ARKTOS: A Knowledge Engineering Software Tool for Images

Leen-Kiat Soh
Computer Science and Engineering
University of Nebraska
115 Ferguson Hall, Lincoln, NE 68588-0115 USA
Tel: (402) 472-6738 Fax: (402) 472-7767
E-mail: lksoh@cse.unl.edu

Costas Tsatsoulis
Electrical Engineering and Computer Science
University of Kansas, ITTC
2335 Irving Hill Rd.
Lawrence, KS 66045-7612 USA
Tel: (785) 864-7749 Fax: (785) 864-0387
E-mail: tsatsoul@ittc.ku.edu

Abstract – The goal of our ARKTOS project is to build an intelligent knowledge-based system to classify satellite sea ice images. It involves acquiring knowledge from sea ice experts, quantifying such knowledge as computational entities, and ultimately building an intelligent classifier. In this paper we describe a two-stage knowledge engineering approach that facilitates explicit knowledge transfer, converting implicit visual cues and cognition of the experts to explicit attributes and rules implemented by the engineers. First, there is a prototyping stage that involves interviewing sea ice experts, transcribing the sessions, identifying descriptors and rules, designing and implementing the knowledge, and delivering the prototype. The objective of this stage is to obtain a modestly accurate classification system quickly. Second, there is a refinement stage that involves evaluating the prototype, refining the knowledge base, modifying the design, and re-evaluating the improved system. Since the refinement is *evaluation-driven*, the experts and the engineers are motivated explicitly to improve the knowledge base and are able to communicate with each other using a common, consistent platform. Moreover, since the classification result is immediately available, both sides are able to efficiently assess the correctness of the system. To facilitate the knowledge engineering of the second stage, we have designed and built three Java-based graphical user interfaces: arktosGUI, arktosViewer, and arktosEditor. arktosGUI concentrates on feature-based refinement of specific attributes and rules. arktosViewer deals with regional evaluation. arktosEditor has a rule indexing and search mechanism and knowledge base editing capabilites.

1 INTRODUCTION

The goal of our ARKTOS (Advanced Reasoning using Knowledge for Typing Of Sea ice) project is to perform automated, intelligent satellite sea ice image classification. Our approach is to acquire the classification knowledge from sea ice geophysicists and photo-interpreters and implement the knowledge in a rule-based system that also utilizes image processing methodologies. To facilitate a knowledge transfer from the sea ice experts to the knowledge and software engineers, we employ a two-stage knowledge engineering approach that includes rapid prototyping and evaluation-driven refinement. The rapid prototyping allows the experts and the engineers to design and implement a modestly accurate classification system. Given the functional system, the experts are then able to evaluate its classification results. To facilitate the evaluation process, we create Java-based software tools such that implicit visual cues and cognition of the experts can be explicitly expressed and fine-tuned in attributes and rules. The prototyping stage involves

(1) face-to-face interactive sessions between sea ice experts and knowledge engineers, and (2) discussion sessions between knowledge and software engineers, to ultimately design and implement a system prototype. The second stage is refinement, which requires sea ice experts to use and evaluate the prototype and provide feedback to the software engineers. To facilitate this knowledge refinement cycle, we have built a suite of software tools: arktosGUI, arktosViewer, and arktosEditor. These tools deal with feature-based refinement of specific attributes and rules, regional evaluation to determine the accuracy of the classification, and knowledge editing and maintenance, respectively.

Our approach to knowledge engineering is to first use the prototyping stage to jump-start the process, and refine the system later. Traditionally, image-related knowledge engineering has involved manual interviews and training between the domain experts and knowledge engineers, by inspecting images either digitally or on paper. Since it is difficult for the experts to articulate their knowledge explicitly or in quantifiable terms and similarly difficult for the engineers to ask the appropriate questions of the experts, knowledge acquisition has been viewed as the bottleneck of knowledge engineering (Michie and Johnston 1985). This problem is many times magnified in image analysis from both ends of the spectrum. Experts deal with visual cognition that is inherently implicit, abstract, and difficult to pinpoint. Software engineers have to interpret complex image data that is difficult to understand. However, by focusing on rapid prototyping, we aim at obtaining a modestly complete knowledge base such that a prototype can be designed and implemented quickly. During the refinement stage, the experts can then explicitly evaluate the knowledge base—by running the prototype on images and inspecting the classification results and, consequently, effectively evaluating and quantifying the suitability of an attribute or the correctness of a rule. This evaluation-driven approach has two advantages that are usually absent in traditional approaches: (1) it is efficient in that each refinement results in a direct improvement to the classification, and (2) it strengthens the links between the experts and the engineers since both sides can now communicate using explicitly computable entities and visual examples.

Before concluding this section, we observe two important points of our knowledge engineering approach. First, note that the effectiveness of our approach depends on the generality and extensibility of our prototype and the easy-to-use GUIs of the refinement tools. If the prototype is not general and not easily extensible, then it will make the refinement process non-trivial.

If the refinement tools do not provide instant, visual feedback and user-friendly features, then the experts will be discouraged from using them. Second, our knowledge engineering approach and software tools are useful to machine vision and intelligence. The approach is a practical, effective way of acquiring and refining visual cognitive knowledge, and the tools help in molding the knowledge into an intelligent system. Our work demonstrates an application of our approach and an example on transforming human intelligence to machine intelligence.

In this paper, we focus our discussions on the knowledge engineering process and tools of ARKTOS. In the next section, we provide some background on satellite sea ice image classification and its importance and why intelligent systems are useful in such applications. In the third section, we present an overview of the ARKTOS system. In Section 4, we describe our prototyping stage. Then, in Section 5, we detail the three modules of the refinement stage. In Section 6, we discuss the evaluation of our software tools by sea ice experts. In the seventh section, we discuss some implications of our approach and the software package in machine intelligence and computer vision. Finally, we conclude the paper.

2 BACKGROUND

2.1 Sea Ice Remote Sensing

Remote sensing of the polar regions has important applications in meteorology and global climate. For example, the thickness of sea ice influences the heat flux between the atmosphere and water surface; thus, the classification and temporal tracking of sea ice can be used as an indicator in global climate monitoring. In addition, localization and classification of ice floes are important to commercial, scientific, and military navigation of the polar regions for offshore and deep-sea fishing, oil drilling, marine biology, and geology explorations.

With the increased concern regarding global climate and the subsequent increase in the number of earth-orbiting satellites, there is a dramatic increase in the volume of imagery data available to scientists. The imaging resolution of these satellites is higher, and the satellites can collect data at a faster pace with on-board storage. Thus, some sort of automation in sea ice image classification is much desired to assist sea ice experts in daily processing of the increasing volumes of images (Tsatsoulis and Kwok 1998).

The development of a rule-based system also provides a convenient platform in multi-source data fusion. As discussed briefly in Section 3, ARKTOS incorporates several other

sources of data and information before classifying an image. By automating this data fusion process, ARKTOS helps sea ice experts in retrieving and disseminating data needed to classify images, allowing them to concentrate on more important decision making tasks. Moreover, with the knowledge explicitly written in a rule base, the experts can review their reasoning process to become more consistent, can exchange their different viewpoints in certain classifications, can transfer their knowledge readily in training new sea ice analysts, etc. It also allows the knowledge to be better organized, portable, and accountable.

2.2 Related Work

Of all the knowledge engineering stages, knowledge acquisition (KA) has traditionally been difficult and has prompted much research in computer-based tools (Nwana *et al.* 1991; Boose 1993) that help make the process easier. Most KA tools require expert users to supply the description of a case or knowledge, outlining either completely or partially the attributes or concepts involved in the knowledge tasks or applications—for example, MORE (Kahn *et al.* 1985a; 1985b), AQUINAS (Boose *et al.* 1995, Boose and Bradshaw 1987), MIKE (Eisenstadt and Brayshaw 1990), SALT (Marcus 1987), KITTEN (Shaw and Gaines 1987), KNACK (Klinker *et al.* 1987), KSS0 (Gaines and Shaw 1993), ALTO (Major and Reichgelt 1990), CMBKATs (Charlet *et al.* 1992), KATEMES (Dieng *et al.* 1992), CERISE (Vicat *et al.* 1993), KADS (Schreiber *et al.* 1993; van Heijst *et al.* 1997), and many others . However, none of these KA tools has dealt with image-based knowledge ontology and devised a tool that facilitates either the acquisition or the refinement of the knowledge.

Some KA tools exploit visual programming methodologies for knowledge elicitation and subsequent knowledge organization. These tools allow the experts to create, arrange, and link concepts or attributes as visual items and traverse the semantic space in a natural form (such as diagrams or graphs) on the screen, and then automatically encode the relationships into a representation structure as the knowledge base (Eisenstadt *et al.* 1990)—for example, KEATS (Motta *et al.* 1989; Rajan *et al.* 1988), CLARITY (Addis *et al.* 1993), CGKEE (Munday *et al.* 1995), and law encoding diagrams (Cheng 1996).

The work of Gaines and Shaw (Gaines and Shaw 1980; 1993; Shaw and Gaines 1993) incorporated the repertory grid and personal construct psychology of Kelly (1955) into a novel methodology for eliciting conceptual structures and also for quantifiable feedback and evalua-

tion. There are KA tools based on this principle—for example, ETS (Boose 1986), KSS0 (Gaines and Shaw 1987), and AQUINAS (Boose and Bradshaw 1987). Our work is similar in that we also view that some human knowledge is not readily available and explicit, and we aim at acquiring such knowledge in an intelligent manner. The work of Gaines and Shaw used the repertory grid and the theory of personal construct psychology, assuming that attributes/concepts were readily available. In our work, we use an image-based ontology that is derived from the image understanding domain and designed based on an initial knowledge transfer of sea ice classification from the experts to the engineers, and a subsequent computer-aided knowledge refinement process.

Note that our knowledge engineering approach is necessary since visual cognitive cues and reasoning process are extremely difficult to extract and that makes the other approaches impractical or ineffective. Note also that our approach is basically a KA technique—one that requires a rapid prototyping and a GUI-supported refinement software package, and one that specializes in image-based ontology and database.

3 OVERVIEW OF ARKTOS

ARKTOS is a multi-source, rule-based sea ice classification system. It has four primary modules, as depicted in Figure 1. First, the image segmentation module of ARKTOS segments the original raw image into regions and tags each region as a feature. Then, for each feature ARKTOS measures the value of a set of attributes. At the end of the measurement module, ARKTOS has a set of numbers describing each feature. To better utilize this information, ARKTOS converts the numbers into symbols using the fact generation module. The fact module uses thresholds initialized by the engineers and later refined by the experts to quantize the numbers into symbolic facts. Suppose that the attribute "size" has a range between 50 pixels and the size of the entire image and there are attribute-value pairs: "size=small", "size=average", and "size=large". Thus, we need two thresholds, *T1*, and *T2*. The conversion algorithm is then:

If the size of the feature is smaller than T1 **Then**
    it is a *small-sized* feature
Else If the size of the feature is smaller than T2 **Then**
    It is an *average-sized* feature
Else
    It is a *large-sized* feature

In addition to conversion, the fact generation module also performs data fusion. It reads in various sources of data such as the image's corresponding land mask, ancillary data that accompanies the original image, Special Sensor Microwave/Imager (SSM/I) ice concentration maps, and historical climatology maps, retrieves corresponding data from these sources, and converts these data to symbolic facts as well. Finally, given the features and their symbolic facts, the fourth and final module performs a rule-based classification using a Dempster-Shafer belief system. For each feature, the reasoning process fires matched rules sequentially and collects the weight of each fired rule as a piece of evidence for or against a particular ice classification (e.g., open water, first year ice, fast ice, or old ice). The evidence is collected as mass values and totaled in the form of beliefs and plausibilities. The module finally labels each feature to either the ice class that garners the highest product of belief and plausibility or the *unknown* class when the evidence is weak and no ice class reaches an acceptable classification threshold. In addition to the classified image, ARKTOS also generates a log of the classification that includes the actual measurements, facts, and rules fired.

