**Defence Research and Development Canada**    Recherche et développement pour la défense Canada

**DEFENCE R&D DÉFENSE**

# Development and Implementation of Classification Tools with Sidescan Sonar Image Database - Final Report

*Fiscal Reporting Period 2004/2005*

*Troy Kennedy*
*MacDonald Dettwiler and Associates Ltd.*

*Arthur Abbott*
*Centre of Geographic Sciences*

## Defence R&D Canada – Atlantic

Contract Report
DRDC Atlantic CR 2005-054
August 2005

Canada

This page intentionally left blank.

# Development and Implementation of Classification Tools with Sidescan Sonar Image Database - Final Report

*Fiscal Reporting Period 2004/2005*

Troy Kennedy
MacDonald Dettwiler and Associates Ltd.
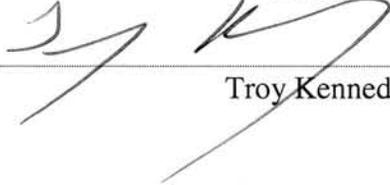
Arthur Abbott
Centre of Geographic Sciences

## Defence R&D Canada - Atlantic

Author

_Troy Kennedy_
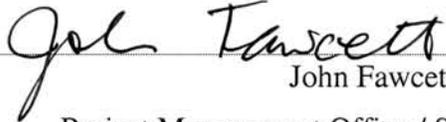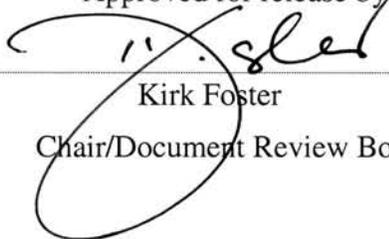
Troy Kennedy

Approved by

_John Fawcett_

John Fawcett

Project Management Office / Scientific Authority

Approved for release by

_Kirk Foster_

Kirk Foster

Chair/Document Review Board

# Abstract

This report describes the associated activities pertaining to the second year of the 2-year "Development and Implementation of Classification Tools with Sidescan Sonar Image Database" contract (contract no. W7707-032060/001/HAL). This report is to provide the reader with a summary of the progress against the requirements of contract no. W7707-032060/001/HAL as it pertains to the technical topics of discussion.

This publication will provide the reader with detailed technical progress and implementation activities associated with the satisfaction of requirements progressed during the second year of this 2 year contract. Additionally, this report will provide the reader with information gathered throughout the second year of this 2 year contracts, in addition to presenting a Requirements Traceability Matrix (RTM) of the most current status of the requirements of the contract.

# Résumé

Le présent rapport décrit les activités correspondant à la deuxième année du contrat de 2 ans (n° de contrat W7707-032060/001/HAL) relatif au développement et à la mise en oeuvre d'outils de classification avec la base de données d'images du sonar à balayage latéral. Ce rapport est un résumé des progrès effectués par rapport aux exigences du contrat n° W7707-032060/001/HAL en ce qui concerne les aspects techniques.

Cette publication présente les détails techniques sur le progrès et sur les activités de mise en oeuvre permettant de répondre aux exigences d'avancement pendant la deuxième année de ce contrat de 2 ans. De plus, ce rapport contient des informations recueillies pendant la deuxième année de ce contrat et présente aussi une matrice de traçabilité des exigences (RTM) de l'état le plus courant des exigences du contrat.

This page intentionally left blank.

# Executive summary

## INTRODUCTION

Over the last several years, DRDC Atlantic has been involved with the development of new minehunting systems for the Canadian Navy, the Remote Minehunting System (RMS) being an important example. In addition, DRDC Atlantic has carried out research with respect to the processing of sidescan sonar data and the development of new algorithms for the computer-aided detection and classification of minelike objects on the seabed. The DRDC Atlantic Sonar Image Processing Systems (SIPS) has been a testbed for this software development. Over the last 2 years, the contract described in this report has been concerned with the development of computer detection, segmentation, and classification algorithms (and the associated computer interfaces) for use with real sonar images. These algorithms are important in helping a human operator deal with the very large amounts of sonar data which result from a minehunting survey. Much of the development work was done with the DRDC Atlantic SIPS. However, in addition, the computer-aided detection (CAD) software from the Remote Minehunting System (RMS) Technology Demonstration Project was integrated with the Route Survey Data Analysis Facility (RSDAF) Sonar Analysis Console (SAC) software and database. Also, some of the new classification tools developed with this contract were integrated with this new RMS sonar analysis software.

## RESULTS

The DRDC Atlantic SIPS is now integrated with a number of very useful image database and image classification tools. A number of GUI's allow for the user definition of mine and non-mine classes and the simple application of classification algorithm training and testing. As well, there are tools for populating image databases and the automated generation of feature files. Work on the addition of computer-classification tools to the RMS sonar analysis software has begun. Preliminary results have shown that the computer classification of sidescan sonar images appears promising. The development of robust computer classification algorithms will be able to help human operators in their analysis of large amounts of data. In the future, there may also be a role for such algorithms with autonomous sonar platforms.

## FUTURE WORK

There remains work to do with respect to the porting of the SIPS classification functionality to the RSDAF/RMS SAC software. It is also anticipated that the feature extraction and classification algorithms will continue to evolve with the future use of the classification software with trials data. There remains interesting questions on how to optimally combine computer and human detection and classification capabilities.

Kennedy. T, Abbott. A. 2005. Development and Implementation of Classification Tools with Sidescan Sonar Image Database Final Report. DRDC Atlantic CR 2005-054, MacDonald Dettwiler and Associates Ltd.

# Sommaire

## INTRODUCTION

Ces dernières années, le centre RDDC Atlantique a été actif dans le développement de nouveaux systèmes de chasse aux mines pour la marine canadienne, dont l'important système télécommandé de chasse aux mines (RMS). De plus, RDDC Atlantique a mené des recherches sur le traitement des données du sonar à balayage latéral et sur le développement de nouveaux algorithmes pour la détection et la classification assistées par ordinateur d'objets ressemblant à des mines sur le fond marin. Le système de traitement d'images sonar (STIS) de RDDC Atlantique a servi de banc d'essai dans le développement du logiciel. Durant les deux dernières années, le travail en vertu du contrat faisant l'objet du présent rapport visait le développement d'algorithmes de détection, de segmentation et de classification par ordinateur (et des interfaces d'ordinateur connexes) destinés être utilisés avec de vraies images sonar. Ces algorithmes sont importants, car ils aident l'opérateur humain à traiter la très grande quantité de données sonar obtenues d'une campagne de chasse aux mines. Les travaux de développement ont été faits en grande partie avec le STIS de RDDC Atlantique. De plus, le logiciel de détection assistée par ordinateur (CAD) du système télécommandé de chasse aux mines (RMS) faisant partie d'un projet de démonstration de technologies a été intégré au logiciel et à la base de données de la console d'analyse des données sonar (SAC) du centre d'analyse des données de levés des fonds marins (RSDAF). Certains des nouveaux outils de classification développés dans le cadre de ce contrat ont aussi été intégrés à ce nouveau logiciel d'analyse de données sonar du RMS.

