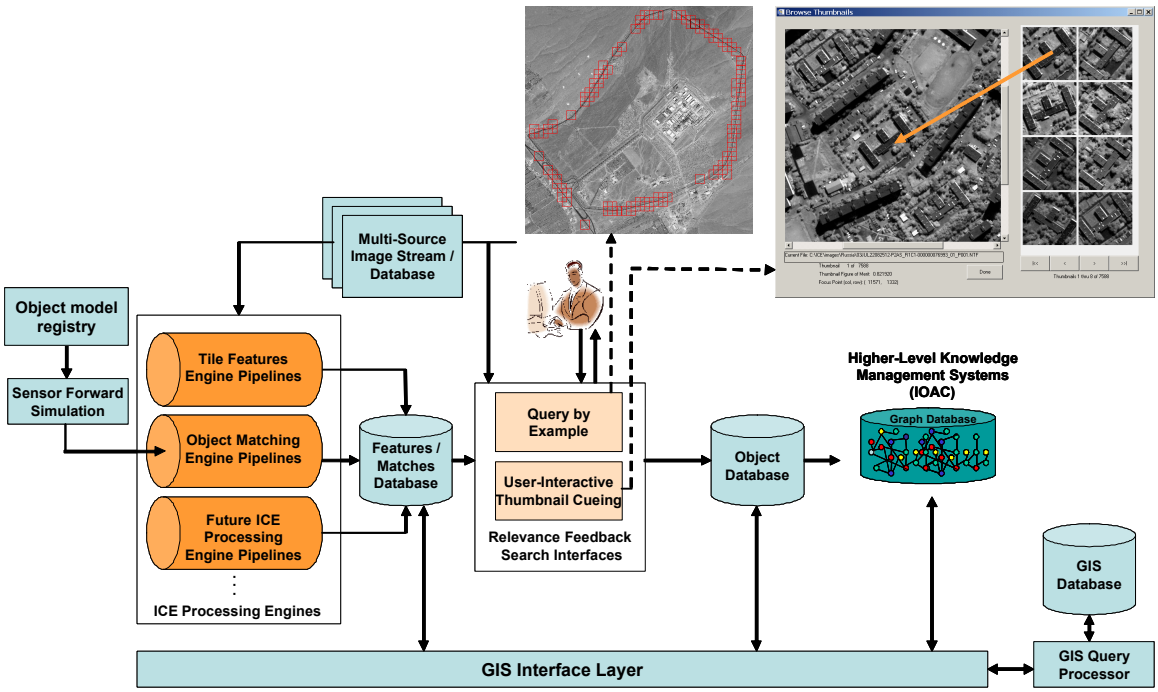

Figure 2.1. The high-level conceptual design of an ICE image retrieval system.

2.1 Use Cases and applications

In this section we will describe some of the typical scenarios for the use of ICE tools. These examples describe the ways in which we currently envision ICE being used but the tools will be developed so that they can be very flexibly adapted to different scenarios. The common thread throughout the applications is the requirement to build image models that allow us to search large sets of image data. The applications discussed here are being used to drive the R&D activities in the SI.

1. Train detection

   The application is to search for a particular type of railroad train. The trains can be uniquely characterized by a particular combination of railroad cars. We start with a set of three-

dimensional models of the particular rail cars of interest. These models can be used to estimate the image signature of the cars which, in turn, allows us to search the database. We can also identify railroad tracks which are useful for suppressing false alarms – if the cars are not located on or near a track it is likely a false alarm. We can then build graphs of the types of cars detected in the trains allowing us to search for the particular combination of cars that defines the target. This application requires capability in efficient model-based object detection and in graph construction and matching.

2. Enigma facility search

In this application we want to search a large set of imagery for a particular type of facility. An analyst may have discovered a new facility in Country X. The function of the facility is not clear but its construction has a distinctive set of structures and relationships. We want to search the country for other examples of this arrangement. If we can find related facilities they may give us clues to the enigma facilities functionality. The known site can be used to construct a hierarchical spatial model which can be used to efficiently search the large set of images.

3. Searching a set of tomography images

Another important class of application in a different component of LLNL's mission is the searching large sets of x-ray tomography images of weapons components. If a flaw has been discovered we can search the images to see if similar flaws have gone unnoticed in previously inspected components. In this application we would build a model of the flaw based on the discovered image and then match it against image models in the database. For a database containing thousands of large three-dimensional images searching by a human would be a daunting and error-prone task.

4. LSST image analysis

In the LSST application we classify objects such as stars, galaxies, quasars, etc. and detect transient events and moving objects. The main challenge in this application is to do these detection and object classification problems on a multi-gigapixel/sec dynamic data stream within a 30 second time response requirement.