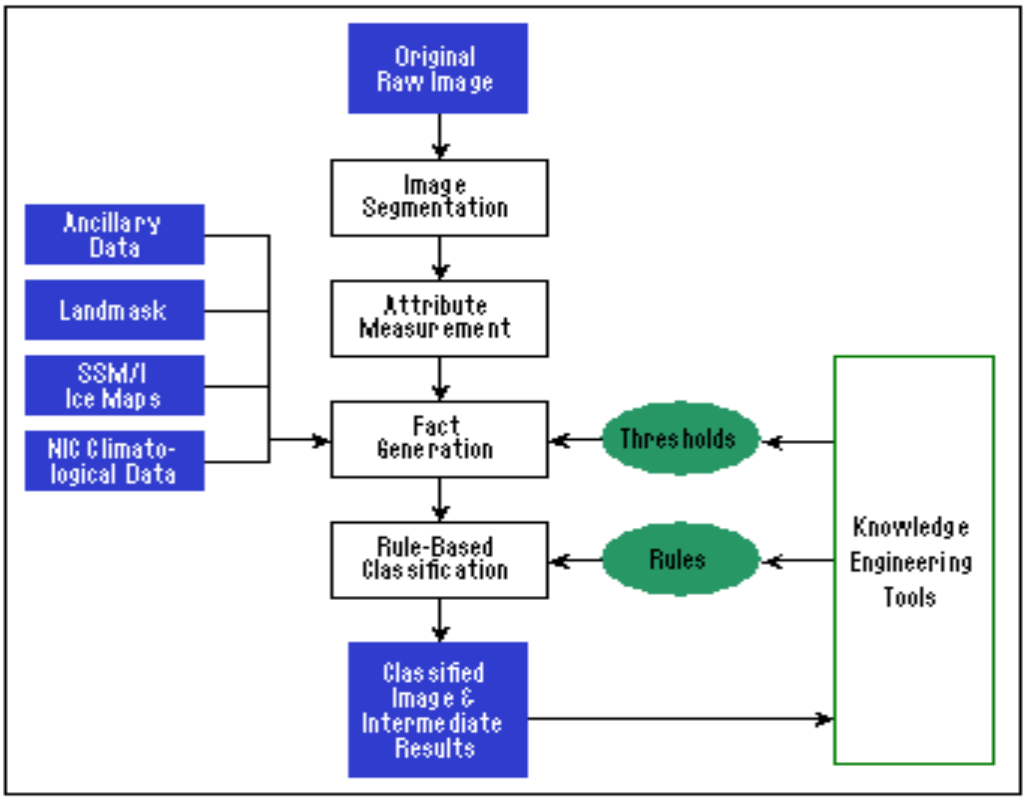


Figure 1 Overview of ARKTOS, the intelligent sea ice image classifier.

ARKTOS has been designed with primarily in two roles in mind once it is fully deployed and incorporated into the flow of the operations: (1) assist geophysicists or photo-interpreters in identifying sea ice classes, and (2) automate pre-processing and data fusion tasks to help streamline the daily workflow.

4 PROTOTYPING STAGE

As discussed in the introduction, our approach to the knowledge engineering process of ARKTOS is divided into two stages. The first stage is for prototyping. It consists of primarily six steps, as depicted in Figure 2: (1) Knowledge Acquisition, (2) Knowledge Pre-Processing, (3) Knowledge Verification, (4) Knowledge Extraction, (5) Design and Implementation, and (6) System Development and Prototyping.

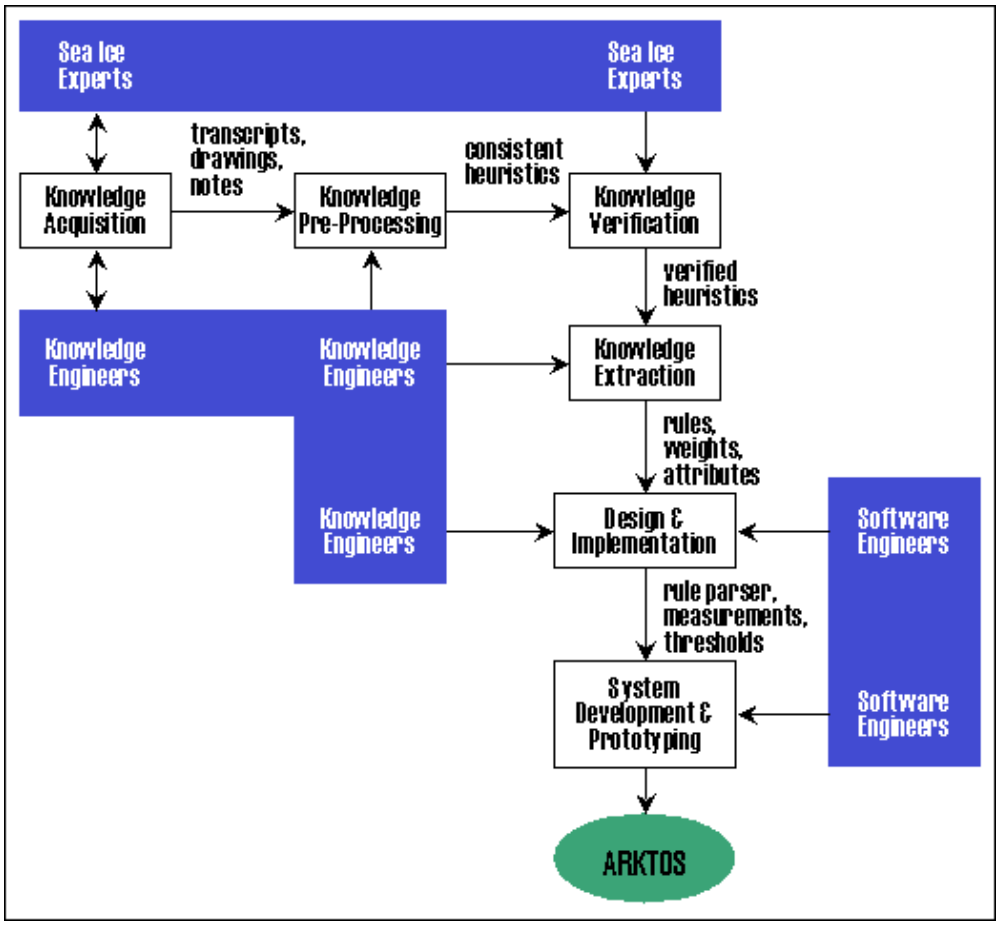


Figure 2 Stage 1 of our knowledge engineering process: prototyping. The goal of this stage was to acquire knowledge from sea ice experts and create a common platform on which experts and engineers can exchange and refine ideas. That platform is a set of attributes, weights, thresholds, and rules. The evaluation is through the prototype ARKTOS—judging its classification results on a set of satellite sea ice images.

4.1 Knowledge Acquisition

Knowledge acquisition is the task of acquiring knowledge from experts, and it is non-trivial and difficult. Open-ended issues such as personalities, manners of expression, reasoning processes of the experts, forms of questions and follow-ups, recording and notation methods, must be addressed in order to transcribe accurately what the experts mean. We designed our knowledge acquisition sessions based on the works of (Scott, Clayton, and Gibson 1991) and (Hart 1992): protocol, blind test, and cross reference. We conducted interviews with three expert sea ice interpreters, two from the Canadian Ice Service (CIS) and one from the National Ice Center (NIC). Each interview lasted between three and five days.

Protocol is based on the talk-through method in which the expert thinks aloud while classifying images. The expert being interviewed was seated in front of a computer monitor and asked to classify verbally an electronically displayed SAR sea ice image. This method provided an unstructured interview since no questions were pre-determined. It also closely simulated the normal work environment of sea ice experts in classifying sea ice images.

The *blind test* method is based on the twenty questions concept (Hart 1992), where the interviewed expert asks questions about an unseen case, and it forces the expert to verbally describe the classification process by classifying an image without actually seeing it. In this sense, we were able to obtain the chain of reasoning—observations to conclusions—explicitly from the experts. This method was conducted with an expert, a knowledge engineer to oversee the process, and a novice who knew nothing about sea ice images. Only the novice was allowed to view an electronically displayed SAR sea ice image. The expert then visualized the image—and eventually classified it—by asking the novice questions. The goal was to bypass the unconscious visual processing of humans and to force the experts to externalize it. In this way we managed to acquire explicit definitions of vague concepts such as "wiggliness" and "mottled," since the experts had to describe the low-level visual characteristics they would use to identify them. The more abstract concepts were eventually defined by a combination of image features (intensity, length of perimeter, size, etc.) that image processing algorithms could extract.

Cross reference-based sessions use predetermined questions to elicit specific information. The expert was asked to comment on heuristics that we had obtained from other experts. This helped resolve conflicts, verify assertions, and refine knowledge in terms of certainty and accuracy.

Each acquisition session was recorded on audiotapes. Images were displayed on a computer monitor allowing the ability to zoom in on and enhance specific areas. In addition, high-quality laser prints of all images were provided such that the expert could annotate the images accordingly. The audio record was later transcribed with references to corresponding images and ice features.

4.2 Knowledge Pre-Processing

As mentioned, after the acquisition sessions, we obtained audio, imagery, and textual transcripts. We first performed a noise filtering to all transcripts. We weeded out transitional conversations and reconciled contextually similar assertions. After noise filtering, we organized the information. The first step was to compile a list of clear statements from all the knowledge data that we had gathered. A statement could be an if-then type assertion, a tautology, a rationale, or an observation. From these statements, we then obtained classification heuristics. Each heuristic was tagged with the expert who made the assertion, an example of the image or feature in question, and a summary of the notes that supported the assertion. In this manner, we grouped the heuristics into different ice classes, identified contradictions, and assigned weights to the heuristics based on linguistic variables used by the experts, such as "probably," "I would say," or "must be."

For knowledge verification, we gave our compiled heuristics to all sea ice experts to review. Heuristics that all experts agreed on would not be modified. Heuristics that only garnered partial support would have their weights lowered. After this verification process, we had a set of consistent heuristics. Note that since we had a knowledge pre-processing step, we did not overburden our experts with the raw transcripts.

4.3 Knowledge Extraction

The objective of this extraction step was to obtain attributes and rules. We assimilated the heuristics to obtain dozens of attributes such as adjacent_to, angular, brighter, smooth, roundness, encloses, elongated, jaggedness, etc. By explicitly identifying these attributes, we qualified the knowledge base, and were able to form consistent rules. Here, we also defined a common terminology to describe attributes and their values. For example, experts use interchangeably *circular*

and *rounded* to describe round features. We realized that both terms referred to the same concept, and thus used only *roundness.*

The following is an example of a Q&A session:

> *A: Sort of looks like a piece of old ice on top there.*
> *Q: OK. Let's call that A. Why do you think so?*
> *A: Again, because of a smooth rounded edge with still a type of mottled texture inside it. Sort of a darker gray picture. I would say that is all old ice.*

The above example was translated into:

If feature has mottled texture and is not jagged, then old ice; 0.6.

The 0.6 value denotes the confidence the expert had with the rule and it corresponds to the choice of words (such as the use of *sort of* and absence of definitive adverbs or adjectives) as transcribed. Rules were arranged into two modules: winter and summer. So, for a winter image, only winter rules will be used for matching and firing, and vice versa. This modularity is computationally efficient because (1) the search space for matched rules is restricted to a certain rule base for each image and (2) winter and summer rules are significantly different, since sea ice looks and acts differently during these two seasons.

