RL-TR-96-290
Final Technical Report
April 1997

IMAGE UNDERSTANDING FOR DATABASE QUERY

PAR Government Systems

Sponsored by
Advanced Research Projects Agency
ARPA Order No. A931

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-96-290 has been reviewed and is approved for publication.

APPROVED: Peter J. Costianes

PETER J. COSTIANES
Project Engineer

FOR THE COMMANDER:

JOSEPH CAMERA, Technical Director
Intelligence & Reconnaissance Director

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/IRRE, 32 Hangar Rd., Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

IMAGE UNDERSTANDING FOR DATABASE QUERY

Contractor: PAR Government Systems
Contract Number: F30602-94-C-0050
Effective Date of Contract: 3 September 1994
Contract Expiration Date: 30 September 1996
Program Code Number: 62301E
Short Title of Work: Image Understanding for Database Query
Period of Work Covered: Sep 94 - Sep 95

Principal Investigator: Dr. Charles Ferrara
Phone: (315) 330-3896
RL Project Engineer: Peter J. Costianes
Phone: (315) 330-4030

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by Peter J. Costianes, RL/IRRE, 32 Hangar Rd, Rome, NY.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE April 1997	3. REPORT TYPE AND DATES COVERED Sep 94 - Sep 96

4. TITLE AND SUBTITLE IMAGE UNDERSTANDING FOR DATABASE QUERY	5. FUNDING NUMBERS F30602-94-C-0050 PR - 4594 PE - 62702F 62301E TA - BA36 WU - 998065
6. AUTHOR(S) Dr. Charles Ferrara, PhD, Mr. Scott Barrett	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) PAR GOVERNMENT SYSTEMS 8383 SENECA TURNPIKE NEW HARTFORD NY 13413	8. PERFORMING ORGANIZATION REPORT NUMBER N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency, 1701 North Fairfax Drive, Arlington VA 22203-1714 Rome Laboratory/IRRE, 32 Hangar Road, Rome NY 13441-4114	10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-96-290

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Mr. Peter J. Costianes/IRRE/x4030

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The objective of the Image Understanding for Database Query (IU4DBQ) effort was to combine the powerful reasoning capabilities of the Loom knowledge representation system with the extensive image processing and feature extraction capabilities of KB Vision TM. Another important aspect of this effort was to apply the algorithms developed to realistic reconnaissance imagery. A major challenge was to integrate Loom and KB Vision TM within a useful context for the softcopy reconnaissance imagery available in the Rome Laboratory (RL) IE 2000 facility. The most practical use for Loom is to query the imagery server databases and determine images that have specific characteristics that are most suitable for processing. This suitability is based on the target area of interest, the presence of unconfirmed targets, and the quality of the imagery.

14. SUBJECT TERMS knowledge representation, machine vision, feature extraction	15. NUMBER OF PAGES 28 16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UNLIMITED

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The objective of the Image Understanding for Database Query (IU4DBQ) effort was to combine the powerful reasoning capabilities of the Loom knowledge representation system with the extensive image processing and feature extraction capabilities of KBVision™. Another important aspect of this effort was to apply the algorithms developed to realistic reconnaissance imagery. A major challenge was to integrate Loom and KBVision™ within a useful context for the softcopy reconnaissance imagery available in the Rome Laboratory (RL) Imagery Exploitation (IE) 2000 facility. The most practical use for Loom is to query the imagery server databases and determine images that have specific characteristics that are most suitable for processing. This suitability is based on the target area of interest, the presence of unconfirmed targets, and the quality of the imagery.

The issue of image quality is multi-faceted. If an image is of very high quality, a photo-interpreter (PI) is usually capable of analyzing it without computer assistance[1]; however, in such cases processing can still facilitate the PI's task, by pointing out items of interest that are potential targets. Slightly lower image quality is more difficult for a PI and there may be unconfirmed targets. Such an image could still have sufficient quality for automatic processing. If the image quality is lower still, the automatic processing is likely to have difficulties and there would be little value in attempting to use such images.

The selected images are processed by the KBVision™ subsystem using the pixel dimensions of the targets. The crucial information needed to do this are the absolute target dimensions and the size of the pixel footprint. Using this information, KBVision™ returns a set of segmented regions (tokens) that potentially contain targets. Loom then adds these tokens to the database. Although Loom does have recognition capabilities of its own, these capabilities were not used in this effort.