2.2 Hierarchical image models

A central idea in ICE is the development of a hierarchical model of an image. This model provides an index into the content of the image (Figure 2.1). It serves to focus search strategies on relevant content. The model can compress the information to make searches more efficient; it can also be invariant to image differences like shifts, scale changes, etc. that do not affect content. We have defined three general levels of abstraction for image models:

- Image feature level – At the lowest level we compute spatial maps of image features. Each feature is computed from a localized region of the image called a tile. A number of different features can be computed for each tile; examples include intensity or color statistics or responses to directional or edge filters. When the features computed at each tile location are collected we call this set a feature vector.
- Object level – Collections of image features that match an external model are referred to as an object. Objects are detected by comparing image feature maps with the feature distribution computed for the object model and then applying a decision rule or threshold.
- Graph level – Graphs provide a natural approach to representing collections of objects or selected feature vectors and their spatial relationships. Graphs provide a generalization of spatial relationships and a natural way of linking spatial models to other knowledge types.

2.3 Model- based detectors and queries

When we search a set of images for a specific object or set of objects, the search is based on an external model of the target object. The model is created externally to ICE and defines the three-dimensional structure of the target object. To search the image set the target model is projected into the image space using a model of the sensor and environment. For example, the model can include the angular pose of the target, the illumination angles, the imaging wavelength and many other parameters depending on the level of fidelity of the model. The model project gives us a target signature – the image space representation of the target model.

The signature is computed in terms of the hierarchical model components described in 22. To search the image the target signature is compared with the set of image feature maps and the similarity (or detection statistic) at each spatial location is compared to a decision rule or threshold. This threshold parameter can be varied interactively by an analyst to set the appropriate detection rates and false alarm rates. This interaction is very fast when the spatial detection statistic map is already computed. This detection process results in a set of detection locations which we refer to as an object table. The object table can be directly queried or used to build higher-level graph representations.

2.3.1 2D Signature Projections

A *spatial signature* is a 2D representation of how an object is expected to look in an image. 2D spatial signatures are critical to the task of matching and detecting (generalized) objects in images. In ICE, spatial signatures only address object edges, since object edges tend to be relatively insensitive to the type of sensor and image acquisition conditions.

In certain limited cases, the structural appearance of a 3D object does not vary noticeably as a function of location or orientation within an image. This might be the case, for example, in ortho-rectified images of fixed spatial resolution. In this case, an object spatial signature can be derived directly from a similar sample image by using a mouse to trace its apparent edges.

However, it is more typically the case that the structural appearance of a 3D object varies both as a function of location and orientation within an image. For example, in oblique images, objects in the distance appear smaller than objects that are close in. In this case, object spatial signatures must be generated from 3D models on an image block-localized basis at each of several orientations. In remotely sensed images, the traditional way to accomplish this is to use

the *rational polynomials* supplied with the image to project the edges of the 3D model onto the image block. It is important to project the object at each of several orientations and to remove hidden lines (occluded edges) from the projection. The rational polynomials are functions of the [*x, y, z*] = [longitude, latitude, altitude] coordinates of the object model edge points offset by the geographical location of the center of the image block. A rational polynomial contains numerator and denominator polynomials, both of which are polynomials in [*x, y, z*] (typically of order 3 or less) with cross terms. Each image has two rational polynomials, one for pixel column and one for pixel row coordinates as a function of [*x, y, z*].

2.3.2 Phase Sensitive Matching in the Fourier Domain

The goal of phase sensitive matching for object detection is to efficiently detect 3D objects in images regardless of sensor type or acquisition conditions. In the context of an edge boundary in a 2D signature, the phase at a boundary pixel refers to the direction normal to the direction of boundary flow at that pixel. In the context of an image, the phase at an image pixel refers to the direction of flow in spectral energy at that pixel. Phase sensitive matching seeks similarity between the direction of boundary flow at pixels along signature boundaries and the direction of flow in spectral energy inherent in the corresponding image pixels. In the spatial domain, phase similarity can be simply expressed as

$$S(\Delta_c,\Delta_r) = \frac{1}{N_s} \sum_{[c,r] \in B} cos^2[\theta_s(c,r) - \theta_I(c+\Delta_c, r+\Delta_r)] \in [0,1] \tag{2.1}$$

where S denotes "similarity", N_s is the number of signature pixels with [*column,row*] coordinates [*c,r*] on the signature boundary B, $\theta_s(c,r)$ and $\theta_I(c,r)$ are signature and image phase at pixel [*c,r*], and [Δ_c,Δ_r] is the [*column,row*] offset of the signature within the image block. It is computationally expensive to evaluate expression (2.1) at each signature offset and orientation within an image block. However, $S(\Delta_c,\Delta_r)$ can be re-expressed as a function of 2D spatial correlations between real and imaginary parts of complex phase images, and these 2D correlations can be efficiently evaluated using the FFT. Fig.2.2 depicts phase sensitive matching in the Fourier domain, where $\boldsymbol{\Theta_I} = \{\Theta_I(c,r) = e^{j2\theta_I(c,r)}\}$ is *wxh* complex image with subscripts "*r*" and "*i*" indicating real and imaginary parts. Similarly, $\Theta_{Sn}(c,r) = a_n(c,r)e^{j2\theta_{Sn}(c,r)}$, where n is the signature orientation index and $a_n(c,r) = 1$ at signature edges (and 0 otherwise).