## RÉSULTATS

Le STIS de RDDC Atlantique est maintenant intégré à plusieurs outils de base de données d'images et de classification d'images très utiles. Plusieurs interfaces graphiques permettent à l'utilisateur de définir les images comme des mines ou autres que des mines de procéder à la simple application de l'apprentissage de l'essai des algorithmes de classification. Il existe également des outils pour alimenter les bases de données d'images et pour la génération automatique de fichiers de caractéristiques. Les travaux sur l'ajout d'outils de classification par ordinateur au logiciel d'analyse de données sonar du RMS ont commencé. D'après les résultats préliminaires, la classification par ordinateur d'images de sonar à balayage latéral semble prometteuse. Le développement d'algorithmes robustes de classification par ordinateur aidera l'opérateur humain dans l'analyse de la grande quantité de données obtenues. À l'avenir, ces algorithmes joueront peut-être un rôle dans les plates-formes sonar autonomes.

## RECHERCHES FUTURES

Les travaux ne sont pas terminés en ce qui a trait au transfert des fonctions de classification du STIS au logiciel de la SAC du RSDAF/RMS. Nous prévoyons également que les algorithmes d'extraction et de classification des caractéristiques continueront d'évoluer avec l'utilisation

future du logiciel de classification sur les données d'essais. Certaines questions intéressantes demeurent relativement à la façon de combiner efficacement les fonctions de détection et de classification par ordinateur aux capacités de détection et de classification de l'humain

Kennedy. T, Abbott. A. 2005. Development and Implementation of Classification Tools with Sidescan Sonar Image Database Final Report. DRDC Atlantic CR 2005-054, MacDonald Dettwiler and Associates Ltd.

# Table of contents

# List of figures

# List of tables

This page intentionally left blank.

# Introduction

Over the last few years, the Department of National Defence (DND)/ DRDC Atlantic has developed software for the importing, displaying, and processing of sidescan sonar data. Recently, a simple database structure for the saving of target imagery and associated ancillary data has been developed. An associated Database Manager (DM) allows for the subsequent display of these targets and provides some image processing tools. The requirement for fitting SIPS and the DM with classification tools to aid the operator in their classification of images is presently being accomplished through the execution of the Development and Implementation of Classification Tools with Sidescan Sonar Image Database contract. Display and database software, developed under RMS TD, is also being modified for integration into the DM.

At the commencement of year two of this contract, the list of prioritized requirements, in the form of a RTM, was reviewed and it was mutually concurred by all members of the IPT that the existing listing of prioritized requirements remained valid. The second section (Table 1) of this publication presents the prioritized RTM.

During this reporting period, the majority of the effort expended against the technical progression of the project was spent on the primary component of the project, involving the implementation of contractual requirements into the Sonar Information Processing System (SIPS) software product. This implementation entailed the inclusion of feature computation algorithms, an interface with databases for the training and testing of computer classifiers, and the implementation of the specified classification algorithms. The classifiers were also linked with the playback window of SIPS, allowing for the "real-time" classification of images.

At the commencement of year two of this contract, the list of prioritized requirements, in the form of a RTM, was reviewed. It was mutually concurred by all members of the Integrated Project Team (IPT) that the existing listing of prioritized requirements remained valid. Section two (Table 1) of this publication presents the prioritized RTM.

The beginning of year two also saw the completion of software amendment 001 to the original contract. This software amendment was for the implementation of the Computer-Aided Detection/Computer-Aided Classification (CAD/CAC) software package into the MacDonald Dettwiler and Associates Ltd. (MDA)-developed Sonar Analysis Console (SAC) software, in use by the Route Survey offices of both coasts.

Effort was also expended upfront in year two in order to transfer the MATLAB licenses procured by MDA, on behalf of the Crown under the Remote Minehunting System (RMS) Technology Demonstrator (TD) contract, over to official Defence R&D Canada (DRDC) – Atlantic ownership.

The next activity planned for project execution was the implementation of three classification algorithms for use with SIPS task. The three classification algorithms were as follows:

(1) A tree-classifier algorithm (DRDC Atlantic supplied)

(2) A N-nearest neighbour algorithm

(3) A support vector machine (SVM)

As mutually agreed by the IPT at the Kick-off Meeting, the development of the SVM portion of the Implement Three Classification Algorithms, for use with SIPS requirement, was assigned a low priority for implementation within the requirements analysis session.

Planning for the Implementation of Three Classification Algorithms for use with the SIPS task commenced in December 2003, with discussions held at the December Project Review Meeting. A draft release of the workflow plan was issued to the IPT in January 2004 and finalized in February 2004. Design and development on this task commenced as a parallel activity to the issue resolution phase of Feature Extraction.

The 15 April 2004 software product release not only includes issue resolution patches for the Feature Extraction software but also encompasses the classification algorithm source code.

Other areas of progression during the course of the second year of this contract are as follows:

- Additional improvements to the of Feature Extraction aspect of the SIPS software product were rendered

- Development of a GUI Interface into the SIPS software product occurred

- A work plan for mid-term software maintenance was approved and implemented

- Implementation of template derived features were added into the SIPS software product

- The Centre of Geographical Sciences (COGS) completed an upgrade to Interactive Data Language (IDL) version 6.1

- Product build releases and general upkeep/maintenance to the configuration management (CM) server (Fireball) continued throughout year two of the contract

- Software maintenance to the Remote Minehunting Tactical Center (RMTC), located at the DRDC Atlantic lab, also occurred throughout the second year of the contract

Lastly, the implementation of the Work Plan for Feature Extraction into the CAD/CAC software occurred throughout the fall of 2004 with rollout occurring in January 2005.

## Purpose

The purpose of this Development and Implementation of Classification Tools with Sidescan Sonar Image Database – Final Report is to provide the reader with a summary of the progress to date on contract no. W7707-032060/001/HAL, as it pertains to both programmatic and technical topics of discussion.

This publication will provide the reader with: a) a detailed technical information gathered throughout the second year of this 2 year contract, b) requirements traceability information relevant to the contract, and c) a conclusion section complete with recommendations for follow-on support.

This document also provides an Executive Summary and Abstract section, which precedes this section of the report.