Figure 1 illustrates the overall system and how the various components interact. As the figure indicates, Loom is the core of the IU4DBQ system. It is responsible for interacting with the user, the imagery database, the target database, and the KBVision (KBV) subsystem. Chronologically, the following happens:

1. The user selects a target type and a location in the world where he/she would like to find the targets. For the demonstration, the target type is predetermined. An interface is currently not in place to permit the user to select the target type. The user specifies the imagery database to be used. A shell script must be run on the desired imagery

[1] This depends on the size of the target. The critical factor is the number of pixels on a target.

database before it can be used by IU4DBQ. The shell script creates some metadata files containing information from a CATIS or IESS (Combat Aided Tactical Information System, Imagery Exploitation Support System) report.

2. Loom queries the imagery database for images that contain the target of interest at the specified geographic location(s). The query can also use NIIRS ratings or the presence of unconfirmed targets. The images of interest are returned as well as the pixel footprint size. The demonstration is based on NIIRS rating, sensor type (optical), and lack of confirmed targets.

3. Loom queries the target database to determine the target dimensions; then the target pixel dimensions are computed for each image.

4. Loom passes the images and target information to the KBV Subsystem. Potential target locations are returned along with feature information about these regions.

5. Information about targets and their potential locations are passed back to the user. In the demo, the tokens resulting from the KBV task are shown overlayed on the original image. Each of the tokens is added to the set of items Loom knows about, labeled as unconfirmed instances of the target type found by KBV.

6. A PI uses this information as a decision aid for determining actual target locations.

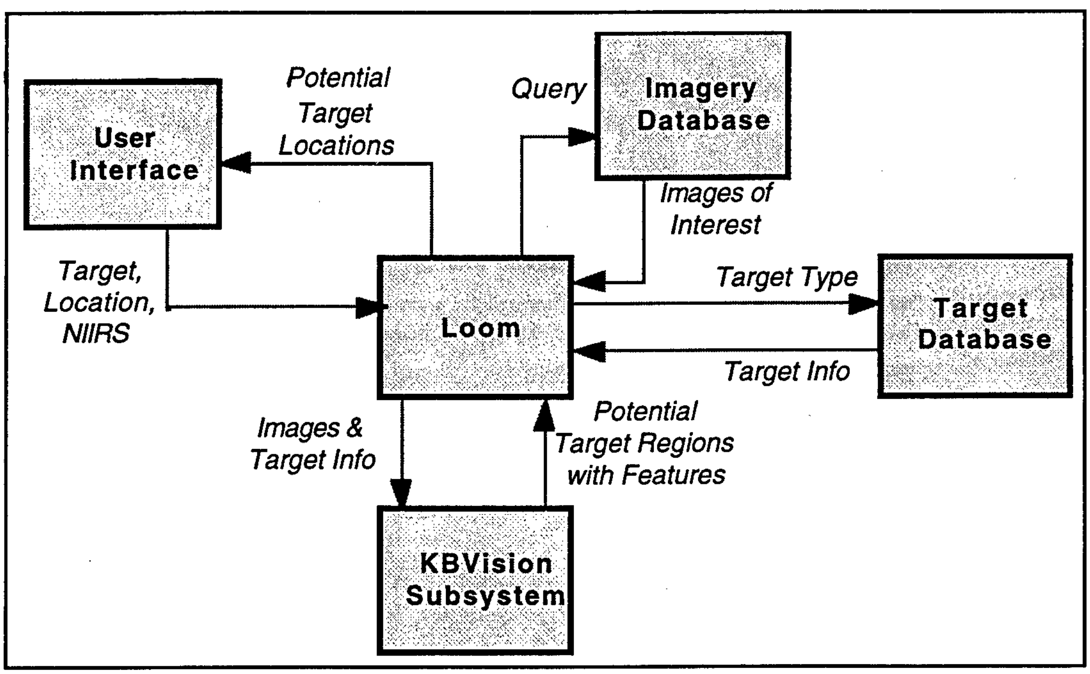

Figure 1: IU4DBQ System

2. LOOM / LISP DATABASE QUERYING SYSTEM

The integration of Loom with KBVision was performed in order to make the results of image processing and segmentation available for use by Loom's recognition system. Additionally, KBVision image processing tasks can be invoked with parameters determined by the Loom inference.