The output of phase sensitive matching on an image block is a 2D array $\boldsymbol{S}$ of match similarity values. The vast majority of the matches are ambiguous. ICE uses a novel disambiguator to find the unambiguous matches (i.e., those whose signatures do not overlap signatures of other matches

of greater similarity).

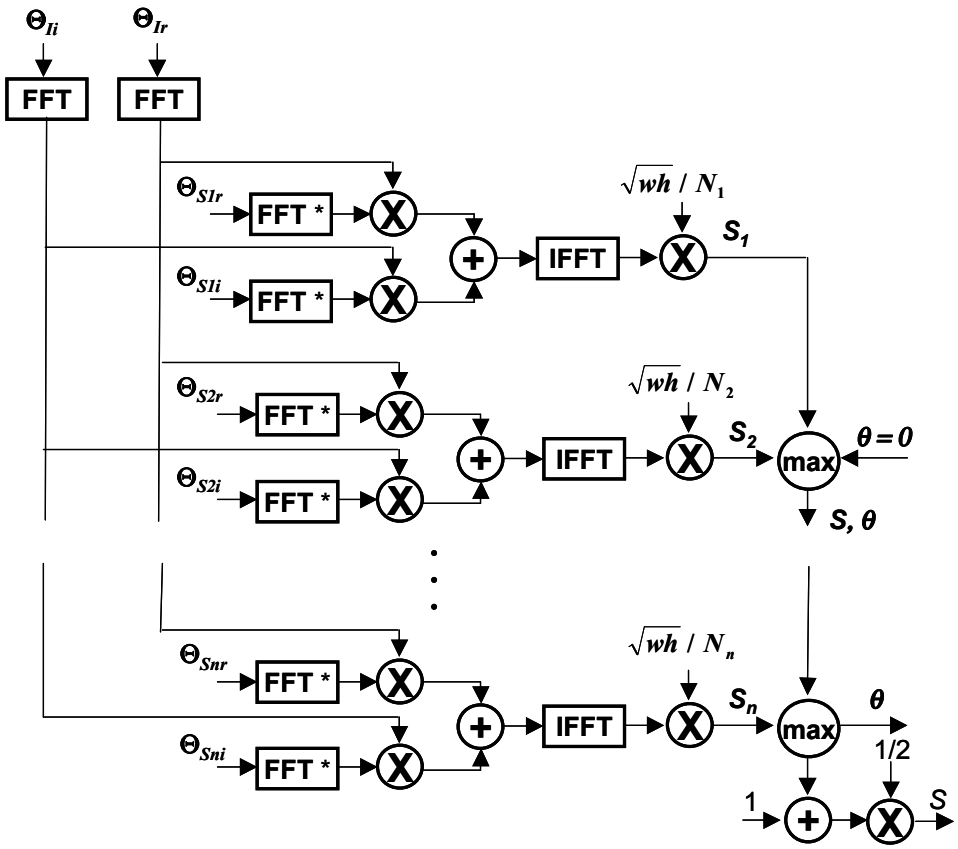

Fig.2.2 Block diagram for (unitary) FFT-based phase sensitive matching.

2.3.3 Status of Phase Sensitive Matching

In FY05 we completed work on the algorithm development, software implementations, and performance testing for the phase sensitive image search algorithms. An example result of a test search is shown in Figure 2.3. Users are able to build three-dimensional tarfet models, specift sets of images to be searched, and run the feature extraction and model matching algorithms on a high-performance parallel compute cluster. The analyst can then interactively query the set of model match points that were detected in the image to eliminate false alarms. We have applied phase sensitive image search to a selected set of programmatic problems. In one case the ICE search algorithm was able to find >90% of the targets found by exhaustive human analysis and do it in < 1/10 the time.