## Scope

The scope of this final report is captured under the following section:

Introduction (includes the purpose and scope of the publication)

Requirements Traceability Matrix

SIPS and CAD/CAC Technical Progress – Fiscal Year 2004/2005

Report Conclusion and General Recommendations for Follow-on

References

Acronyms

Distribution Listing

# Requirements Traceability Matrix

Table 1 outlines the final status of requirements, post year 2, of the 2-year contract for the "Development and Implementation of Classification Tools with Sidescan Sonar Image Database" (contract no. W7707-032060/001/HAL). Table 1 presents the list of contractual requirements in their entirety and provides status of progress towards the completion of the individual requirements.

*Table 1: Summary of Contract Work Completed*

| SOR Ref. # | Tasking Title | Description of Tasking | Priority | Status |
|---|---|---|---|---|
| 9. i. | Feature Extraction | For a given target image a set of features will be computed. Many of these features will be based upon the segmentation of the image into shadow/highlight/background regions. DRDC Atlantic will provide preliminary code for image segmentation and the computation of quantities, such as elliptical parameters of the highlight and shadow, the estimated length and height of the object, statistical moments for the highlight and shadow regions, etc. The contractor will ensure that these algorithms compute the features for subsequent classification work. Some of these algorithms have already been included in SIPS and the RMS software; in these cases the contractor should verify their accuracy and robustness. | 1 | Implemented into both SIPS and CAC/CAC software products |
| 9. vi. | Implement Three Classification Algorithms | The contractor shall implement three classification algorithms for use with SIPS: a tree classifier algorithm (DRDC-supplied), a N-nearest neighbour algorithm and a SVM. | 2 | Complete |
| N/A | Implementation of the CAD/CAC software in the upgraded RSDAF SAC software | Effort breakdown of tasks for implementation:<br><br>1) Database: implement schema changes and data layer updates<br><br>2) CAD/CAC: interface verification<br><br>3) SAC: region drawing, region management and CAD/CAC target updates<br><br>4) Overall functional testing<br><br>5) Project engineering of the new requirement | New Requirement | Amendment No. 001 was implemented into the CAC/CAC software product |

| SOR Ref. # | Tasking Title | Description of Tasking | Priority | Status |
|---|---|---|---|---|
| N/A | Spring 2004 Pre-Trials and Trials Support Requirement | **Part A**<br><br>Prior to the scientific trial of May 3 – 14, 2004, the contractor is required to work with DRDC to ensure that the Sonar Analysis Software Suite, including hardware and software, is set up and operational in the RMTC.<br><br>**Part B**<br><br>During the scientific trials, the contractor will be responsible for the operation of the Sonar Analysis Software Suite to support the operation of the RMS. The trial will last for a period of 18 working days at CFB Esquimalt. During this time, both the Klein 5500 side scan sonar and the Reason 8125 bathymetric multi-beam sonar will be operated in the shallow waters off Esquimalt and Victoria Harbours. | New Requirement | Amendment No. 002 was completed by MDA during the required period(s) |
| 9. iii. | Develop GUI | Graphical User Interface (GUI) shall be developed to allow for the easy training and testing of classification algorithms on a specified database. In the case of SIPS, the database is list of files, ancillary data, manual classification and pointers to files, which contain small swaths of sidescan sonar data. (The RMS database structure is much different as it actually contains the entire sidescan data for a survey. This would need to be adapted to allow for the quick accessing of "snippets" of data corresponding to a list of targets.) | 3 | Implemented into the SIPS software product in year 1 of the contract |
| 9. x. | Implementation of an Operator Display | A display shall be implemented, which will allow the operator to view with a display of the object, the results of the Feature Extraction (e.g. shadow/highlight segmentation, determined ellipses, shadow perimeter, etc.). For a classifier, (if possible) the operator will have the ability to query the system as to why the classifier made its decision. For example, in the case of a tree classifier the values of the various features that were used to make the classification could be listed. For a nearest-neighbour approach the "closest" images that were used could be displayed, etc. | 4 | Implemented into the SIPS software product in year 2 of the contract |

| SOR Ref. # | Tasking Title | Description of Tasking | Priority | Status |
|---|---|---|---|---|
| 9. iv. | Implementation of Training and Testing Options | A variety of different training and testing options will be allowed. For a specified database, a user can specify how to group individual classes into the classes for classification. For example, all the individual cylinder classes could be associated as a single class, or all Mine-Like Objects (MLO) might be grouped together as a class. | 5 | Implemented into the SIPS software product in year 1 of the contract |
| 9. v. | Implementation of Training and Testing Options | The user could then specify on a file-by-file basis or as a percentage, the files that should be used training - a random (but specified percentage) selection would be one option. | 6 | Implemented into the SIPS software product in year 1 of the contract |
| 9. vii. | Classifier and ASCII Header Implementation | The output from the training with an algorithm will be a file, which can be saved which will completely specify the resulting classifier. This file can then be used with any database for classification. | 7 | Implemented into the SIPS software product in year 2 of the contract |
| 9. viii. | Implementation of a GUI for Testing | There shall be a GUI for testing, which will: (a) work the portion of the database used for training (but the files not used for training), or (b) another database. The output will be the statistics of the classifier performance (correctly classified, incorrectly classified and a confusion matrix). | 8 | Implemented into the SIPS software product in year 1 of the contract |
| 9. ix. | Implementation of a Simple Detection Scheme | A simple detection scheme (e.g. output of a convolution scheme) shall be implemented with the playback mode in SIPS and a specified saved classification filter can be specified to automatically perform the feature extraction of the associated small image. A specified classifier can also be used in association with the "zoom" window in the SIPS playback. | 9 | Requirement 9. ix. is satisfied in the existing CAD/CAC detection code (Contract Amendment 001).<br><br>Implemented into the SIPS software product in year 2 of the contract |
| 9. xi. | Develop Multi-aspect Classification Tools | These would be classifiers, which would use the features determined for the same target taken as two different aspects in the same survey. The classification algorithms to be considered would consist of simply fusing the individual classification results and also considering combined feature sets. | *Previously* 10 | Deferred from this contract in favour of other functionality |

| SOR Ref. # | Tasking Title | Description of Tasking | Priority | Status |
|---|---|---|---|---|
| 9. xii. | Accommodations for Easy Addition of User-supplied Algorithms | A methodology allowing for the easy addition of user-supplied algorithms for detection (in playback, real-time and post analysis modes) and for classification (in playback, real-time or post analysis modes) shall be developed. | *Previously* 11 | Deferred from this contract in favour of other functionality |
| 9. ii. | Ray Tracing | The contractor shall develop a simple and fast ray tracing code for specified solid models of MLOs and generating synthetic sidescan sonar images, e.g. cylinders, truncated cones, spheres, and will add a window to the existing DM to allow the operator to compare the predicted image to the real image. A simple interface would also allow the operator to interactively tilt and bury the object. | *Previously* 12 | Deferred from this contract in favour of other functionality |
| N/A | Implementation of Template Features | New feature extraction tools based on a series of templates, which represent different types of targets, will be implemented in SIPS. The implemented classification algorithms will be modified to utilize the new features. | New Requirement | Implemented into the SIPS software product in year 2 of the contract |
| 9. i. | Implementation of SIPS Feature Extraction Routines in the CAD/CAC System | The basic SIPS feature extraction tools (e.g. object height, length, lacunarity, etc.) will be implemented in the CAD/CAC system. These tools will be ported into a form that is usable from within the CAD/CAC framework. | See Priority 1 | Implemented into the CAD/CAC software product in year 2 of the contract |

# SIPS and CAD/CAC Technical Progress – Fiscal Year 2004/2005

## Implement Three Classification Algorithms (9. vi.)

As an update to the information regarding this contract line item in the SIPS Interim Report, a SVM algorithm has been added to SIPS. This brings the total number of classification algorithms available to SIPS to four.