The IU4DBQ demonstration system implements the following scenario to demonstrate the integration of Loom and KBVision:

1. An imagery database containing imagery of various sensor types and imagery meta data is loaded into the Loom/Lisp system. Free-form text comments mentioning order-of-battle elements in each image are scanned from the imagery meta data. Instances of concepts in a simple Loom domain model, such as `IMAGE` and `OB-ELEMENT`, are created and supporting relations are asserted.

2. A Loom query for optical images containing order-of-battle (OB) elements identified with a high degree of certainty are made available for display to the operator.

3. A Loom query for optical images of better than average quality containing low-certainty OB elements are made available for KBVision processing.

4. A KBVision task is parametrized using Loom domain model knowledge about the type(s) of OB elements being sought in the image.

5. The KBVision task is invoked, generating a tokenset containing tokens believed to be OB elements of the desired type.

6. The tokenset is imported into Loom, adding the tokens as possible OB elements.

7. The processed image, with tokens outlined, is presented to the operator for cueing purposes.

Loom Contexts

Loom provides contexts as a means of partitioning knowledge and beliefs. Contexts may be inherited, forming a directed acyclic graph. As models or assertions are added to a context, each of its descendant contexts also see the changes. The set of contexts used in IU4DBQ may be seen in Figure 2.

The root of contexts generated by the IU4DBQ system is `IU4DBQ-THEORY`, which is a child of the default context in Loom, `BUILTIN-THEORY`. The `IU4DBQ-THEORY` context contains all of the concepts, relations, and methods necessary to support representation and manipulation of KBVision tokensets. It also contains concepts and relations in a simple model of the imagery and order-of-battle domain. This context is created at system initialization and forms the basic theory for IU4DBQ processing. It is not changed during system operation.