4.4 Design and Implementation

So far, the previous engineering process involved only the sea ice experts and the knowledge engineers. For design and implementation of the rules and attributes, the knowledge engineers now turned to the software engineers, who were knowledgeable in image processing as well. The aim was to qualify the attributes as (programmable) mathematical equations as closely as possible, and not necessarily accurately. The software engineers also took computational constraints into account such as memory requirements and execution time of the algorithms, and attributes that required too much memory and time to compute were approximated. These behaviors were a key to our rapid prototyping objective. First, sea ice experts were a costly resource in terms of flying them into town to conduct interviews and of scheduling appropriate meeting times within a short period of time. Thus, it would be counter-productive to consult the experts for each design to ensure accuracy at this stage. Second, the knowledge engineers were a conduit between the experts and the programmers, making the conceptualization process more effective. This

was because the engineers were more familiar with each other's language and concepts. So, by excluding the experts from this step, we traded accuracy for expedience in the design of the resultant attributes and rules. We could afford this since our knowledge engineering approach has a second, refinement stage. By emphasizing accuracy later during the evaluation-driven refinement stage, we were able to build an intelligent classifier more efficiently and effectively.

4.5 System Development and Prototyping

Instead of presenting the design and implementation of the attributes to the experts and having them evaluate their validity using examples, we instead proceeded with prototyping ARKTOS. Indeed, the system development of ARKTOS started before the attributes were finalized. The image segmentation, conversion, and classification modules were all completely or partially built before the final rules and attributes were available. The software engineers solved all issues as closely as possible. Now, the experts would have to evaluate the prototype and refine the knowledge.

In the knowledge engineering framework, rapid prototyping is seldom useful without two paradigms. First, the prototype must be developed such that it is general and readily extensible. It should be modular so that any module could be replaced or tested individually. If one ends up having to overhaul the prototype considerably, then one loses the advantages of having built the prototype in the first place. Second, there must exist a convenient way of refining the prototype as it grows; otherwise, the prototype would not achieve its full functionality correctly. Even though the evaluation responsibility lies on the shoulders of the experts, the knowledge and software engineers should also provide ample support through GUI tools and documentation that details the design of the attributes and rules. For our ARKTOS project, we have created a software package to facilitate the refinement process, as discussed in the next section.

5 REFINEMENT STAGE

The goal of the refinement stage is to refine the knowledge base to improve the classification accuracy of the ARKTOS system. Towards that goal, we have built three GUI-based software tools in Java: (1) arktosGUI, (2) arktosViewer, and (3) arktosEditor. Each has a different functionality: arktosGUI concentrates on feature-based refinement of specific attributes and rules; arktosViewer deals with regional evaluation; and, arktosEditor has a rule indexing and search

mechanism and knowledge base editing capabilites. Figure 3 shows the incorporation of the three software tools into the refinement stage. First, the ARKTOS system generates classified images and logs of its reasoning processes. The user may use arktosGUI to review each feature visually, home in on a particular rule or attribute, and change the rules and thresholds using arktosEditor. For structural changes that require re-coding such as the design of an attribute, the information is passed along to the software engineers for another round of design and implementation. On the other hand, the user may use arktosViewer to mass-process a large number of images and concentrate on regional classifications. This allows the user to review and identify regions in which ARKTOS does well or poorly. The review is fed back into a knowledge refinement step where knowledge engineers identify what the problems are, e.g., poor segmentation, inappropriate rules, etc. Finally, the lessons learned are used in another round of design and implementation. The improved ARKTOS is then used to generate new classified images and the refinement cycle repeats.

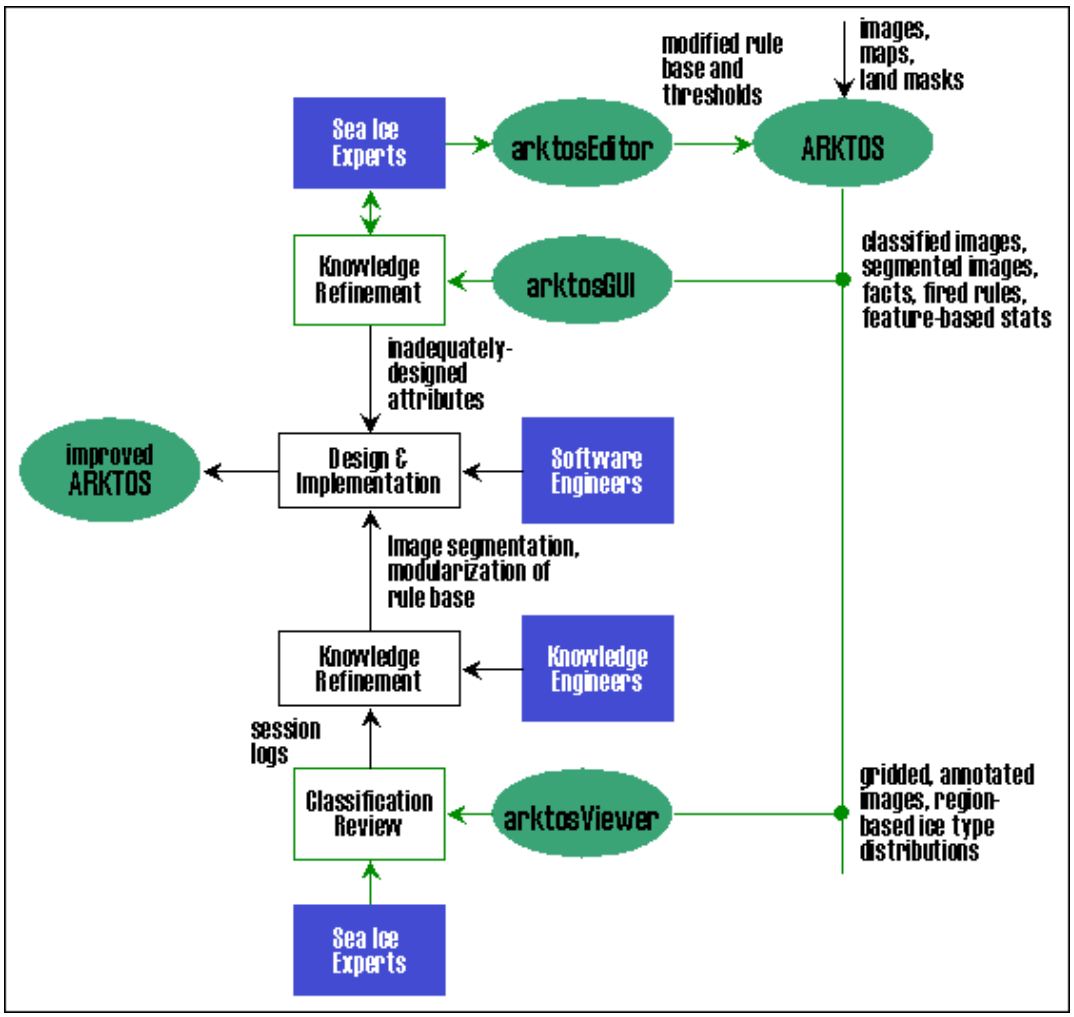


Figure 3 Stage 2 of our knowledge engineering process: refinement. The goal of this stage is to refine the knowledge by reviewing the classification results of the prototype. Thus, the refinement is evaluation-driven and quantifiable (in terms of improvements in the classification). In addition, to facilitate such refinement, we have created a software package with GUI-based and user-friendly tools.

5.1 arktosGUI

The objective of this module is to encourage feature-based knowledge refinement, by focusing the user's attention on individual features, specific attributes, and rules. Hence, this software is very useful for fine-tuning the ARKTOS rule base. arktosGUI has: the following abilities (1) feature-based inspection in which users can examine a specific ice feature (e.g., a floe) visually, symbolically, and numerically, and can also analyze the reasoning process that led to the classification of that feature, (2) compare-and-contrast of different ice features simultaneously, and (3)

attribute impact analysis in which users can view and examine, in color-coded images, the effects of their modifications of the attributes.

Figure 4 shows a flow diagram of arktosGUI. The work panel lists all images in the data directory, and has several functional buttons. The user may opt to view the segmented or the color-coded classified image, and the corresponding image will be displayed. The user may also choose to view the classification log of the entire image, in which the reasoning process for each feature is described. When the user clicks to view the original image, he or she can perform basic image enhancements such as brightening and histogram equalization or conduct our feature-based interaction. When the user prefers to perform attribute impact analysis, a window with a menu of all attributes is displayed. The user may choose from one of the attributes and the corresponding attribute image is displayed. We discuss further the feature-based interaction and the attribute impact analysis in the following subsections.

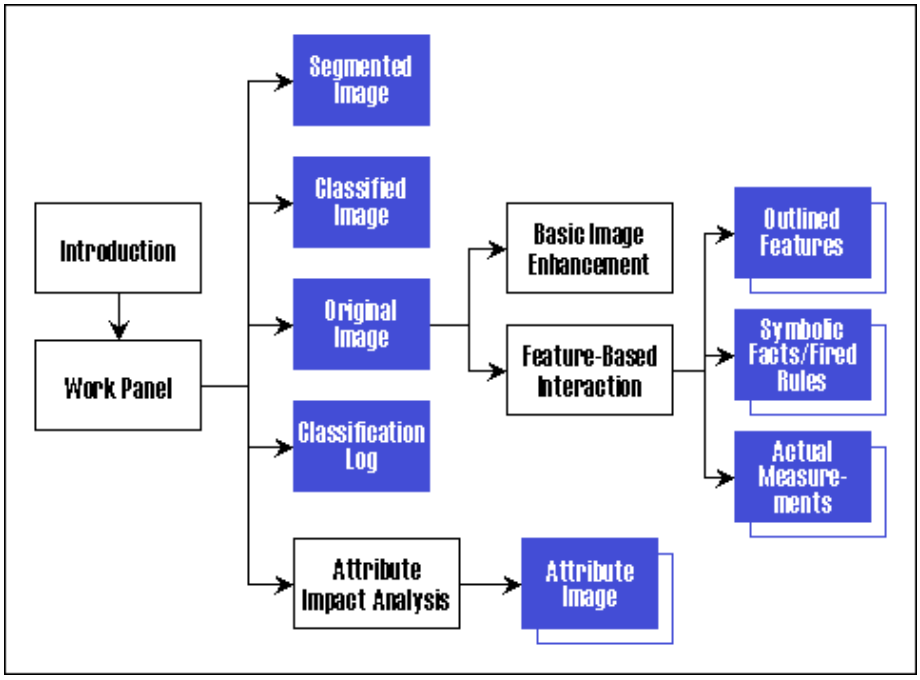
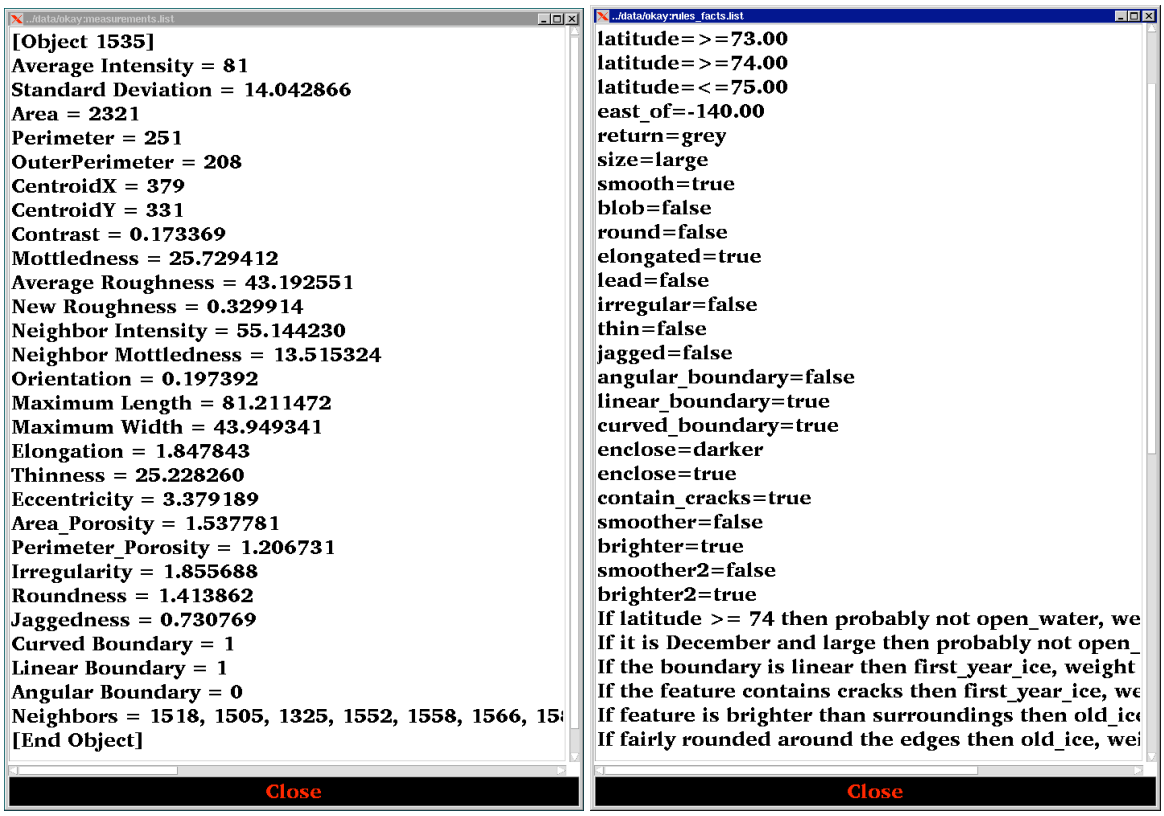