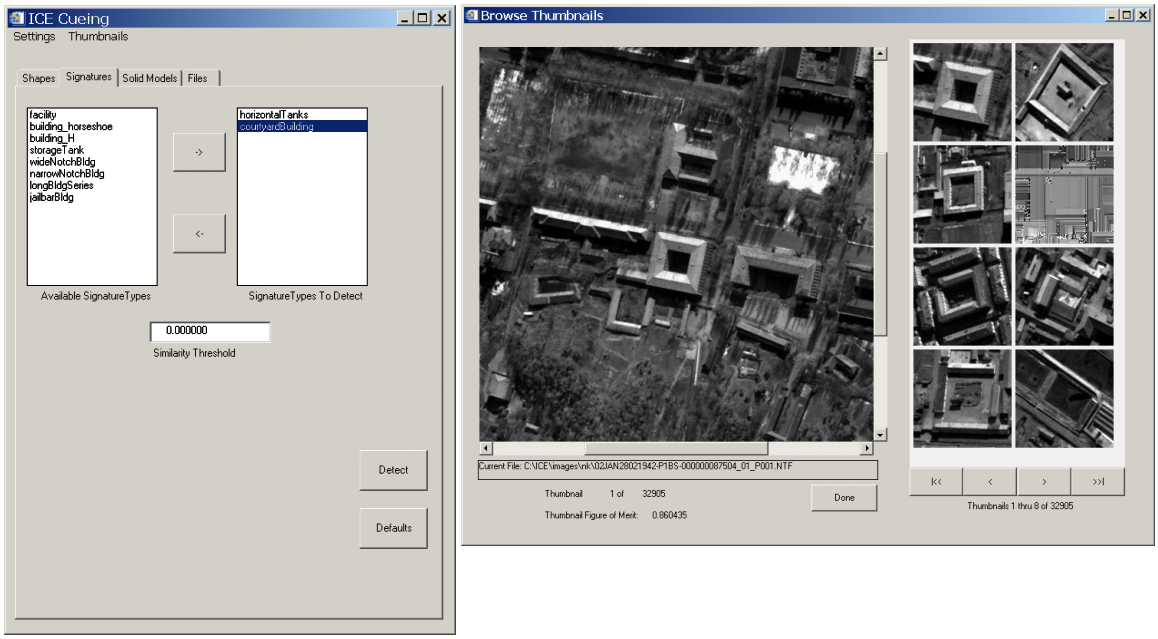

Figure 2.3 An example of a phase sensitive image search in ICE. The search target was a model of a building with a courtyard. The small thumbnail images on the right show the first set of returned results. The first image is the building used to build the model. Subsequest results show a set of building with similar geometries that were found throughout the search space.

2.4 Query-by-example

One of the most common methods of specifying an image database query is through a set of example images. The user selects a set (one or more) of images that represent the class of objects or patterns of interest. We then wish to search the database for all the instances of data that also belong to this class. A schematic view of the process is shown in Figure 2.4.

When we have multiple example images we can establish the relative importance of the various components of the hierarchical feature model by analyzing the variance of the features over the set of examples. The variance estimates allow us to weight the search more heavily on those features that are consistent throughout the example set.

Relevance feedback is a technique for iteratively improving query performance by increasing the size of the example set. At each iteration the user selects the new images that they believe satisfy the search criteria. These images are used to update the weights described above. An example of query-by-example with relevance feedback is shown in Figure 2.5. In this case the tiles database being searched was the result of an earlier model-based search for a particular class of building. Searching that first-order search result at a more detailed level results in a significant decrease in the number of false detections of the targeted building.