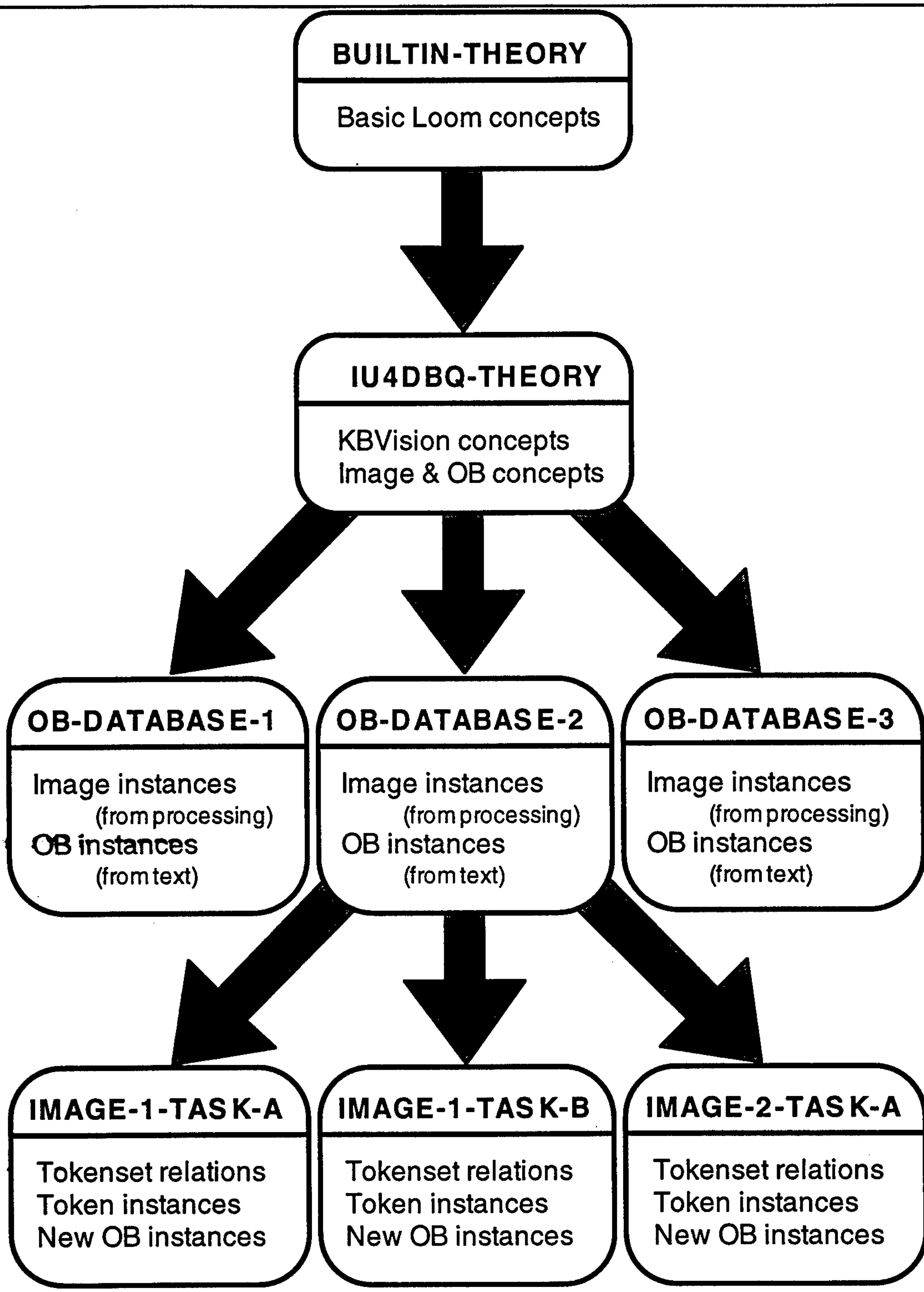

Figure 2: Example IU4DBQ Context Graph

A new context (e.g., `OB-DATABASE-1`) is created for each imagery/OB database that is loaded into the system. This context contains assertions about the imagery meta data and OB elements as parsed from the database. Loom queries may be formulated against these contexts to identify images having OB elements of interest.

When a KBVision task is invoked on an image, the resulting tokenset is imported, generating a new context containing new relations defined by the tokenset's lexicon and assertions representing feature values on the tokenset's tokens. Since different KBVision tasks represent distinct strategies for segmenting an image, distinct contexts are created for each task.

Running KBV Tasks

When a candidate image for KBVision processing is found by a Loom query, the type(s) of OB elements believed to be in the image (albeit with low confidence) are used to select task parameters. As described in section 3, the parameters for the demonstration were the expected length and width of the OB element being sought. The conversion from metric measurements found in the domain models was accomplished using meta data for the image to be processed.

Importing Tokensets

In order for Loom to be able to interpret the results of processing by the KBVision system, a tokenset must be converted into a set of Loom relations and assertions about the tokens in the tokenset. A KBVision tokenset has two sections, a lexicon, which describes the features found in the tokenset; and the token descriptions, which provide the feature values for each token.

KBVision has a set of basic types (e.g., KBV_COLOR, KBV_CONSTELLATION, KBV_POINT, and KBV_FCHAIN) and standard features (e.g., EXTENTS) for which equivalent Loom concepts and relations (respectively) were implemented. These concepts and relations were defined in the base IU4DBQ-THEORY context. Additional relations were defined from a tokenset's lexicon into a new context. Each tokenset has its own context, since the lexicon of a tokenset is determined by the KBV task(s) used in its creation.

Once the relations are defined for a tokenset's lexicon, the tokens' feature/value pairs are used to make assertions about newly-generate instances of the TOKEN concept in Loom. In the demonstration system, it was assumed that the KBVision task would produce a sparse tokenset, where each token represented a patch of the image where an OB element of the desired type might be found. Each resulting TOKEN instance in the tokenset's context was asserted as being an instance of the OB element type being sought.

3. KBVISION SUBSYSTEM

The KBVision subsystem processes imagery and target information that is passed from the Loom subsystem and returns token sets that mark the locations of potential targets. It relies on the size and profile of the targets. Since the size of the targets are constant, the absolute target dimensions are used, and the pixel footprint size to determine the targets' pixel dimensions is used. These dimensions are defined in terms of *length* and *width* with the length being the longer of the dimension. The targets are typically long and narrow; consequently, the profile of the targets will take on a ridge or valley characteristic, depending on whether the target is brighter or darker than the background. All this information is used to segment and detect targets in the imagery.