Figure 4 The flow diagram of arktosGUI. Feature-based interaction allows the user to select and view ice features individually. The work panel itself allows the display of multiple images simultaneously. The Attribute Impact Analysis and the Feature-Based Interaction both allow the display of multiple windows simultaneously, hence the box shadow on attribute image, outlined features, etc.

5.1.1 Feature-Based Interaction

When the user clicks on a pixel on the original image, arktosGUI outlines the ice feature that encompasses the pixel, and simultaneously displays two textual windows. One window contains

the raw, numeric measurements associated with the selected ice feature (see Figure 5a). The other, symbolic, window contains the symbolic facts of the measurements and other sources of information such as SSM/I maps, land masks, historical climatology maps, and ancillary data (see Figure 5b).



a **b**

Figure 5 a. The window that displays the actual measurements of a feature. b. The window that displays the symbolic facts of a feature, the rules that have fired during the classification process, and the classification.

In addition, the symbolic window lists the rules that have fired leading to the classification of the particular ice feature and the log of the combined mass of evidence and the belief and plausibility measures of the classification. This facility allows the user to visually inspect what a particular ice feature looks like and how ARKTOS numerically and symbolically describes that feature. Thus, the user can compare ARKTOS' views to his or hers. For example, if the user thinks that the feature is not elongated but ARKTOS thinks so, then the user might want to change the design of the "elongatedness" measurement and its thresholds. In addition, the symbolic window

allows the user to see how the brain of ARKTOS works by following its trail of fired rules and can identify problematic rules for potential editing. The user is also able to compare multiple features by clicking on different pixels on the original image and viewing their textual windows simultaneously. Figure 6 shows an example of multiple highlighted features.

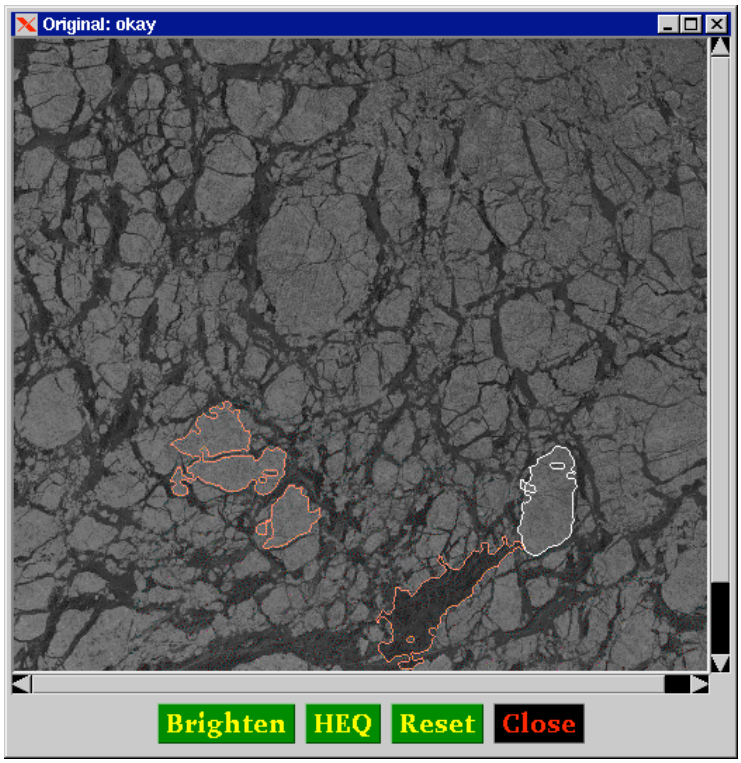


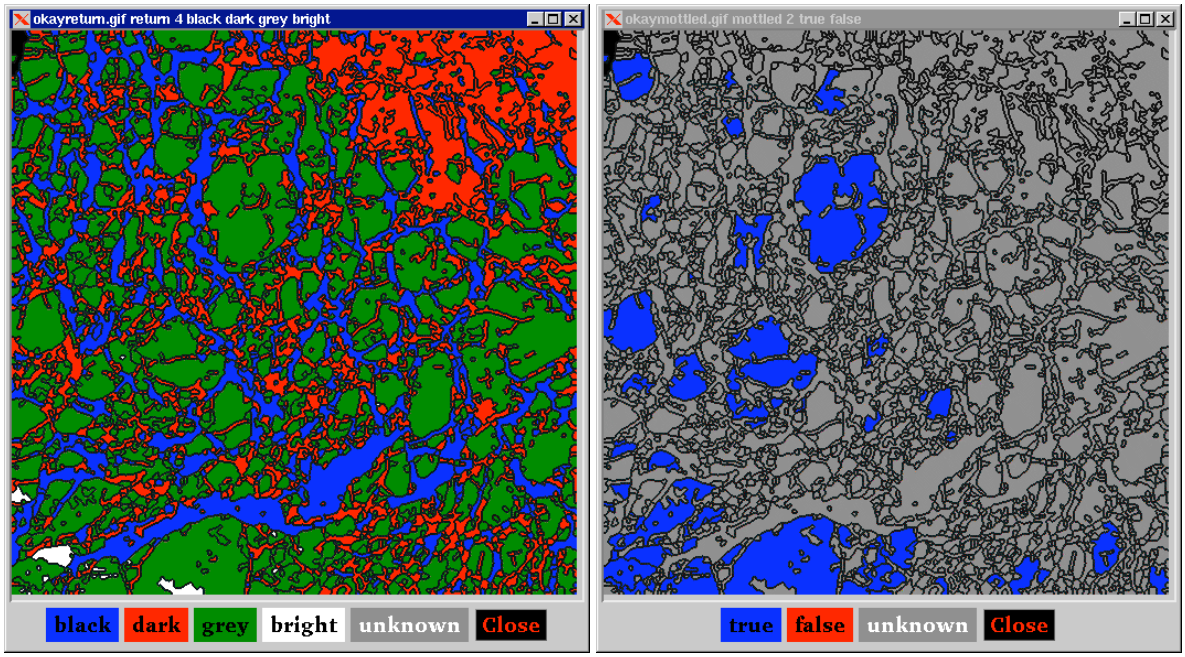
Figure 6 Feature-based interaction allows the user to select and view ice features individually. The window shows four outlined features. The feature with the white boundary is the most recently selected feature.

The feature-based interaction has been utilized intensively to refine the threshold values used to convert numeric measurements into symbolic facts and the creation and modification of our classification rules.

5.1.2 Attribute Impact Analysis

This software feature allows the user to select a particular attribute and view the distribution of its attribute values of all features in the image. For example (see Figure 7a), suppose the user selects "return", the average intensity of an ice feature, which takes four values: black, dark, gray, and bright. arktosGUI subsequently displays a feature-based image, with each feature colored differently: all "black" features are blue, "dark" features are red, "gray" features are green,

"bright" features are white, and features with "unknown" returns are gray. See Figure 7b for a similar example on "mottled" – the description of the surface texture of a feature. This powerful feature enables the user to perform an effective review of the attribute's correctness. At a glance, the user is able to detect features that are colored differently from what he or she has expected. The user can then change the thresholds of that particular attribute, re-run the ARKTOS program, and view the impact of that change on the screen. This instant feedback allows the user to focus on and fine-tune a particular threshold value interactively.



a **b**

Figure 7 Two examples of ARKTOS' attribute impact analysis feature. (a) the attribute is "return," the average intensity of a feature, and it has four possible values (black, dark, gray, and bright) and unknown, (b) the attribute is "mottled," the surface texture of a feature.

5.2 arktosViewer

The objective of this module is to facilitate a region-based evaluation, helping the user concentrate (1) on regions instead of individual ice features and (2) on ice type distributions instead of the classification of a particular feature. Therefore, this module is important in broad-based evaluation of the ARKTOS system and in determining the types of images ARKTOS classifies well or poorly. arktosViewer has (1) a region-based inspection in which users can examine annotated, gridded areas to assess whether the classification results are acceptable, (2) a bookkeeping of the user's evaluation sessions, and (3) a text-based recording of the user's additional

comments. The gridded, annotated image is an image divided into 16 regions and annotated with ice type distributions for each region, and is generated by ARKTOS.

Figure 8 shows the flow diagram of arktosViewer. The work panel lists a set of images in the data directory and allows the user to proceed to the evaluation. When the evaluation button is clicked, arktosViewer automatically pops up four windows to display the segmented image, the classified image, the annotated, gridded image, and a note recorder. The user then may interact with the gridded image to record its visual evaluation of the regions. The interactions are recorded in the action log for each individual image. The note recorder is a free-text window for the user to type in additional comments. When the user is done with the comments, the text is saved and tagged with the particular image under evaluation. The action log and the comments serve as a feedback to the knowledge and software engineers for improving ARKTOS.

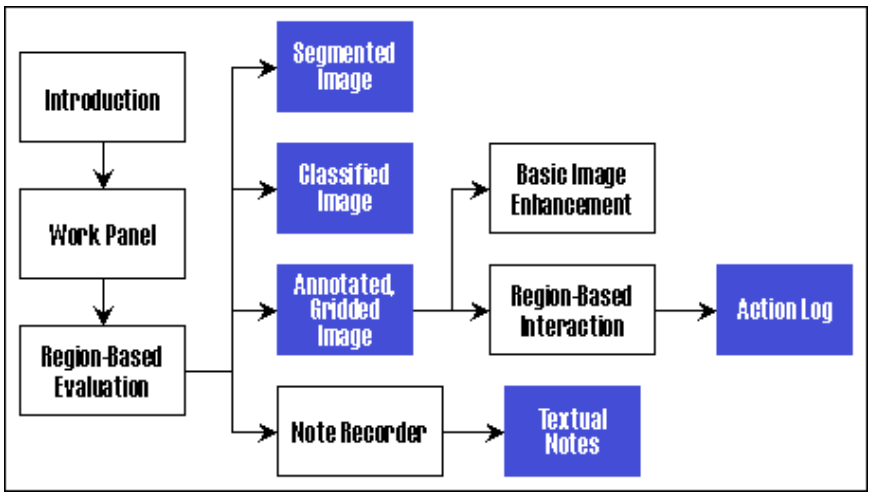


Figure 8 The flow diagram of arktosViewer. Textual notes contain additional observations by the experts. Action log contains the mouse-click actions of the user designating regions with unacceptable classification, etc.