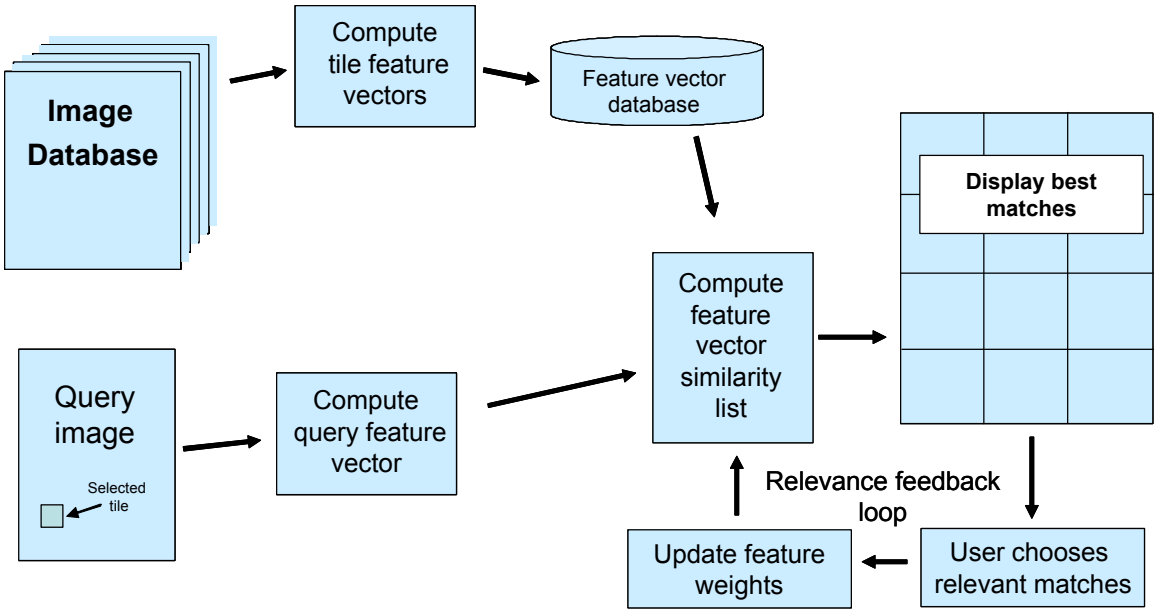

Figure 2.4. Structure of the ICE query-by-example processor. Tile feature vectors are computed through the ICE pipelines discussed above. A query image is selected and query vectors are computed. The query feature vector is compared to all feature vectors from the image database and the most similar are displayed for the user. The user can provide relevance feedback to the selection algorithm to improve performance.

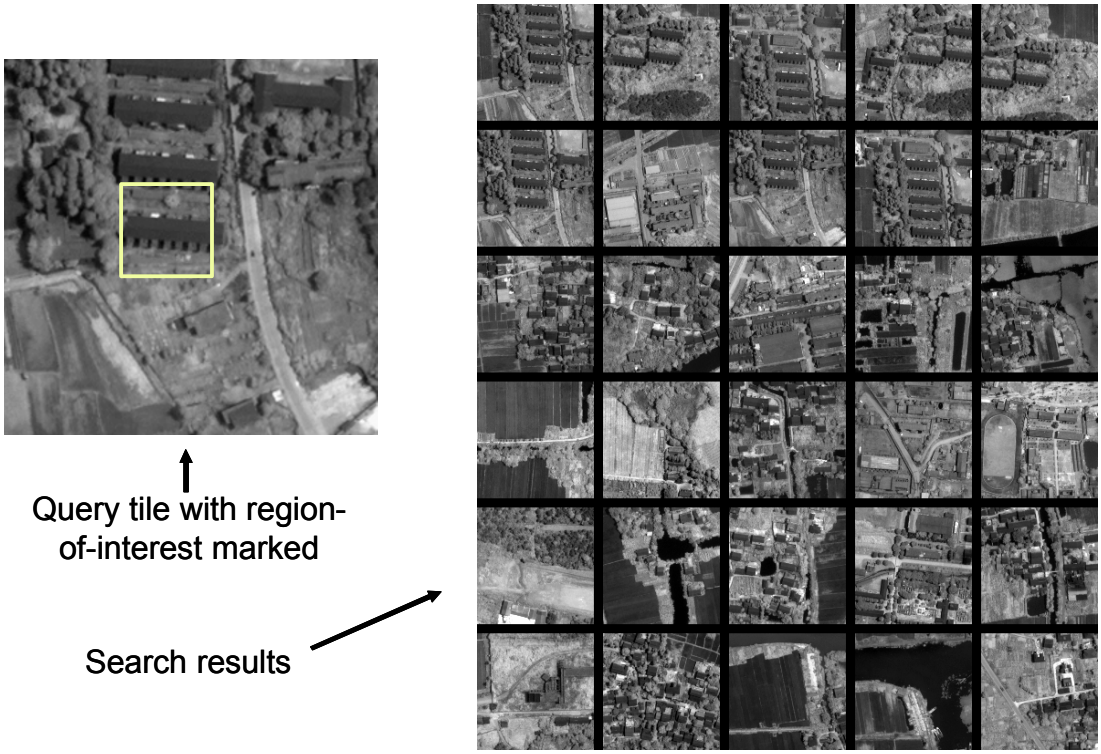

Figure 2.5. Results of query-by-example with two iterations of relevance feedback. Best matches are at the top-left. The first nine tiles are good matches.

2.5 Real-time image stream architectures

ICE addresses image analysis problems that are centered on extremely large data sets. To work effectively with these data sets we must develop hardware/software architectures that can exploit the spatial and algorithmic parallelism implicit in these imaging applications. In this section we will discuss the design and development of real-time image stream architectures that provide feature extraction, object detection and classification, and temporal change detection under strong time constraints. These architectures will be applied to tile and object-level processing and to prototyping the data analysis pipelines for LSST. The cluster structure being used for both ICE and LLST development is shown in Figure 3.1. The software implementations being developed will be portable to the large cluster systems available to us for testing.

2.5.1 Feature extraction pipelines

The ICE architecture creates image content by extracting features from images. Generally, the features extracted belong to one of three groups: region-based shapes, tile-based feature vectors, and model-based objects. Because practical considerations (available physical memory, processor cache sizes, etc.) do not, in general, allow an entire image to be processed in its entirety, an image is broken into *blocks* prior to processing.

Blocks consist of square sub-regions of an image, and (depending on the type of content being extracted) may overlap one another. The block size is chosen based on processing efficiency and speed considerations, as well as the size of the content that is being extracted. The block overlap is determined by the size of the content being extracted, so that the desired object is guaranteed to exist in its entirety in at least one of the overlapping blocks. Content that is duplicated in the overlapping areas between two blocks (i.e., extracted twice) is dealt with after the fact.