The KBV subsystem is broken into two phases. The first is the "image processing" phase that segments ridge structures from the imagery. The second is the "feature extraction / token set analysis" phase that analyzes the geometric properties of the segmented regions (or tokens). Figure 3 shows the two phases and also describes how they work. Figure 4 depicts the actual

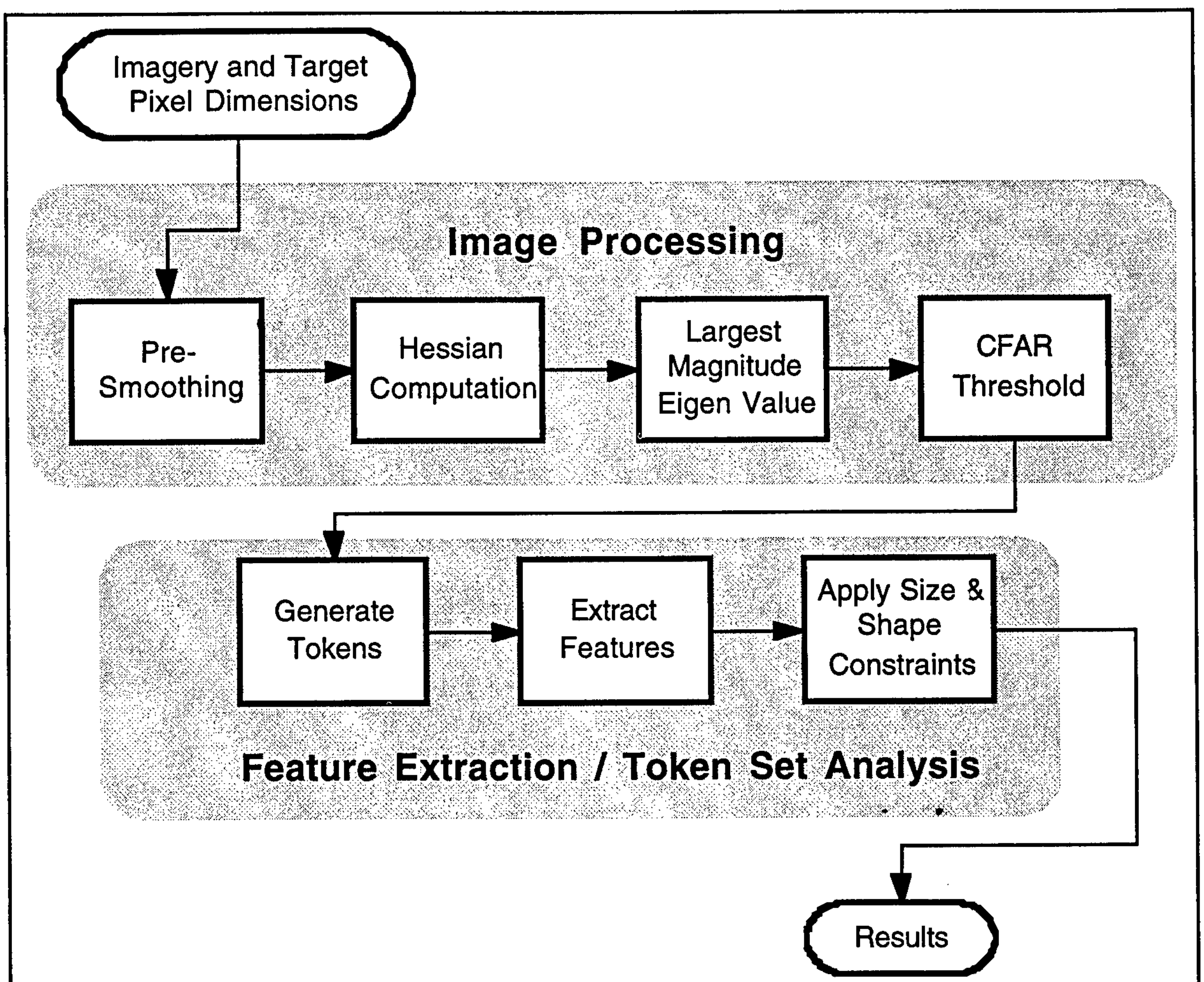

Figure 3: KBVision Subsystem

KBVision interface. We will first discuss the image processing phase and then the feature extraction / token set analysis phase.