Note that arktosViewer has been designed to facilitate mass evaluations of classification results of ARKTOS. Therefore, it displays the segmented and classified images at a smaller scale automatically, has only a reduced set of tools, and does not perform any on-line computations.

The user may survey the gridded and annotated image and, if the classification is completely acceptable or unacceptable, simply click the “acceptable” or the “unacceptable” buttons, respectively, to submit the evaluation. If the classification in some regions of the image is acceptable and in some not, then the user may click on the regions that he or she thinks are incorrectly classified. The regions will then be highlighted, and the user is allowed to de-select a re-

gion anytime they want. After the user is done with the evaluation, he or she may click on the "submit" button to submit it. All actions submitted to arktosViewer are time-stamped and written to the Action Log associated with each image. Figure 9 shows an example of arktosViewer in action.

5.3 arktosEditor

The main goal of arktosEditor is to allow better editing of the knowledge base. This module has (1) rule editing and threshold editing capabilities, and (2) a rule indexing and search mechanism. It features indexed, attribute-based, and conditioned search and filtering, editing of all aspects (ID, conditions, weights, classifications, and text description) of a rule, and editing of the thresholds used to convert numeric measurements to symbolic facts. Hence, this module is important in organizing and modifying our knowledge bases.

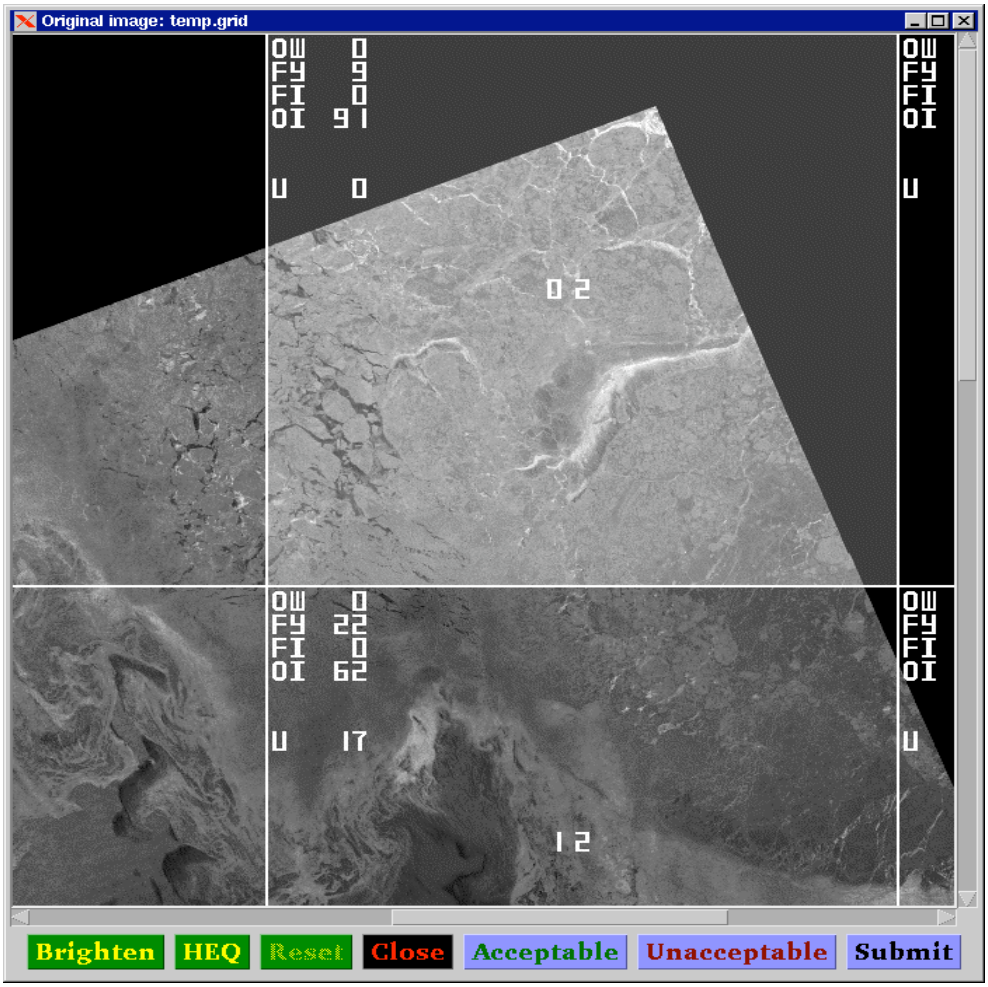


Figure 9 An example of arktosViewer. The selected (highlighted) region is 02. ARKTOS has classified the region to 0% open water (OW), 9% first year ice (FY), 0% fast ice (FI), 91% old ice (OI), and 0% unknown (U).

Figure 10 shows the flow diagram of arktosEditor. The work panel allows the user to delete, rename, copy, or edit a selected file. The selected file can either be a theshold file (where attribute thresholds are stored) or a rule-base file. arktosEditor recognizes the two different files by their naming suffixes. If the selected file is a threshold file, arktosEditor immediately displays its threshold editor window. If the selected file is a rule-base file, arktosEditor pops up a search window in which the user is able to select search keys to obtain the set of rules that he or she wants to edit. Then, arktosEditor brings up the rule editor window. We further discuss the rule and threshold editors, and the rule search in the following subsections.

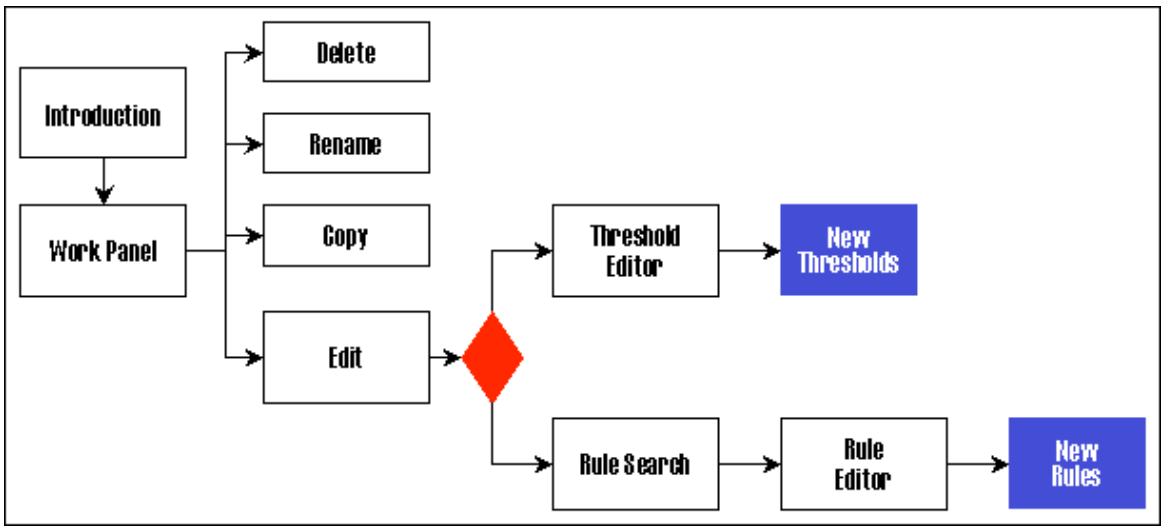


Figure 10 The flow diagram of arktosEditor. The user is allowed to delete, rename, copy, or edit a threshold file or a rule-base file. When a threshold file is selected, the threshold editor is invoked. When a rule-base file is selected, the user is able to enter a search for a particular set of rules to edit.

5.3.1 Threshold Editor

The threshold editor allows the user to modify the threshold values of the attributes. Figure 11 shows a threshold editor window. As shown in the figure, there are 12 sets of attribute thresholds. If the user wants to modify the threshold values of the attribute "size", he or she clicks on the "size" button to display the comments associated with the attribute, and may then change the thresholds. When done, the user simply clicks the "Save" button.

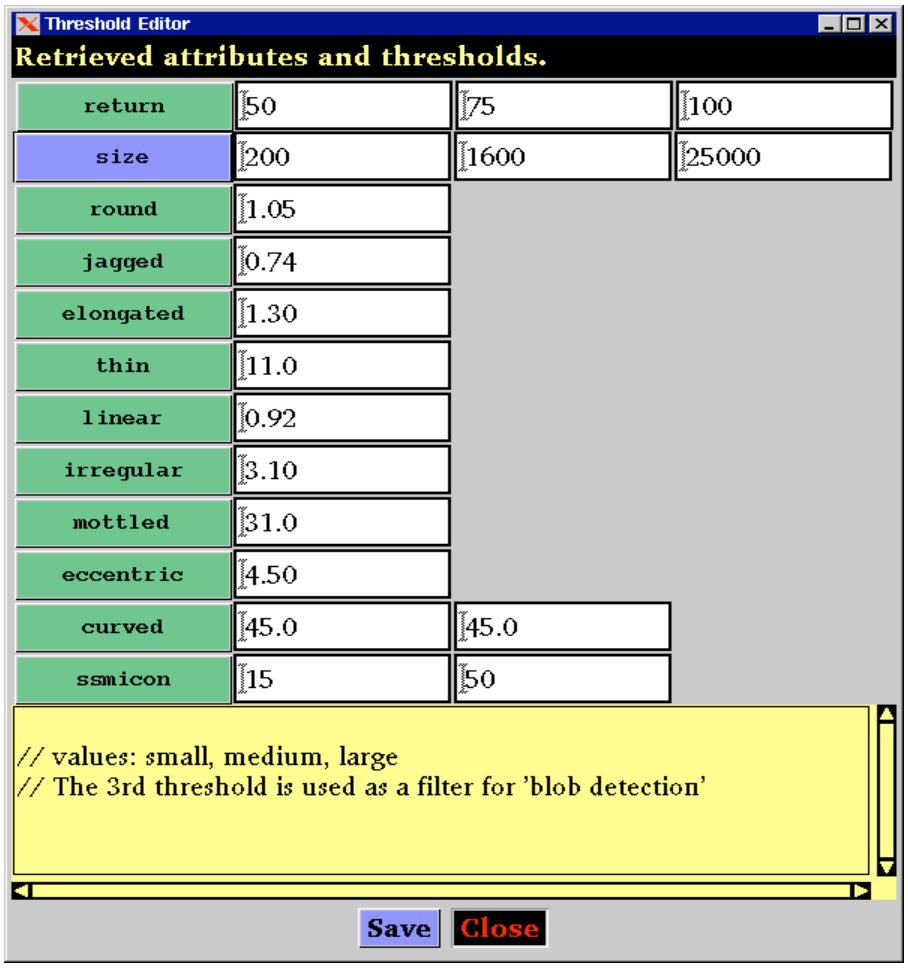


Figure 11 An example of the threshold editor window of arktosEditor.

5.3.2 Rule Search

Figure 12 shows an example of the rule search window. The rule search window first requires the user to specify which sets of rules of the rule base he or she wants to edit. Our current rule base consists of two sets of rules: summer and winter. Then, the user may retrieve a particular rule by supplying its identification number, or a group of rules based on a set of search keys. There are three different search keys. First, the user may search by the ice class. One may choose to retrieve all rules that are related to the ice class "old_ice", for example. Second, the user may search by the weights of the rules. One may compose an interval by using the comparison operators (=, >, ≥, etc.), the binary operators ("and" or "or"), and two real numbers. For example, one may want to retrieve all rules with weights between 0.4 and 0.6. Finally, the user may search by the attributes. Once again, the user may compose the search key by using one or two attributes, and with a binary operator ("and" or "or"). For example, one may want to re-

trieve all rules with the attribute “round”. The user is also able to toggle off the search keys by de-selecting their checkboxes on the screen. So, suppose the user wants to retrieve all winter rules: he or she simply selects the “winter” season, toggles off all search keys, and proceeds with the search.