Since the block sizes and overlaps need to change based on the content being extracted, multiple feature extraction pipelines must exist, one per unique block size and overlap. In the case of region-based shapes and tile-based feature vectors, it is assumed that one suitable block size and overlap may be chosen for each. However, in the case of model-based objects, analysis indicates that a separate feature extraction pipeline must be created for each desired object model.

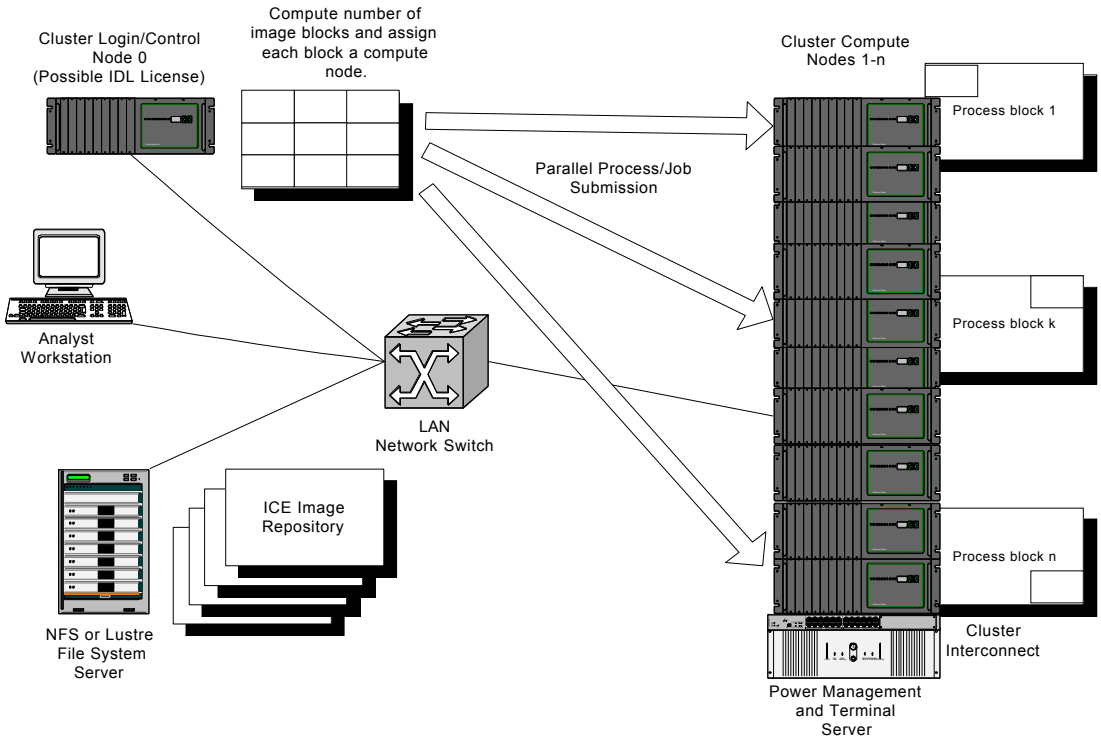

Figure 2.6 General computer cluster approach to parallelizing ICE pipelines.

Although each of the three types of pipelines (region-shapes, tile vectors, and object models) must perform different types of processing, they share some similarities. In each case the pipelines assume that they are being given a block from an image, rather than the entire image. The image is broken into blocks prior to being sent to the various pipelines. Each pipeline performs image pre-processing steps, designed to ensure more consistent results – range and quantization adjustments, for instance.

After pre-processing, the pipelines actually perform the compute intensive feature extraction calculations: segmentation, feature vector calculations, and model-based object matching.

The final stage may be considered post-processing. This is the point where similarity values are calculated, and, if necessary, the data generated in the previous step is cleaned up.

The data produced at this point do not yet constitute *detections*, rather they are potential detections. This data will still need to be gated through various detection thresholds and decision criteria before it is considered content.

Given the above description, it is straightforward to see how a cluster of computing resources may be applied to the feature extraction problem. First, the total number of pipelines required must be determined. Next, for each of the pipelines the number of blocks, as well as their sizes and overlap, must be determined. With that information in hand, one CPU may be assigned the task of extracting the features for a single block and a single feature type. Feature extraction tends to be compute bound, rather than IO bound, making this an attractive solution.

After the features have been extracted from each block, the data generated must be reassembled into a data set representing the entire image. This is a straightforward task, which does not involve either a great deal of computation or IO, relative the feature extraction calculations.