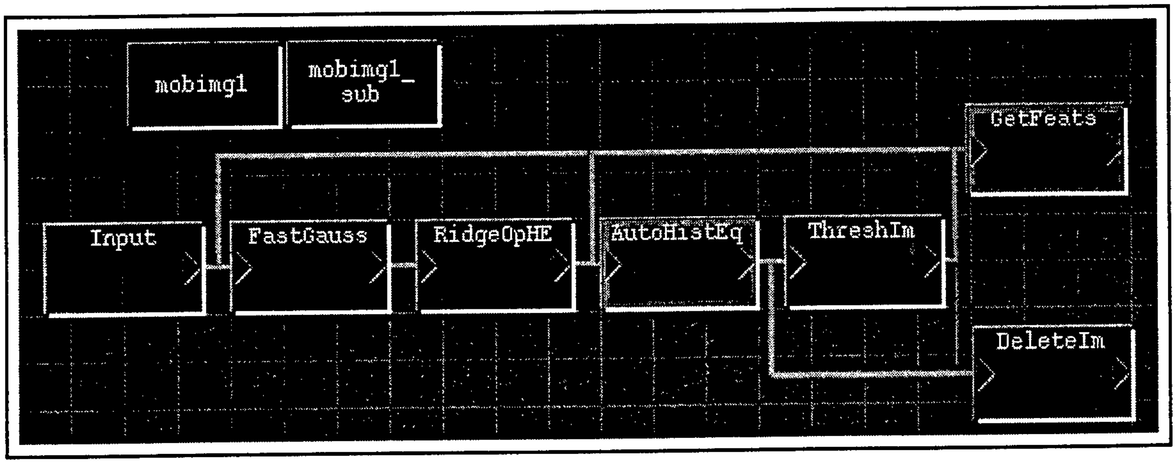

Figure 4: KBVision Interface

The KBVision's facet modeling capability was used as the basis for the image processing. These techniques produce a Hessian matrix which is associated with four directional second derivatives. Second derivative processes are inherently sensitive to noise; thus, a pre-smoothing process was used to alleviate these sensitivities (*FastGauss* task). This task applies a gaussian weighted convolution to the data. The kernel size is determined by the target width which attenuates artifacts of higher spatial frequency.

The Hessian matrix is used to find ridge and valley shaped profiles in the data. Ridges are characterized by large magnitude negative eigen values and valleys are characterized by large magnitude positive eigen values. Since there is no *a priori* knowledge on whether a target is brighter or darker than the background, the eigen value that has the largest magnitude is used. The KBV task that performs this operation is called *RidgeOpHE* and details of its operation are detailed in Appendix A. The image is segmented using a Constant False Alarm Rate (CFAR) threshold. Specifically, the darkest 5% and brightest 5% of the pixels are segmented to form tokens. This CFAR technique uses a custom compound task called *AutoHistEq* and the KBV task *ThreshIm.*

KBVision's token set capability is used to segment regions and extract their features. The custom compound task that does this is *GetFeats*. First, the token sets are created from contiguous thresholded regions. To facilitate processing, extremely large tokens are eliminated. Next features are extracted from the tokens. Although the algorithm actually extracts dozens of features, only three are used. All of these features are based on the minimum bounding rectangle (MBR) of a token. The MBR, as the name implies is the smallest rectangle that encloses the token in question. The advantage of such a feature is that it is independent of the target's orientation. Bounds are then place on the length and width of the MBR, which are directly related to the pixel dimensions of the target which the KBV subsystem uses as input. Tolerances are included for the MBR dimensions to allow for noise effects and thresholding uncertainties. The current tolerance is ±10 pixels. We also apply a lower limit to the MBR fill ratio feature, which is the fraction of the MBR that the token fills. This prevents irregularly shaped tokens that have the expected MBR dimensions from being detected. Currently, this threshold is set at 0.33.