### Implement Three Classification Algorithms - Planning

It was decided to use a Kernel Ridge Regression approach to create a SVM. The initial kernel selection was the Radial Basis Function [R-4].

Since a SVM is a binary classifier, the main result is to classify a target as either mine-like or non-mine-like. Mine-like targets are then further classified to an individual classification criteria virtual class by the use of the virtual class means of the mine-like virtual classes. This portion of the classification uses a nearest means approach.

### Implement Three Classification Algorithms - Implementation

The SVM training process uses, as its training set, the contents of the feature files designated by a loaded classification criteria file. Any feature listed in the criteria file with a weight of 0.0 is removed from the training set. The training set is stored internally as a 2-dimensional array of doubles. The last column of each row contains the training target classification.

To train a SVM, a kernel matrix is filled using the kernel function. The current function is the Radial Basis function, shown as follows:

$$K(x, y) = \exp\left( -\frac{\|x - y\|^2}{\sigma^2} \right)$$

The value of $\sigma$ is controllable by the user. The default value is 0.5. Hooks have been placed into the code to allow for the addition of different kernel functions.

Once the matrix has been filled, the series of linear equations may be solved. This is accomplished using the built-in IDL function *LA_LINEAR_EQUATION*. This function is based on Lu decomposition to solve the system of linear equations, $AX = B$. The returned result is a vector of the solutions for the equations set.

A summary of the SVM algorithm is as follows:

- Remove the zero weighted features from the training set

- Normalize the training set

- Fill the kernel matrix

- Solve the set of linear equations

- Return the vector of solutions to the set of equations

## Implement Three Classification Algorithms – Issues

Table 2 presents the issues of significant, and notable, relevance to the implementation of the Implement Three Classification Algorithms.

| Issue | Resolution |
|---|---|
| The results that were returned from the initial SVM tests were considered "too good to be true". | The training target class was removed from consideration when creating a summation of the discriminate. |
| Initially Cramer's rule was used for solving the set of equations. This proved to be slow. | LA_LINEAR_EQUATION was substituted for Cramer |

*Table 2: SVM Issues*

## Implement Three Classification Algorithms - Rollout

The SVM classifier was initially rolled out in late November 2004. Final rollout occurred in early December 2004.

## Implement Three Classification Algorithms - Recommendations for Follow-on

The following are recommendations as they pertain to the *Implement Three Classification Algorithm* requirement:

- Implementation of additional kernel functions as alternatives to the Radial Basis Kernel

- Implementation of alternative SVM algorithms; one such algorithm could be the Sequential Minimal Optimization algorithm

- Consideration should be given to the implementation of an ID3 tree as an alternative classifier to the CART tree

- Consideration should also be given to the investigations into a fuzzy means classification function

# Classifier and ASCII Header Implementation (9. vii.)

The output from the training with an algorithm is a file that can be saved, which completely specifies the resulting classifier and an associated American Standard Code for Information Interchange (ASCII) header. The ASCII header contains all the details about the classifier. This file can then be used for classification with any database. Currently, only the CART tree and the SVM create classifier files. Classifier files are not required for nearest neighbour and nearest means.

### Classifier and ASCII Header Implementation - Planning

Planning for this item involved, for each classifier, determining what information would be needed to use the classifier to deal with a future unclassified target.

### Classifier and ASCII Header Implementation - Implementation

The following subsections define the Classifier and ASCII Header implementation activities performed during the second year of this contract.

#### *CART Tree*

The following items are written to the CART tree classifier file:

- Number of features used

- Number of virtual classes

- List of feature weights

- Contents of each node of the CART tree

The contents of a tree node are as follows:

- Node number

- Split feature

- Split level

- Terminal flag

- Number of nodes in this node's subtree

- Misclassification rate of the node's subtree

- Node's misclassification rate

- Node's small G pruning value

- Node's large G pruning value

- Node number of the left child

- Node number of the right child

- Number of training targets attached to the node

- Training set attached to the node

### *SVM Classifier*

The following items are written to the SVM classifier file:

- Number of training targets used

- Solution to the set of linear equations. (The solution vector is written out as one element per file line.)

- Feature weights

- Support vectors (training set)

- Kernel flag

- Any kernel related variables (each variable would be written to the file, one per line)

Currently, there is only one kernel function available, the Radial Basis Function (RBF) kernel. This kernel only requires one variable, gamma, which is written to the file. Hooks have been placed into the code that allows for the addition of new kernel functions.

## Classifier and ASCII Header Implementation – Issues

Table 3 presents the issues of significant, and notable, relevance to the implementation of the Classifier and ASCII Header Implementation.

| Issue | Resolution |
|---|---|
| The loaded CART tree could not deal with test sets that contained a different number of virtual classes as the training set that was used to build the tree. | The number of classes used to build the tree was added to the file. |

*Table 3: Classifier File Issues*

## Classifier and ASCII Header Implementation - Rollout

The rollout for the CART tree classifier took place in late February 2004. The CART tree classifier was re-rolled out in April 2004 after the implementation of classification functionality within the zoom window. The rollout for the SVM classifier occurred in December 2004.

### Classifier and ASCII Header Implementation - Recommendations for Follow-on

The following is our recommendations pertaining the *Classifier and ASCII Header Implementation* requirement:

- The establishment of a naming convention for classifier files

# Implementation of a Simple Detection Scheme (9. ix.)

The implementation of a simple detection scheme within the SIPS software product involved scanning the image area currently displayed in the replay window for possible Regions of Interest (ROI). Detected ROIs would then be viewed in the zoom window. ROIs may be found by one of two methods: a lacunarity scan, and/or a matched filter scan.

As a part of this requirement, the currently implemented classification algorithms were added to the zoom window.