An illustration of the above method of using a computational cluster is shown below. If, for example, an 8192x8192 pixel image is broken into an average of 256 blocks (512x512), and we are looking for 10 different model-based objects (in addition to the segmentation and tile pipelines), we will need 12 pipelines total, and could productively make use of up to 3073 (12*256+1) CPUs.

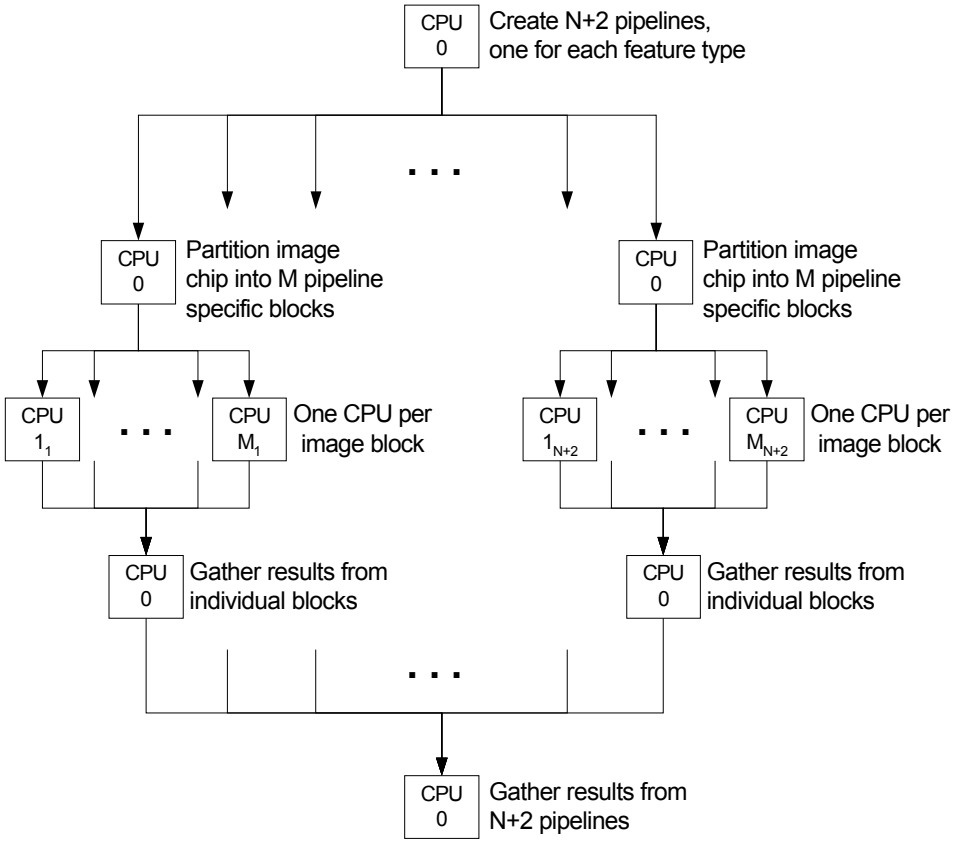

Figure 2.7. Block-parallel approach to implementing high-performance implementations of ICE pipelines on parallel computer clusters.

In FY05-06 we completed initial implementation of a parallelized ICE pipeline. In initial tests the system demonstrated more than 60% of the theoretical maximum speedup for the processor cluster being used. We have also completed initial implementation and testing of a prototype pipeline for the Large Synoptic Survey Telescope. In this application we have parallelized a SuperMacho pipeline that does image conditioning and difference image detection of moving or varying objects. In initial testing, the parallelized pipeline design has demonstrated nearly linear speedup as the number of processors increased (Figure 2.8).

- Used existing ICE test scripts for benchmarks.
- On average we process 2 image blocks per second on 32 cpus.
- Speedup over running code on a single processor cpu is 18x to 22x.

- nk_nitf: 3802 blocks in 29 minutes.
- china_barge: 1062 blocks in 8 minutes.
- Russia_3prongbuild: 529 blocks in 5 minutes.