4. RESULTS

The system was tested using missile order of battle (MOB) imagery and intelligence data. The scenario used Loom to query the database for unconfirmed MOB elements. Next, Loom tasked the KBVision Subsystem to process these images. Only one image met the query specifications. Unfortunately the resolution of this image was much too low for KBV to process with any reasonable degree of accuracy. There were not enough pixels on a target to detect and hundreds of false detections were scattered throughout the scene. The imagery where the KBV is effective is the images which had confirmed MOB elements. Specifically, 13 confirmed targets are scattered throughout the scene. These results on this image are tabulated below:

Table 1, Results of Experiment

Performance Comparison	
	Image 1
NIIRS Rating	
Confirmed Targets	13
Detections	10
False Alarms	15

Ten out of thirteen confirmed targets were detected and there were fifteen false detections. Such results can be helpful, because they allow a PI to quickly scrutinize potential areas of interest. In this particular case, a PI would evaluate 25 areas of interest. Ten of these are actual targets and fifteen are false detections. There would then be three additional targets that the PI would need to locate by traditional means. Although, the use of this system would not increase the overall accuracy of a PI's examination, it can facilitate tasking such that the imagery could be evaluated more efficiently.

5. CONCLUSIONS

This effort integrated the knowledge representation and reasoning capabilities of Loom with the image processing and token feature extraction capabilities of KBVision. The lisp subsystem is responsible for querying the extensive imagery database in the IE2000 facility and selecting images with the appropriate location, target content, and NIIRS rating for automatic processing. The specific query was set up to find images with unconfirmed targets. For the single experiment accomplished for testing, one image was returned based on this query.

The KBVision interface was reasonably successful at detecting the targets of interest in reconnaissance imagery with already confirmed targets. Specifically, results were achieved that would be useful in helping a PI locate targets; it cues the PI to where targets are potentially located. The detection rate is high enough and the false alarm rate low enough that this is feasible approach. However, it can not and should not be used as the final decision on target locations. False detections and missed targets can and will invariably occur. It is up to the PI to evaluate the information provided by KBV and then make the final decisions.

KBVision was unable to effectively process the image which came out of the Loom database query due to insufficient spatial resolution. Therefore, the image processing can handle images that a PI can handle. However, like a PI, the system will not be effective on lower quality images. Therefore, the system is not detecting anything that a PI cannot already detect. The primary benefit of the KBV processing is to assist the PI with higher quality imagery.

KBV was sluggish when handling large images. Even though Version 3.1 has new features for supporting large images, RAM or CPU was never fully utilized. A 6 Mbyte image (which becomes 24 Mbytes when we convert from byte format to floating) caused swapping, even though the workstation had 128 Mbytes of RAM. The problem is believed to be with KBV's memory management. It took hours to convert a Sun Raster image to the KBV format.

The image processing portion of this effort and the querying capability developed with Loom were both successful. However, AAI no longer supports the KLI interface, which simplified the integration of KBV and Loom. In its place, AAI is developing a new Image Understanding Environment (IUE). The IUE is an object-oriented software environment, based on C++. It is likely that future IU efforts will move in this direction. Further information about the IUE is available at the following URL on the internet:

http://www.aai.com/AAI/IUE/AboutIUE.html.

An alternative approach to the IUE is to develop a custom interface between KBV token sets and Loom (or lisp), using KBVision C functions. This would become an additional KBVision task and would be much simpler than using the KLI.

APPENDIX A: RIDGEOPHE TASK DESCRIPTION*

This task finds ridges and/or valleys in an image surface by computing the Hessian at each pixel and then finding its maximum and minimum eigen values and the eigen vector associated with the maximum eigen value. Ridges are characterized by large negative eigen values and valleys by large positive eigen values.

The parameter *MaxMinOutputSelection* selects one of two output methods. If *MaxMinOutputSelection* = 1, then the maximum and minimum eigen values are output in the images *MaxEigenvalue* and *MinEigenvalue*. If *MaxMinOutputSelection* = 2, then the eigen value with the maximum magnitude (absolute value) is output in the image *MaxEigenvalue*.

The eigen vector associated with the maximum eigen value is output in the images *MaxEigenvectorX* and *MaxEigenvectorY*. The components of the eigen vector are normalized so the component with maximum magnitude is scaled to ±100. (The components are not normalized by Euclidean length to avoid use of sqrt()).

If the optional LOGICAL image, *MaskImage*, is provided then processing only takes place at pixels where *MaskImage* = 1.

Interpretation of the Hessian

For a given image function $I(x,y)$, the Hessian is the following 2×2 matrix:

$$\begin{vmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{vmatrix}$$

where the elements are the second partial derivatives of I. The eigen values of the Hessian at a given point are equal to the maximum and minimum values attained by the directional second derivatives of I at that point. The associated eigen vectors point in the corresponding directions. These two directions are orthogonal to each other.