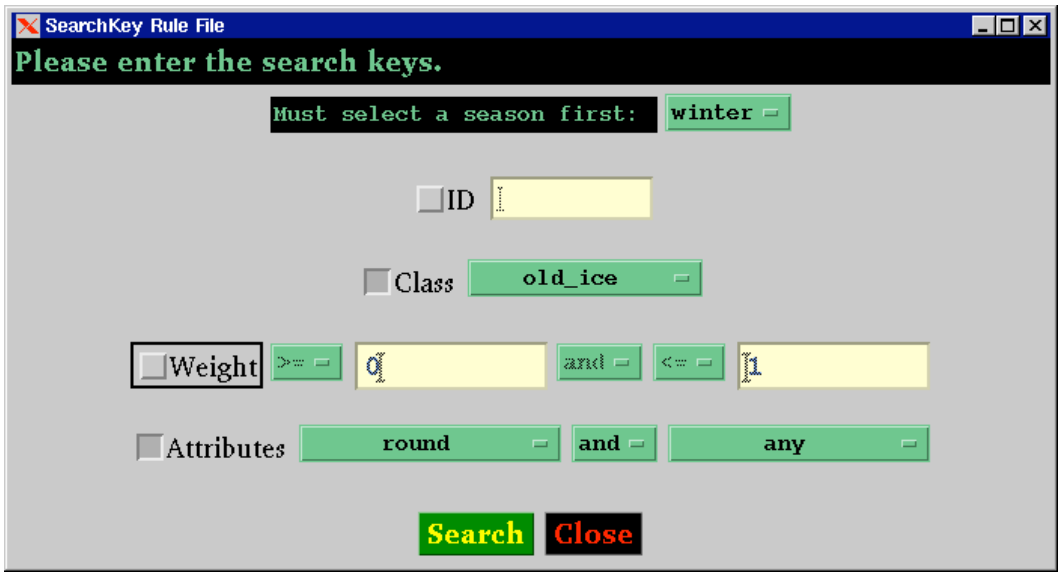


Figure 12 An example of the rule search window of arktosEditor.

5.3.3 Rule Editor

After the rule search, arktosEditor pops up the rule editor window, shown in Figure 13. The rule editor allows the user to delete an old rule, modify an existing rule, and create and save a new rule. It also supplies a count of the rules retrieved. Figure 13 shows a list of 6 retrieved rules, based on the search keys used in Figure 12. The rule with ID = 65 has been selected. To activate the rule editing panel (the bottom portion of the window), the user must either click the “Edit” or the “Create New” buttons.

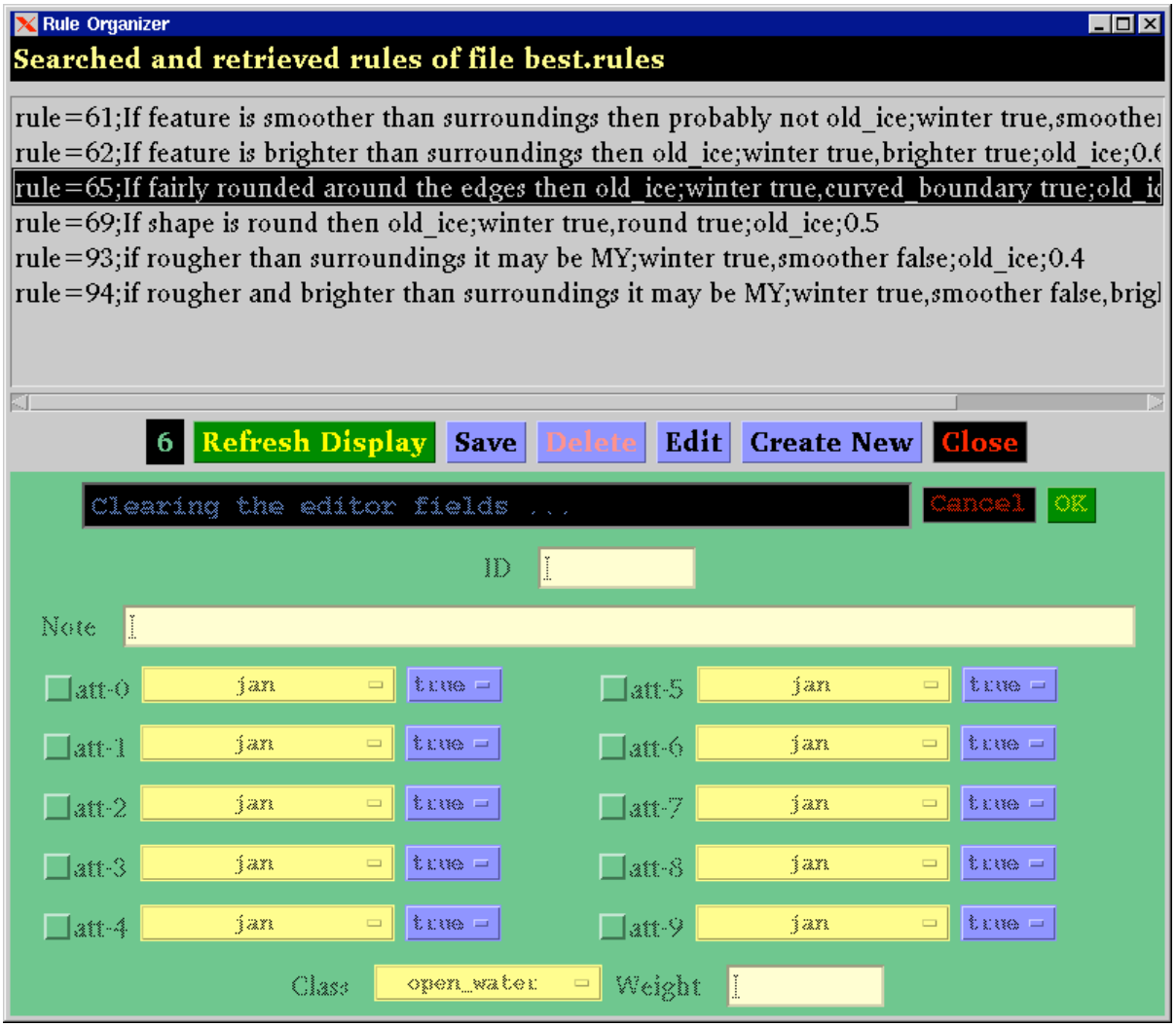


Figure 13 An example of the rule editor window of arktosEditor.

When the user selects a rule and clicks the "Edit" button, arktosEditor activates the editing panel. All fields are initialized to the values of the selected rules: ID, Note, Class, and Weight. The attributes used are also checked. For example, in Figure 14, the rule retrieved has three attributes: "winter", "curved_boundary", and "smooth". Thus, except for att-0, both att-1 and att-2 are highlighted initially. The user may change the attribute values, reduce the number of attributes, or add a new attribute; for example, figure 14 shows that a new attribute, "enclose", has been activated and a corresponding value, "brighter", has been selected. Each attribute has a different set of possible values. As soon as the user selects an attribute from the menu (the yellow bar), the editor automatically retrieves the set of values and makes them available to the user

(the blue bar). Currently, there are about 50 attributes in ARKTOS and the maximum number of attributes allowed for each rule is set at 10.

The arktosEditor relieves the user from having to memorize or look up some manuals to obtain the set of possible values for each attribute and from having to type in the rules manually, facilitating a more convenient and efficient evaluation and refinement process.

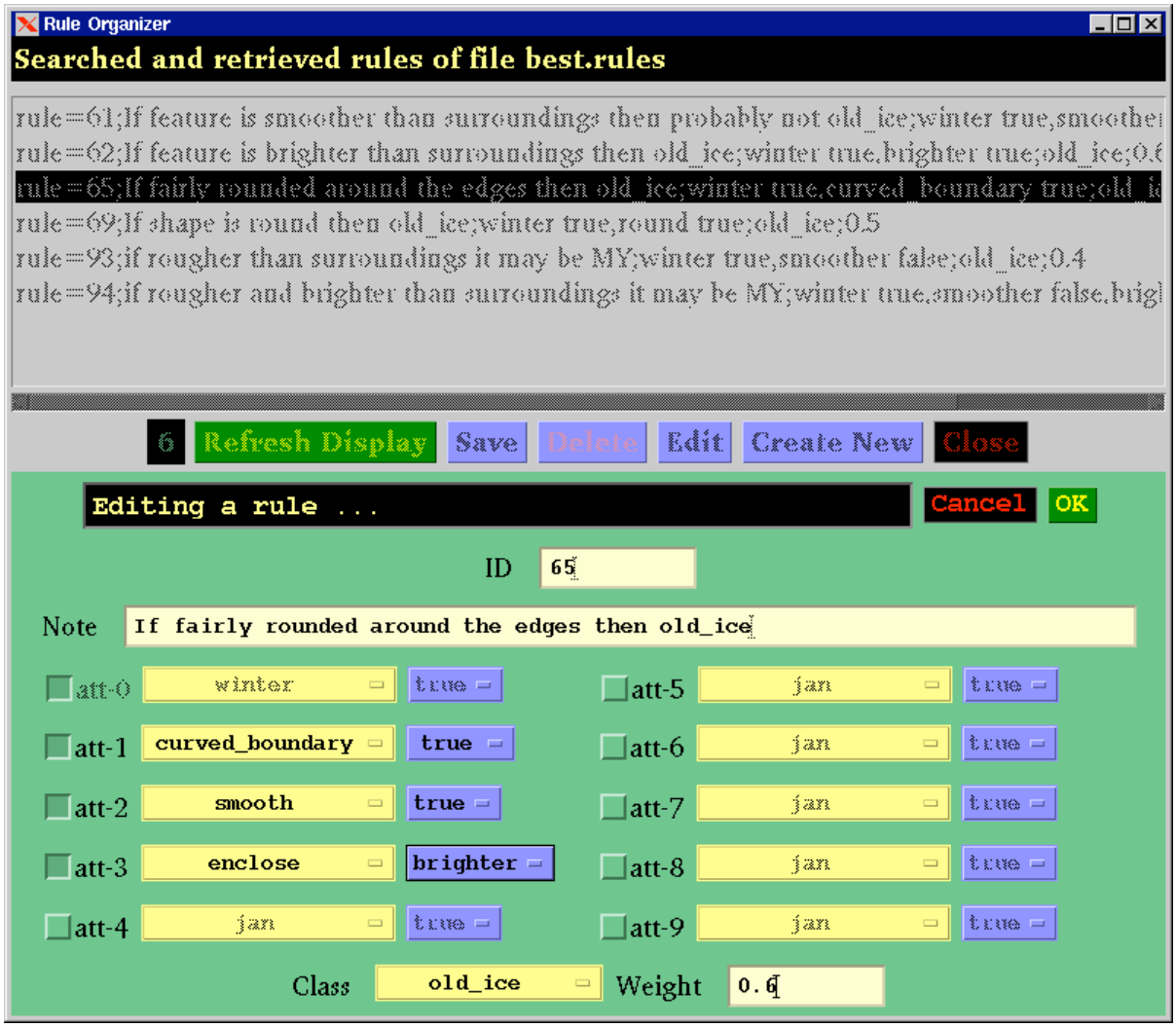


Figure 14 An example of the rule editor window of arktosEditor with an activated editing panel. The attribute "winter" (or "summer") is always de-activated since it is mandatory. Similarly, when one creates a new rule, the attribute "winter" (or "summer") will always be included automatically.