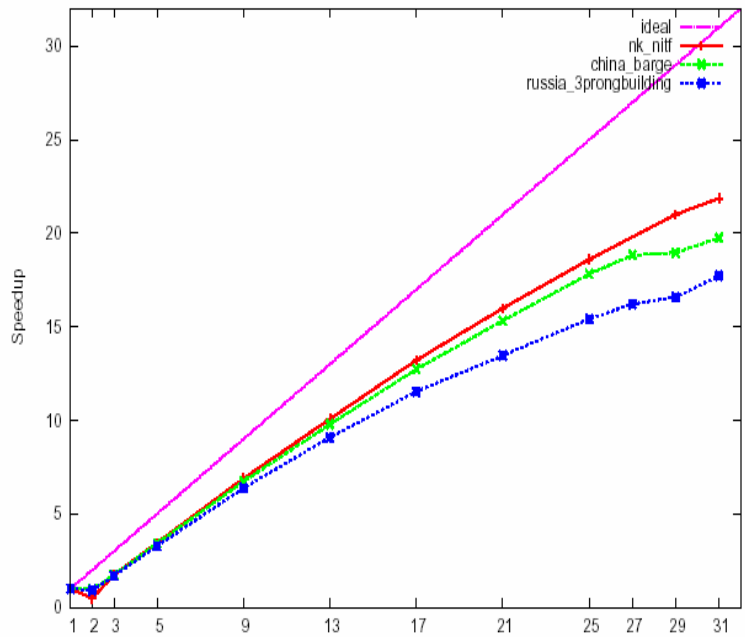

Figure 2.8. The parallelized ICE image processing pipeline demonstrates nearly linear speedup with processor number in realistic image search applications.

3 Exit plan

LLNL has developed a vital and active R&D effort in image analysis and real-time image stream architectures. The science and technology developed in ICE has applications in programs throughout LLNL such as ASCI simulation analysis, microscopy for biological weapons analysis, NIF diagnostic imagery analysis, and non-destructive inspection for the weapons program. We have actively engaged NIF and the weapons program on new applications of ICE algorithms and technology for optics inspection and for analyzing computed tomography images.

However, the main long-term objective of ICE has been develop a self-sustaining effort in image analysis R&D for the national security and science communities at LLNL. Our strategy for sustaining work in this area has two main components. First, the ICE framework and associated tools will have a variety of *internal applications* to long-term programs at LLNL within NAI and other directorates. For example, ICE will enhance Z Division capabilities to engage in non-proliferation activities related to analysis of data from intelligence imagery and collateral sources. ICE will contribute to development of remote sensing technologies in Q Division by helping to validate the utility of new sensors to human analysts. ICE will feed new capabilities back into the IOAC Program. This strategy has been successful. ICE tools are being used for program applications in Z Division and are helping to bring new funding to LLNL for operational intelligence analysis.

In addition, ICE will have *external applications* that will allow LLNL to develop new programs with sponsors in DOE, DOD and the intelligence community. By leveraging connections in NAI (notably from Q and Z Divisions) to the broader community, ICE will interact with prospective

sponsors, provide demonstrations of capability and participate in community-wide exercises, with the goal of demonstrating utility to human analysts. Prospective sponsors internal to the DOE include NNSA. Prospective sponsors external to the DOE include NGA, STRATCOM (J2, the JIC) and CMO. These interactions are very promising. As noted above we have attracted external funding for operational analysis using the capabilities developed under the SI. We are working to develop sponsors in DOE and the IC who be the longer-term sponsors of the continuing R&D that is necessary to keep the effort alive and moving forward.

4 Summary

As imaging sensor capabilities and applications grow the ability to manage and search large sets of images is vital to fully realizing their potential. ICE has developed a strong capability in this area. It has contributed new algorithms and computational methods to advance image analysis science and technology. The capabilities developed in ICE are being directly applied at LLNL in nonproliferation and intelligence programs as well as in NIF and DNT applications.

5 References

J.M. Brase, D.W. Paglieroni, D.N. Poland, G.F. Weinert, C.W. Grant, A.S. Lopez and S. Nikolaev. "Image Content Engine (ICE): A System for Fast Image Database Searches", SPIE Defense & Security Symposium: Optics and Photonics in Global Homeland Security, Proc. SPIE, Vol. 5781, 28-30 March 2005, Orlando, FL.

D.W. Paglieroni, W.G. Eppler, and D.N. Poland. "Phase Sensitive Cueing for 3D Objects in Overhead Images", SPIE Defense & Security Symposium: Optics and Photonics in Global Homeland Security, Proc. SPIE, Vol. 5781, 28-30 March 2005, Orlando, FL.

B.Y. Chen and D.W. Paglieroni. "Using Gradients, Alignment, and Proximity to Extract Curves and Connect Roads in Overhead Images", SPIE Defense & Security Symposium: Optics and Photonics in Global Homeland Security II, Proc. SPIE, Vol. xxxx, 17-21 April 2006, Orlando, FL.

C.W. Grant, S. Nikolaev, and D.W. Paglieroni. "Weighted Model Components for Gradient Direction Matching in Overhead Images", SPIE Defense & Security Symposium: Optics and Photonics in Global Homeland Security II, Proc. SPIE, Vol. xxxx, 17-21 April 2006, Orlando, FL.

W.M. Miller, J.E. Garlick, G.F. Weinert, G.M. Abdulla, "Exploiting Data Parallelism in the Image Content Engine", SPIE Defense and Security Symposium, SPIE Vol. xxxx 17-21 April 2006, Orlando, FL.

D.W. Paglieroni and W.G. Eppler. "Resolution Analysis for Gradient Direction Matching of Object Model Edges to Overhead Images", submitted to International Journal of Computer Vision, 2006.