### Implementation of a Simple Detection Scheme - Planning

As part of the review process associated with this Work Plan, it was decided that this item would be broken down into the following items:

- Implementation of classification tools in the zoom window

- Implementation of a lacunarity scan

- Matched filter scan

- Inspection of regions of interest

In addition to the area currently displayed in the replay window, each scan would also look 2 meters forwards and backwards, in the file set. This would allow for the detection of ROIs at the edges of the displayed area.

### Implementation of a Simple Detection Scheme - Implementation

The following subsections define the Implementation of a Simple Detection Scheme related implementation activities performed during the second year of this contract.

#### *Zoom Window Classification*

A simpler version of the algorithm training GUI is used to classify targets in the zoom window. Instead of displaying the classification result in a separate sub-interface, as in the algorithm training interface, the results are displayed in the main interface's log window. Any of the currently implemented classification algorithms may be used to classify a target with this interface. The final classification is returned and stored in the zoom window. The returned classification is then used as the default classification for any database swaths created using this instance of the zoom window.

### *Lacunarity Scan*

The lacunarity scan was accomplished using a sliding window algorithm. The window size is based on real world measurements. To determine the pixel size of the scan window, the pixel resolutions, in both the along track and across track, are applied. Before the algorithm commences, the average lacunarity for the entire image is calculated.

Starting at the image origin, a 3 meter by 3 meter window of data is fetched. The lacunarity for this window is calculated. If the difference of this lacunarity and the average lacunarity is greater than the user defined threshold, than a 2 meter by 2 meter sized window is fetched. The same lacunarity calculations are repeated. If the difference is again greater than the threshold, then the box is again reduced in size to a meter by 1 meter area and then scanned. The smallest window size that passed the test is returned as a detection. Once a detection has been determined, or not, the base 3 meter by 3 meter window is moved to a new location within the image. The new location overlaps the old location by 1.75 meter in the along track and across track dimensions. When the window reaches the water column interface in the across track dimension, the window is skipped to the starboard side of the towfish and scanning begins again. The algorithm ends when the entire image has been scanned. Any detections are passed back as an array of *scan_result* structures. Figures 1 and 2 summarize the sliding window scan.

**Sliding Window Lacunarity Scan**
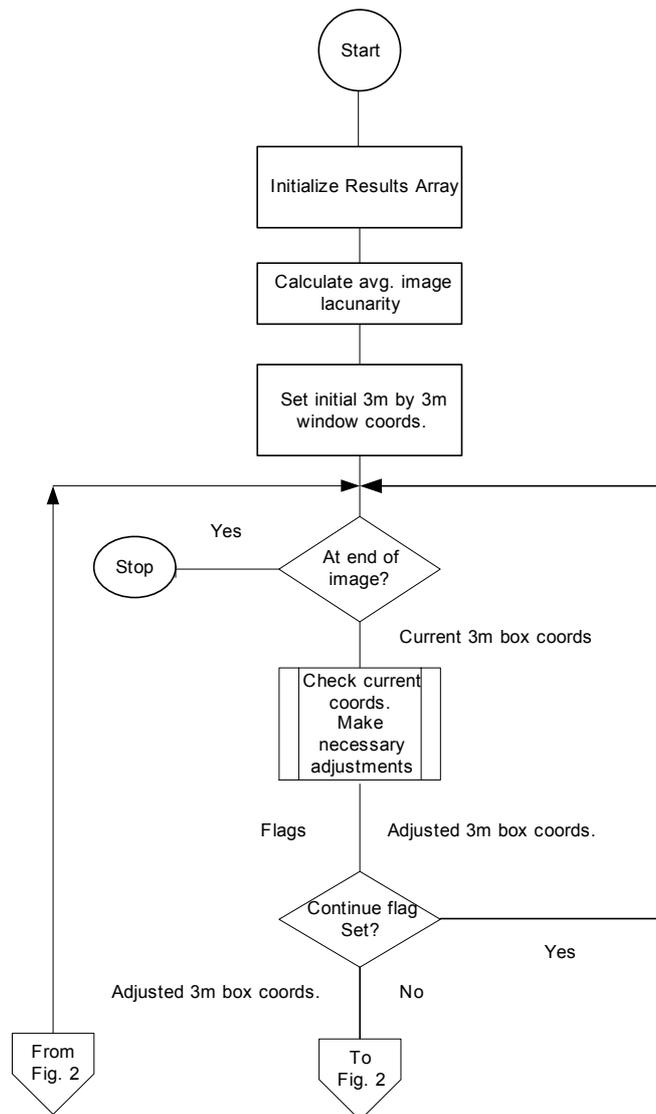


**Figure 1.** *First Part of the Sliding Window Lacunarity Scan*

**Sliding Window Lacunarity Scan**

```
    ┌─────────┐                    ┌─────────┐
    │   To    │                    │  From   │
    │ Fig. 1  │                    │ Fig. 1  │
    └─┐     ┌─┘                    └─┐     ┌─┘
      └─┐ ┌─┘                        └─┐ ┌─┘
        ▲                              │
        │                   ┌──────────────────┐
        │                   │   Get 3m by 3m   │
        │                   │ data window using│
        │                   │  current coords  │
        │                   └──────────────────┘
        │                              │     3m by 3m data window
        │                   ┌──────────────────┐
        │                   │  Do lacunarity   │
        │                   │ scan(s).  Add    │
        │                   │ any results to   │
        │                   │      array       │
        │                   └──────────────────┘
        │                              │
        │                   ┌──────────────────┐
        │                   │  Shift 3m window │
        │     New 3m box    │ coords towards   │
        │       coords      │   image center   │
        │                   └──────────────────┘
        └──────────────────────────────┘
```
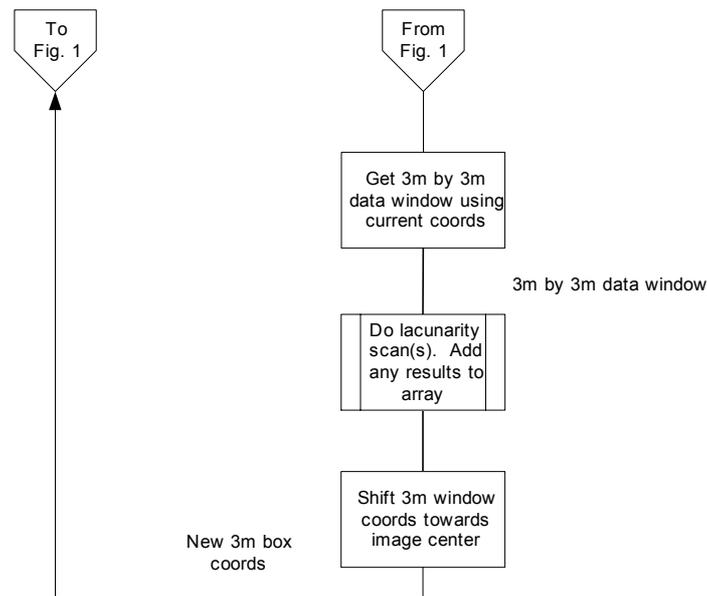
*Figure 2.* Second Part of Sliding Window Lacunarity Scan

### Matched Filter Scan

To determine if there is an area of interest, the matched filter scans an area of an image, using *convolution*. The filter that is used is described in the next paragraph.