6 EVALUATION

In this section we document some of the studies that have been undertaken by sea ice image analysts using arktosGUI, arktosViewer, and arktosEditor, and their evaluation of how these tools helped them refine and evaluate the classification accuracy of ARKTOS. The evaluation is based both on anecdotal evidence (in the form of case studies) and on quantitative measures.

6.1 arktosGUI

The arktosGUI tool was the one used the most during the refinement and evaluation phase. Its abilities to display classification characteristics and image features proved invaluable in understanding *why* ARKTOS classified features a certain way, and how to correct misclassifications.

A sea ice expert from Veridian Corp., associated with the Naval Research Laboratory (NRL) and stationed at the John Stennis Space Center, MS, studied 28 images covering the area of operation of ARKTOS at marginal ice zones in Beaufort Sea and Bering Sea using images taken from October 1997 to May 1998. Through arktosGUI she was able to conclude that the predominant misclassification by ARKTOS was is the identification of multiyear (MY) ice as first year (FY) ice. She pinpointed the cause for the misclassification in the segmentation process of ARKTOS, by viewing the individual segmented features and examining their actual measurements and symbolic facts. This led to the introduction of a new feature, "blob," to represent very large, amorphous areas of sea ice which had to be measured and treated differently from usual floe-sized ice objects. The introduction of the "blob" object reduced the misclassification of MY ice, and was possible by the use of arktosGUI[1].

The expert also used arktosGUI to identify that another primary misclassification was the assignment of bright open water as multiyear ice, since both features tend to have high intensity. This led to the fusion of SSM/I satellite classified data into ARKTOS, since SSM/I tends to reliably classify open water.

The arktosGUI also helped sea ice experts fine-tune rules and attribute thresholds, as well as identify the need for new attributes and rules. For example, the Veridian Corp.'s expert used arktosGUI to fine-tune the usage of "area_porosity" and "perimeter_porosity." Briefly, if an object contains holes or cracks, it has a high "perimeter_porosity" (the ratio of total perimeter

[1] All information regarding the Veridian Corp.'s expert's evaluation of ARKTOS is based on personal communications.

length over the outer perimeter length). If an object is long, stringy and branchy, then it has a high "area_porosity" (the ratio of the area of the bounding rectangle over the true area of the object). Visual inspection and comparison allowed the expert to realize that FY ice would have higher "area_porosity" values while MY ice would have higher "perimeter_porosity," leading to changes in the appropriate values and thresholds, and, subsequently, to improved classification.

The expert also used arktosGUI to introduce new measurements for texture, after observing that the old texture measurements did not always capture the expert's intent and knowledge. Specifically, when the first ARKTOS rule base was built, the experts agreed that a feature that enclosed darker features was most likely a multiyear ice feature. However, through arktosGUI, given the outlined features and their respective measurements and facts, the expert was able to realize that the rule was not accurate. In fact, multiyear ice features were found to usually enclose features that could be darker or brighter, while first year ice features were found to usually enclose only brighter features. This previously unnoticed piece of knowledge was encoded in our feature set and rule base.

Two other sea ice experts associated with the Canadian Ice Service (CIS), in Ottawa, Canada, have also conducted a comprehensive evaluation of ARKTOS. Through the use of arktosGUI they were able to identify erroneous classifications, subsequently realizing how to fix some of the problems in ARKTOS by manipulating thresholds and rules, and eventually obtaining better classification accuracy. The experts found out that thresholds of attributes seemed to have more influence than the weighting of the rules. By looking at the measurements, facts and rules, the experts found that the influential attributes for the first year ice and old ice classes were "mottledness" (a surface texture measure) and "intensity". The experts modified these thresholds and were able to improve the classification accuracy of four problematic images by 16%. By inspecting the numeric, symbolic, and visual data shown on arktosGUI, the experts further compiled a list of recommendations, some of which are listed on Table 1. Reading the recommendations, we observe the following utility of arktosGUI: (1) it was used effectively by the experts to fine-tune ARKTOS since they were able to pin-point exactly what the reasoning process was that led to the classification of each sea ice object, (2) it was used by the experts to match visual cues to symbolic facts—linking what they saw cognitively to what the system saw algorithmically, and, most importantly, (3) it was used by the experts to perform knowledge engineering, to actually recommend better designs of, for example, contrast computation, segmenta-

tion, numeric relative facts, varying degrees of mottledness, and so on. This last utility motivates highly effective application- and task-driven knowledge transfer between the experts and software engineers. That is, given the recommendations, the software engineers know exactly and precisely what changes to add to ARKTOS (Chowdhury and Wilson 2001).

Recommendations
Measures should be taken to consider low contrast images in comparison to contrast images. For example, in a low intensity image a difference of 4 gray levels would be significant compared to a high contrast image.
A fixed threshold should not be used to distinguish between black, dark, gray, and bright. The same intensity in different images can belong to different ice classes. Dynamic threshold can be applied.
Techniques can be applied to preserve the visual shape of the feature. As an example, if the feature is round visually but its boundary is not closed in the segmented image, some technique can be applied to look around the feature's boundary and try to close the boundary if possible to give it a clear shape.
When deriving the relative facts like round, brighter, smoother, darker, enclose brighter, measurements should be taken to consider the visual observation. As an example how much intensity difference should be there to say brighter. Is it .5, 5, 10 or 50?
Rules should be refined to distinguish between Open Water and New Ice.
Rules should be refined to distinguish between thick First Year Ice and Old Ice.
Instead of two sets of rules, four sets could be created which will include melt and freeze up periods.
Variable weights could be introduced when similar attributes are present in more than one class. As an example, mottledness can be present in FYI and in OI.

Table 1 Some recommendations by the CIS experts for the initial evaluation and assessment of ARKTOS.

The expert from Veridian Corp. was also responsible for some evaluation and maintenance of the ARKTOS' rule base. She has devised a procedure for systematic fine-tuning of rules and attributes using ARKTOS' GUI modules. To modify a rule, she inspected five to ten features in an image, for a set of ten to twenty images. The procedure consisted of (1) inspecting the classification results, (2) looking at the rules and attribute measurements for each feature, (3) modifying the rule as needed, (4) re-running ARKTOS, (5) repeating steps (1) and (2) until satisfactory results were obtained. Table 2 documents the average time spent to accomplish the above procedure with and without the GUI modules. The table was for inspecting an average of five features in an image, for a total of 10 images, and for only one iteration of fine-tuning (note that NSIPS is the Navy's image display program). The times for "without GUI" in Table 2 are estimates provided by the expert based on her experience. As can be seen, the use of arktosGUI offered approximately one order of magnitude improvement in productivity.

Tasks	With GUI (min)	Without GUI (min)
Look at the classifications to see good and bad areas with NSIPS	Automatically presented	50
Look at the segmentations to see where the problem features are	Automatically presented	100
Look at the raw image inside of NSIPS to find the center latitudes and longitudes of the features	Automatically linked	150
Examine the big source measurement files to identify the features, remembering the latitude and longitude from NSIPS is merely an estimate	Automatically linked	1000
Match the object number to that in the source rule file and see what the rules are doing	100	500
Modifying the rule as needed	5	10
Re-running ARKTOS	20	20
Look at the classifications again	100	750
Total	**225**	**2580**

Table 2 Time performance of modifying a rule with and without ARKTOS' GUI modules.

6.2 arktosEditor

The arktosEditor module was used sparingly, since most of the effort involved finding out what was wrong in classifications, and actual rule editing was a small part of the refinement. As can be seen from Table 2, the arktosEditor would offer small gains ("Modifying rule as needed" would take an estimated 5 minutes instead of 10 minutes). Where arktosEditor offered major gains was in enforcing rule consistency and avoiding typos. In August 1998 a preliminary testing and analysis of ARKTOS showed that unreasonably much of the multiyear ice on an image was misclassified as first-year ice. An investigation of the ARKTOS rule base revealed that there were spelling errors that prevented rules from getting fired. At that time editing of the rule base was performed manually through a text editor. This experience led to the development of arktosEditor which enforces consistency in rule creation and editing. This indirectly enhances the overall refinement effort, since the experts no longer have to worry about the syntactic correctness of the rules, and can concentrate on the knowledge itself.

6.3 arktosViewer

The arktosGUI proved good for detailed evaluation of the classification of specific features and the fine tuning of rules and feature measures, but sea ice analysts working for the U.S. National Ice Center and the Canadian Ice Service are used to viewing classifications of sea ice as percent-

ages of areas. The arktosViewer allowed analysts to view classified images in a manner that they considered natural and which was consistent with their job training. A sea ice expert associated with the National Ice Center, MD., performed mass-processing of sea ice images using ARKTOS and evaluated the classification results using arktosViewer. Table 3 documents some of her comments (based on personal communications). Note that by dividing the image into quadrants and annotating the ice distribution, the expert was able to make quite a detailed ground-truth inspection of ARKTOS' classification results. As we can see from Table 3, the evaluation here concentrates on regional or overall classification and ice distribution instead of feature-based analysis.

Image ID	Comments
d00211t144236wss	Image is of the Canadian Archipelago, where the ice is about half and half FY and MY, but ARKTOS is identifying it as all MY.
d00211t144428wss	Image has edge and reduced concentrations, and the ARKTOS retrievals agree well in both total ice concentration and in partial concentrations for ice type
d00211t162407wss	Image has land in it, but is located in high arctic pack, which is nearly all MY, but ARKTOS is underestimating the quantity of MY and overestimating the FY concentrations.
d00211t16251wss	Image is 100% wet ice of mixed types, but ARKTOS is calling it all open water
d00212t173456wss	Image contains high arctic pack, which ARKTOS recognizes properly as MY
d00218t162100wss	Only a strip, but the ice concentration and type are very properly categorized
d00220t032721wss	ARKTOS yields good total ice concentrations where coverage is not 100%, but misidentifying the MY ice as FY ice.

Table 3 Some evaluation comments reported by a sea ice expert using arktosViewer to evaluate the ARKTOS' classification result.

6.4 Evaluation of the Classification of ARKTOS

The Canadian Ice Service (CIS) performed an extensive quantitative evaluation of ARKTOS' classification accuracy. They used ten different image dates which comprised of 19 frames from the Beaufort Sea near Banks Island. Figure 15 shows how the CIS overlaid its ice chart on top of the ARKTOS classified image. Next, they visually compared classifications on a polygon-by-polygon basis. In each polygon the CIS and ARKTOS ice types and amounts were approximated to the nearest tenth, and the results were then compared to what percentage of the image areas the polygon contributes to (Chowdhury and Wilson 2001).

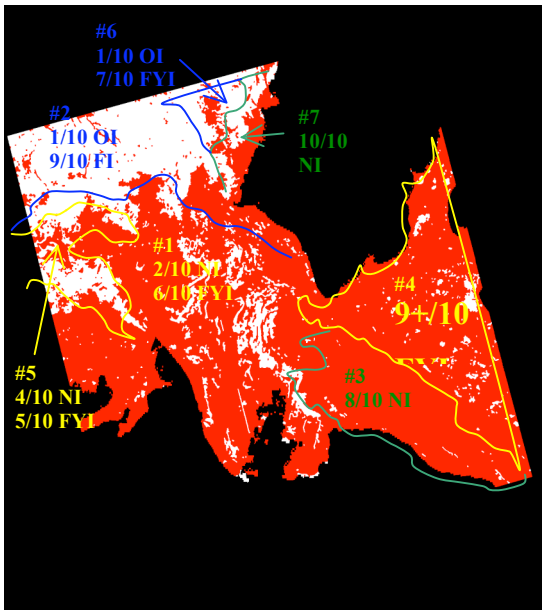


Figure 15: CIS ice charts manually overlaid on ARKTOS classified image to allow quantitative comparison Reproduced from (Chowdhury and Wilson 2001).