The elements of the Hessian are computed as finite difference operators using convolutions with the following kernels:

$$I_{xx} \qquad\qquad I_{xy},\ I_{yx} \qquad\qquad I_{yy}$$

$$\frac{1}{3}\begin{vmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{vmatrix} \qquad \frac{1}{4}\begin{vmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{vmatrix} \qquad \frac{1}{3}\begin{vmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{vmatrix}$$

* This appendix is part of the KBVision™ documentation.

Pre-Smoothing

The results of this ridge finder may be improved by pre-smoothing, such as Gaussian-weighted convolution, for two reasons. First, the second difference convolutions that compute the elements of the Hessian are not robust in the presence of noise. (In fact, as individual operators, they are high-pass filters.) Secondly, to the extent that the ridges to be detected are flat-topped, this operator will give strong response at the sides of the ridge and only a weak, if not zero, response in the center of the ridge. In this case, thresholding may produce a double ridge. Pre-smoothing rounds off the ridge, thereby strengthening the central response.

☆U.S. GOVERNMENT PRINTING OFFICE: 1997-509-127-47190

DISTRIBUTION LIST

addresses	number of copies
MR. PETER J. COSTIANES ROME LABORATORY/IRRE 32 HANGAR ROAD ROME NY 13441-4114	5
MR. ED BOHLING PAR GOVERNMENT SYSTEMS 8383 SENECA TURNPIKE NEW HARTFORD NY 13413	5
ROME LABORATORY/SUL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: RAYMOND TADROS GIDEP P.O. BOX 8000 CORONA CA 91718-8000	1
AFIT ACADEMIC LIBRARY/LDEE 2950 P STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
WRIGHT LABORATORY/MTM, BLDG 653 2977 P STREET, STE 6 WRIGHT-PATTERSON AFB OH 45433-7739	1

ATTN: GILBERT G. KUPERMAN AL/CFHI, BLDG. 248 2255 H STREET WRIGHT-PATTERSON AFB OH 45433-7022	1
US ARMY STRATEGIC DEFENSE COMMAND CSSD-IM-PA P.O. BOX 1500 HUNTSVILLE AL 35807-3801	1
NAVAL AIR WARFARE CENTER 6000 E. 21ST STREET INDIANAPOLIS IN 46219-2189	1
COMMANDING OFFICER NCCOSC RDT&E DIVISION ATTN: TECHNICAL LIBRARY,CODE 0274 53560 HULL STREET SAN DIEGO CA 92152-5001	1
COMMANDER, TECHNICAL LIBRARY 4747000/C0223 NAVAIRWARCENWPNDIV 1 ADMINISTRATION CIRCLE CHINA LAKE CA 93555-6001	1
SPACE & NAVAL WARFARE SYSTEMS COMMAND (PMW 178-1) 2451 CRYSTAL DRIVE ARLINGTON VA 22245-5200	2
SPACE & NAVAL WARFARE SYSTEMS COMMAND, EXECUTIVE DIRECTOR (PD13A) ATTN: MR. CARL ANDRIANI 2451 CRYSTAL DRIVE ARLINGTON VA 22245-5200	1
CDR, US ARMY MISSILE COMMAND RSIC, BLDG. 4484 AMSMI-RD-CS-R, DOCS REDSTONE ARSENAL AL 35898-5241	2
ADVISORY GROUP ON ELECTRON DEVICES SUITE 500 1745 JEFFERSON DAVIS HIGHWAY ARLINGTON VA 22202	1

REPORT COLLECTION, CIC-14 1
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

AEDC LIBRARY 1
TECHNICAL REPORTS FILE
100 KINDEL DRIVE, SUITE C211
ARNOLD AFB TN 37389-3211

COMMANDER 1
USAISC
ASHC-IMD-L, BLDG 61801
FT HUACHUCA AZ 85613-5000

AFIWC/MSO 1
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

NSA/CSS 1
K1
FT MEADE MD 20755-6000

PHILLIPS LABORATORY 1
PL/TL (LIBRARY)
5 WRIGHT STREET
HANSCOM AFB MA 01731-3004

THE MITRE CORPORATION 1
ATTN: E. LADURE
D460
202 BURLINGTON RD
BEDFORD MA 01732

OUSD(P)/DTSA/DUTD 2
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

MISSION OF ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.