The basic zero degree filter is a float array of 14 rows and 34 columns. Column 0 represents highlight, with a value of 0.5. Columns 1 through 3 represent high highlight, with a value of 1.0. Column 4 represents more highlight. Columns 5 and 6 are background, with a value of 0.0. The remaining columns represent shadow with a value of -1.0. Filters of different aspect are achieved by using built-in IDL routines to warp the basic filter. Figure 3 shows a 0 degree aspect filter.

**Figure 3.** *Zero Degree Filter on the Port Side*

The match filter was originally implemented using the sliding window scan algorithm. Each window was scanned using a series of filters depicting a cylinder target at different aspect angles. This was later superseded by a faster algorithm, which scans an entire side of the image at once.

The newer matched filter implementation breaks the image into two separate image arrays, representing each side of the towfish. Each image is then convolved with the filter. The image values of the convolved image are first divided by the number of non-zero elements of the filter. Any pixel values above a user specified level are considered detections. The algorithm is as follows:

- Re-bin image to five discrete levels:
    - Deep shadow (0% – 3.5% of maximum intensity value)
    - Shadow (5% - 25% of maximum intensity value)
    - Background (25% - 80% of maximum intensity value)
    - Highlight (80% - 99% of maximum intensity value)
    - High highlight (99% - 100% of maximum intensity value)
- Split image into port and starboard sides
- For each side of the image:
    - Remove the water column pixels
    - Convolve image with filter
    - Divide the resulting image values by the number of non-zero pixels of the filter
    - Find all pixels at or above the user supplied threshold
    - Label pixel groups

- o Remove pixel groups that are larger than 3 meters or are at the edges of the image
- o Return results

The convolution is accomplished using IDL's *convol* command with the image and filter as arguments. See next section for a description of the results. Figure 4 shows Replay showing three cylinders found by the matched filter detector.
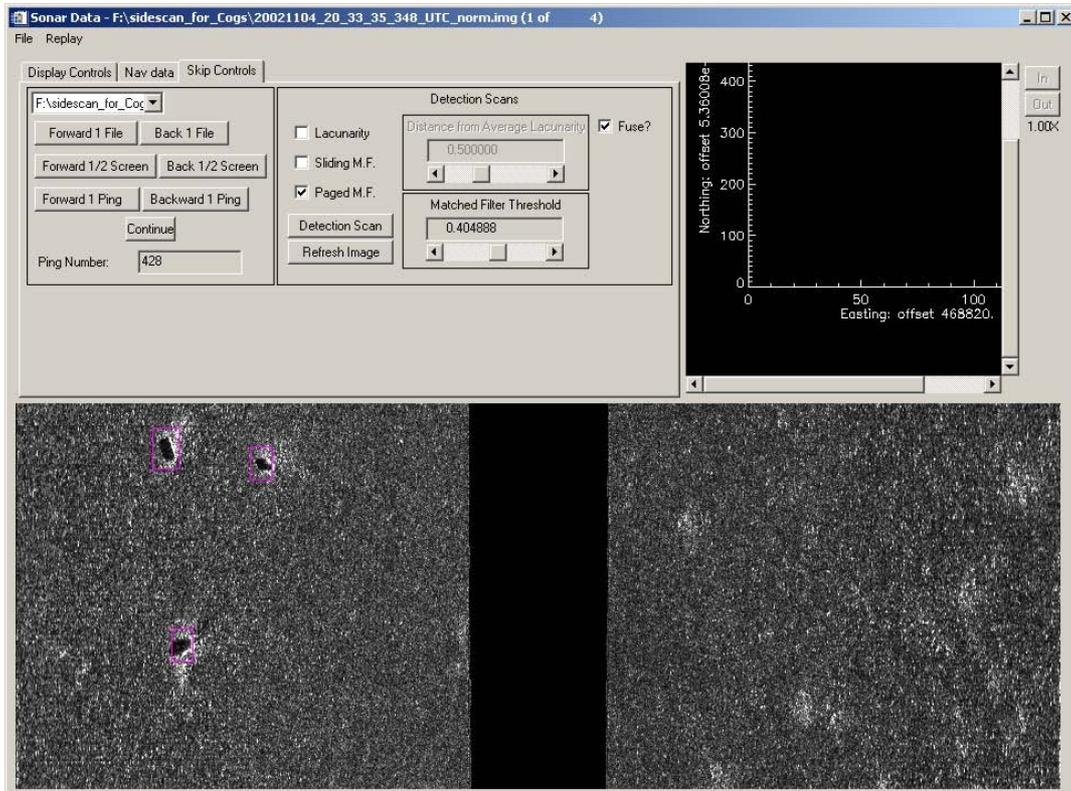


**Figure 4.** *Replay Showing Three Cylinders Found by the Matched Filter Detector*

The matched filter algorithm was based on the paper referenced at [R-6].

### Inspection of Regions of Interest

All distinct detections are returned to the user as an array of *scan_result* structures. The *scan_result* structure is defined as follows:

- x0/x1 : across track bounds of the detection

- y0/y1 : along track bounds of the detection

- scanResult : peak value found

- scanType : flag indicating algorithm

- filternum : which filter used

The x and y coordinates detail the bounding box of the detection. The scanResult value is the correlation value, and the type is a flag indicating which algorithm returned the result. The results are displayed in a GUI, where the user may view one or more of the detections in zoom windows. The detection results are also displayed in the replay window, as well. Each of the detections is displayed as a box, over-plotted on the sonar display. Results from the lacunarity scan are drawn in red; matched filter results are drawn in purple. Any result that was detected by multiple detectors is drawn in yellow.