The average ARKTOS success rate during this test was approximately 74%, and ranged between 95% and 30%. Excluding the four worst classifications, ARKTOS' average performance was 80%. The CIS evaluators changed some of the ARKTOS thresholds (as described in section 6.1) and improved the average ARKTOS classification accuracy to 83% over the whole test image set.

7 CONCLUSIONS

Knowledge engineering is a very important first step to building an intelligent system. Key phases such as knowledge acquisition, representation, and refinement are difficult and more so when images are involved. In our approach to building the ARKTOS intelligent sea ice classification system, instead of acquiring accurate knowledge to begin with, we performed a rapid prototyping and used the evaluation of that prototype to drive knowledge refinement. In this way, our knowledge and software engineers provided a prototype quickly to demonstrate the system's functionality, to generate quantifiable results, and to optimize the programs for better speed and memory usage, resulting in a faster turnaround time. On the other hand, the experts could use the prototype and its support tools to refine the knowledge, interface with the engineers on a common platform, and improve the system with observable feedback.

Since the refinement of ARKTOS was evaluation-driven, the experts could see how each rule they modified or added behaved and interacted with the other rules in the rule base. Without the prototype, the experts would not have been able to predict the intricate interactions among the rules—at least not without considerable effort. In addition, since the refinement was evaluation-driven, the system improved with each change. From a psychological standpoint, the experts had a chance to tinker with their knowledge and were explicitly motivated as they observed the classification improvements. From an engineering standpoint, the two-stage approach reduced significantly the interaction time between the experts and the engineers. The approach released both parties to their other responsibilities, making it a human resource-friendly option.

The image processing technology has its limitations as images are noisy and visual cues are difficult to model. Further, the reasoning process of sea ice experts when classifying an object into different sea ice classes is complex and cannot be fully encoded. The resultant ARKTOS software does not reflect completely accurately what sea ice experts see, process, and analyze. Therefore, our software tools also play the role of training the experts or users to understand the strong points and weaknesses of ARKTOS' underlying methodology.

The sea ice experts were forced to become adept at using the tools which, in turn, led them to understand how ARKTOS worked internally. This took some time, but the time was shortened by the various knowledge elicitation and refinement tools. It was also a more efficient process compared to continuous interactions between knowledge engineers and domain experts, where both parties do not understand each other's vocabulary and processes. As quoted by one expert, "One can look at the features and see the differences, but in some cases the difference is difficult to measure given what the algorithm now calculates. Little by little, I've been developing some new rules which define the difference in terms of what the algorithm 'knows'." (personal communication). This showed us that our software modules are useful knowledge engineering tools not only to bring the technology closer to the experts, but also to bring the experts closer to the technology.

ARKTOS has been evaluated by the Canadian Ice Service and is currently undergoing further testing by the U.S. National Ice Center. Currently it is operating at approximately 80% accuracy, compared to human-generated ice charts. Based on our discussions and interactions with the experts who tested, refined, and modified ARKTOS, we have also shown that the suite

of ARKTOS-related tools (arktosGUI, arktosEditor, and arktosViewer) greatly improved the productivity of the experts, and made the final system possible in a small period of time.

In conclusion, we have described a knowledge engineering approach that is particularly useful in developing intelligent image analysis systems. We have presented a suite of GUI-based software tools that helps the experts and the knowledge and software engineers in refining and improving systems. In addition, the designs and techniques used are applicable to other image domains for capturing visual cues and semantics as well. Therefore, our knowledge engineering approach and tools offer an efficient and effective methodology to developing machine intelligence in computer vision applications.

8 ACKNOWLEDGMENTS

We would like to thank Denise Gineris, Mary Ruth Keller, Mohammad Sharif Chowdhury, and Katherine Wilson for their continuing evaluation of ARKTOS and its knowledge engineering software package. We would like to thank Todd Bowers, Andrew Williams, John Gauch, Hsin-yen Wei, and Yanning Zhu for their participation in the prototyping stage of ARKTOS, and Cheryl Bertoia of the National Ice Center, Ginette Leger, Denis Lambert, and Dean Flett of the Canadian Ice Service, and Kim Partington of NASA for their sea ice expertise. This research work was supported in part by a Naval Research Laboratory research grant, contract number N00014-95-C-6038.

9 REFERENCES

Addis, T. R., D. C. Gooding, and J. J. Townsend (1993). Knowledge Acquisition with Visual Functional Programming, in Proceedings of the 7th European Knowledge Acquisition Workshop (EKAW'93), Toulouse and Caylus, France, September 6-10, 379-406.

Boose, J. H. (1986). ETS: A PSP-Based Program for Building Knowledge-Based Systems, in Proceedings of the IEEE Western Conference of Knowledge-Based Engineering and Expert Systems (WESTEX-86), Anaheim, CA, June 24-26.

Boose, J. H. (1993). A Survey of Knowledge Acquisition Techniques and Tools, in Readings in Knowledge Acquisition and Learning, Buchanan, B. G. and D. C. Wilkins (eds.), San Mateo, CA: Morgan Kaufmann, 39-56.

Boose, J. H. & Bradshaw, J. M. (1987). Expertise Transfer and Complex Problems: Using AQUINAS as a Knowledge Acquisition Workbench for Knowledge-Based Systems, International Journal of Man-Machine Studies, 26(1), 3-28.

Boose, J. H., Shema, D. B., & Bradshaw, J. M. (1995). Recent Progress in AQUINAS: A Knowledge Acquisition Workbench, Knowledge Acquisition, 1(1), 185-214.

Charlet, J., Krivine, J.-P., & Reynaud, C. (1992). Causal Model-Based Knowledge Acquisition Tools: Discussion of Experiments, in Proceedings of the 6th European Knowledge Acquisition Workshop (EKAW'92), Heidelberg and Kaiserslautern, Germany, May 18-22, 318-336.

Chowdhury, M.S., & Wilson, K.J. (2001). Initial Evaluation and Assessment of ARKTOS 1.7, Technical Report, Canadian Ice Service, August.

Cheng, P. C.-H. (1996). Diagrammatic Knowledge Acquisition: Elicitation, Analysis and Issues, in Proceedings of the 9th European Knowledge Acquisition Workshop (EKAW'96), Nottingham, UK, May 14-17, 179-194.

Dieng, R., Gibooin, A., Tourier, P.-A. & Corby, O. (1992). Knowledge Acquisition for Explainable, Multi-Expert, Knowledge-Based Design Systems, in Proceedings of the 6th European Knowledge Acquisition Workshop (EKAW'92), Heidelberg and Kaiserslautern, Germany, May 18-22, 298-317.

Eisenstadt, M. & Brayshaw, M. (1990). A Knowledge Engineering Toolkit (part II), Byte, 15(11), 364-370.

Eisenstadt, M., Domingue, J., Rajan, T. & Motta, E. (1990). Visual Knowledge Engineering, IEEE Transactions on Software Engineering, 16(10), 1164-1177.

Gaines, B. R. & Shaw, M. L. G. (1980). New Directions in the Analysis and Interactive Elicitation of Personal Construct Systems, International Journal man-Machine Studies, 13, 81-116.

Gaines, B. R. & Shaw, M. L. G. (1987). Knowledge Support Systems, in Proceedings of the ACM MCC-University Research Symposium, Austin, Texas, 47-66.

Gaines, B. R. & Shaw, M. L. G. (1993). Eliciting Knowledge and Transferring It Effectively to a Knowledge-Based System, IEEE Transactions on Knowledge and Data Engineering, 5(1), 4-14.

Hart, A. (1992). Knowledge Acquisition for Expert System, 2nd Ed., New York: McGraw-Hill.

Kahn, G., Nowlan, S. & McDermott, J. (1985a). MORE: An Intelligent Knowledge Acquisition Tool, in Proceedings of the IJCAI-85, Los Angeles, CA, 581-584.

Kahn, G., Nowlan, S. & J. McDermott, J. (1985b). Strategies for Knowledge Acquisition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 5(9), 511-522.

Kelly, G. A. (1955). The Psychology of Personal Constructs, New York: Norton.

Klinker, G., Bentolila, J., Genetet, S., Grimes, M. & McDermott, J. (1987). KNACK—Report-Driven Knowledge Acquisition, International Journal of Man-Machine Studies, 26(1), 65-79.

Major, N. & Reichgelt, H. (1990). ALTO: An Automated Laddering Tool, in Current Trends in Knowledge Acquisition, Wielinga, B. J., J. Boose, B. Gaines, G. Schreiber, and M. van Someren (eds.), Amsterdam: IOS Press, 222-236.

Marcus, S. (1987). Taking Backtracking with a Grain of SALT, International Journal of Man-Machine Studies, 26(4), 383-398.

Michie, D. & Johnston, R. (1985). The Knowledge Machine: Artificial Intelligence and the Future of Man, New York: William Morrow and Company.

Motta, E., Rajan, T. & Eisenstadt, M. (1989). A Methodology and Tool for Knowledge Acquisition in KEATS-2, in Topics in Expert System Design, Guida, A. and T. Tasso (eds.), Amsterdam: Elsevier.

Munday, C., Cross, T., Daengdej, J. & Lukose, D. (1995). CGKEE: Conceptual Graph Knowledge Engineering Environment User and System Manual, Distributed Artificial Intelligence Centre, School of Mathematical and Information Sciences, University of New England, Armidale, NSW, Australia.

Nwana, H. S., Paton, R. C., Bench-Capon, T. J. M. & Shave, M. J. R. (1991). Facilitating the Development of Knowledge-Based Systems: A Critical Review of Acquisition Tools and Techniques, AI Communications, 4(2/3), 60-73.

Rajan, T., Motta, E. & Eisenstadt, M. (1988). Acquist: A Tool for Knowledge Acquisition, in Research and Development in Expert Systems V, Kelly, B. and A. Rector (eds.), Cambridge, UK: Cambridge University Press, 113-125.

Schreiber, A. Th., Wielinga, B. J. & Breuker, J. A. (eds.) (1993). KADS: A Principled Approach to Knowledge-Based System Development, London: Academic Press.

Scott, A. C., Clayton, J. E. & Gibson, E. L. (1991). A Practical Guide to Knowledge Acquisition, New York: Addison-Wesley.

Shaw, M. L. G. & Gaines, B. R. (1987). KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices, International Journal of Man-Machine Studies, 27(3), 251-280.

Shaw, M. L. G. & Gaines, B. R. (1993). Personal Construct Psychology Foundations for Knowledge Acquisition and Representation, in Proceedings of the 7th European Knowledge Acquisition Workshop (EKAW'93), Toulouse and Caylus, France,, September 6-10, 256-276.

Soh, L.-K., Tsatsoulis, C., Bowers, T. & Williams, A. (1998). Representing Sea Ice Knowledge in Dempster-Shafer Belief System, Proceedings of the Int. Geoscience and Remote Sensing Symposium (IGARSS'98), 2234-2236.

Soh, L.-K. & Tsatsoulis, C. (1999). Multisource Data and Knowledge Fusion for Intelligent SAR Sea Ice Classification, Proceedings of the Int. Geoscience and Remote Sensing Symposium (IGARSS'98), 68-70.

Tsatsoulis C. & Kwok, R. (Eds.). (1998). Analysis of SAR Data of the Polar Oceans, Berlin: Springer Verlag.

van Heijst, G., Schreiber, A. Th. & Wielinga, B. J. (1997). Using Explicit Ontologies for KBS Development, International Journal of Human-Computer Studies, 42(2/3), 183-292.

Vicat, C., Busac, A. & Ganascia, J.-G. (1993). CERISE: A Cyclic Approach for Knowledge Acquisition, in Proceedings of the 7th European Knowledge Acquisition Workshop (EKAW'93), Toulouse and Caylus, France, September 6-10, 237-255.