Figure 5 depicts a diagram showing the flow of information for detection, to an instance of the zoom window, which will display an individual detection result.
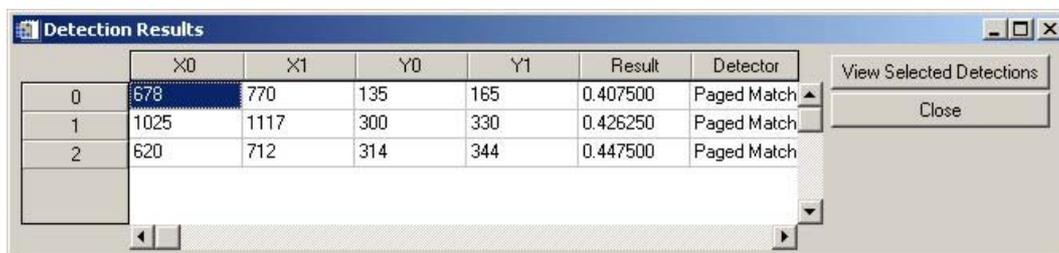
| | X0 | X1 | Y0 | Y1 | Result | Detector | |
|---|------|------|-----|-----|----------|-------------|---|
| 0 | 678  | 770  | 135 | 165 | 0.407500 | Paged Match | |
| 1 | 1025 | 1117 | 300 | 330 | 0.426250 | Paged Match | |
| 2 | 620  | 712  | 314 | 344 | 0.447500 | Paged Match | |

**Figure 5.** *Detection Results*

Figure 6 presents the data flow from Detection results interface to zoom window.

Data Flow from Detection Results Interface to
Zoom Window

```
        ┌──────────────┐
        │  Detection   │
        │   Results    │
        │  Interface   │
        └──────┬───────┘
               │
            Detection
            Coordinates
               │
               ▼
        ╭──────────────╮
        │    Create    │
        │  file_info   │
        │ Structure(s) │
        ╰──────┬───────╯
               │
            File_info
            Structure(s)
               │
               ▼
        ╭──────────────╮          ┌──────────┐
        │  Get Zoom    │◄─Zoom Data─│ fileSet  │
        │    Data      │          └──────────┘
        ╰──────┬───────╯
               │
           Zoom data,
            file_info
           Structure(s)
               │
               ▼
        ┌──────────────┐
        │     Zoom     │
        │    Window    │
        └──────────────┘
```
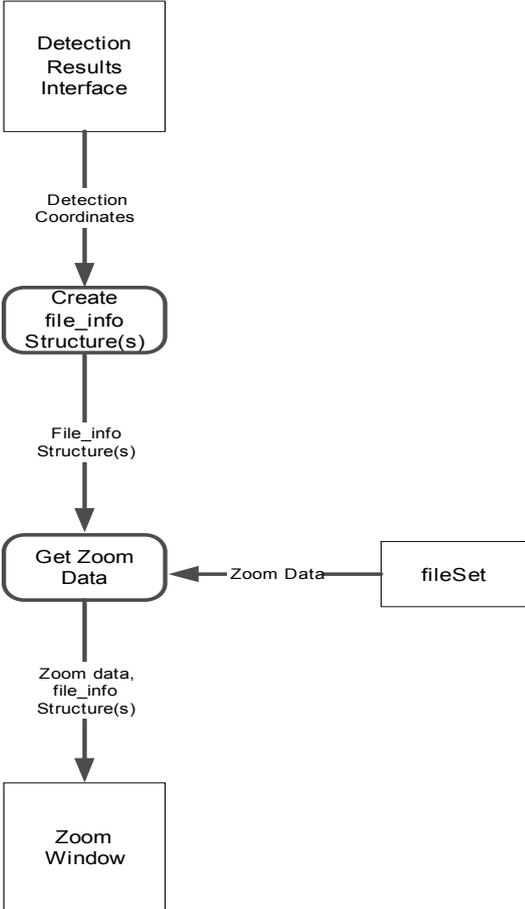
*Figure 6.* *Data Flow from Detection Results Interface to Zoom Window*

# Implementation of a Simple Detection Scheme – Issues

Table 4 presents the issues of significant, and notable, relevance to the implementation of the Implementation of a Simple Detection Scheme.

| Issue | Resolution |
|---|---|
| Scans returned a lot of detections very close to the water column interface. | Results that are within a meter of the water column interface are ignored. |
| The sliding window version of the matched filter took too much time to complete. | The sliding window version was replaced by the paged version. |
| The results for the matched filter seemed to be high. | Testing of an artificial object in an image array was carried out. Post convolution normalization was modified. |
| Some images produced several detections from a single object. | The user is given the option of amalgamating detections that are adjacent to each other or overlap. |

*Table 4: Implementation of a Simple Detection Scheme Issues*

# Implementation of a Simple Detection Scheme - Rollout

Roll out for the lacunarity scan occurred in May 2004. Initial rollout for the sliding window version of the matched filter occurred in June 2004. The paged version rolled out in early July. The final rollout occurred in late July after a round of testing using synthetic data, as recommended by MDA.

## Implementation of a Simple Detection Scheme - Recommendations for Follow-on

The following are recommendations as they pertain to the implementation of a simple detection scheme requirement:

- The addition of "finer" user controls for the matched filter algorithm (The user may wish to manipulate the pre-filter threshold values.)

- The implementation of additional detection algorithms

# Implementation of an Operator Display (9. x.)

As an update to the information relating to this line item, presented in the SIPS interim report, an operator display has been implemented to display the results returned from a classification algorithms.

## Implementation of an Operator Display - Planning

For the CART tree a simpler version of a tree node structure had to be designed to store the information in relation to the nodes visited.

## Implementation of an Operator Display – Implementation

The following subsections describe the effort expended, during year-two of the contract, in the implementation of the Operator Display requirement.

### *Nearest Neighbour*

The following information is displayed to the user for the nearest neighbour algorithm:

- Swath name

- Which type of classification used (standard NN or clutter based)

- The n-nearest neighbours and each distance

- Final virtual class determination

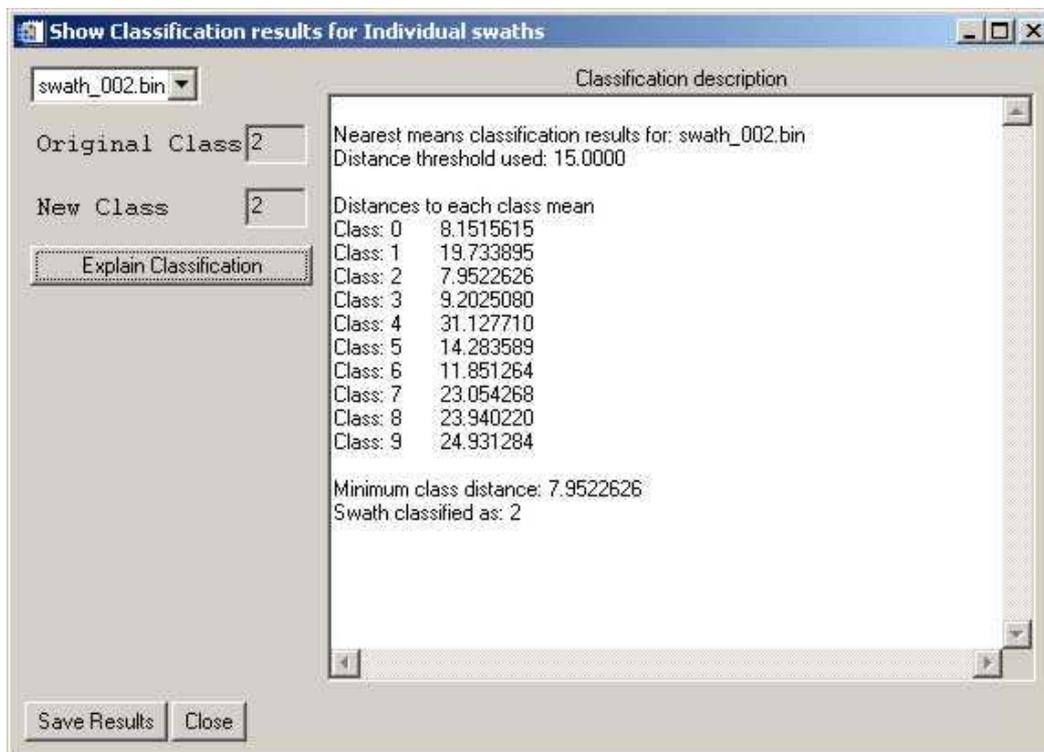Figure 7 depicts a window displaying nearest neighbour classification results.
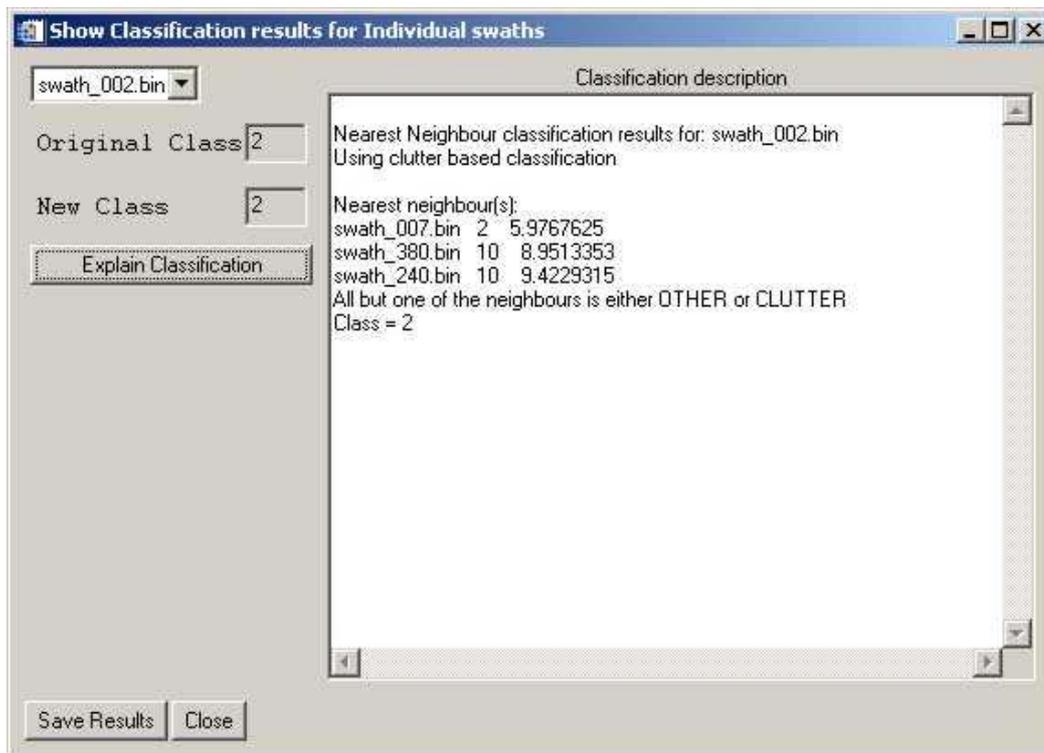
**Figure 7.** *Nearest Neighbour Classification Results*

### Nearest Means

The following information is displayed to the user for the nearest means algorithm:

- Swath name

- Clutter class distance threshold

- Calculated distance, in feature space, to each user defined virtual class

- Minimum class distance found

- Final virtual class determination

Figure 8 depicts a window displaying the nearest means classification results.



**Figure 8.** *Nearest Means Classification Results*

## CART Decision Tree

The following information is displayed to the user for the CART tree:

- Swath name
- List of tree nodes visited
  - Node split feature
  - Feature split value and how it compares to the same target feature
- The terminal node's class populations
- Final virtual classification

To display the previously mentioned information, the nodes visited are stored in an array of *lite_node* structures. The layout of the structure is as follows:

- Number

- Feature

- Level

- Pointer to node class populations (only used when terminal node is reached)

Figure 9 depicts a window displaying the CART tree classification results.
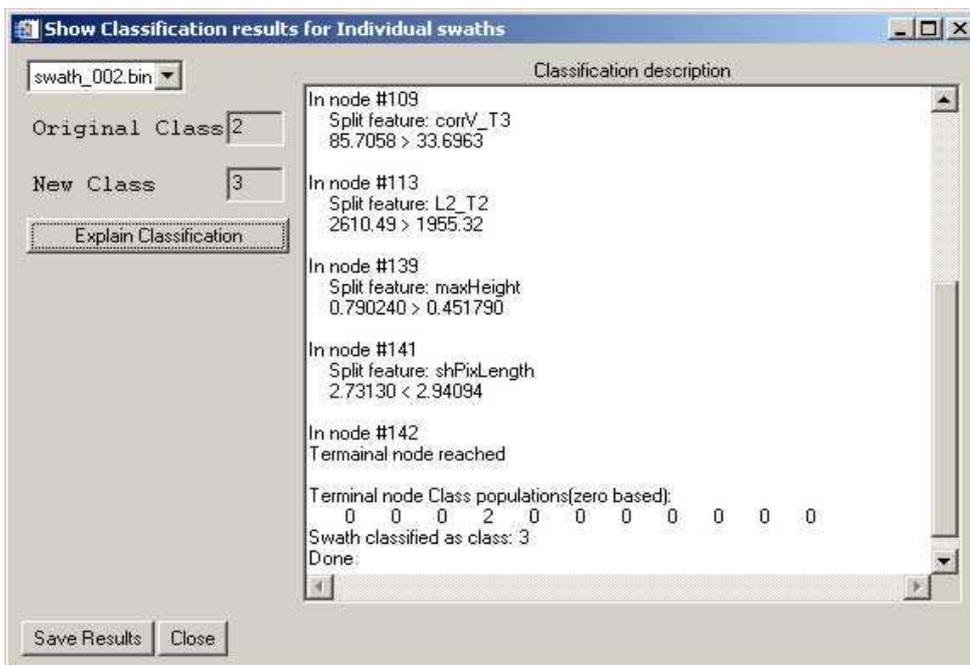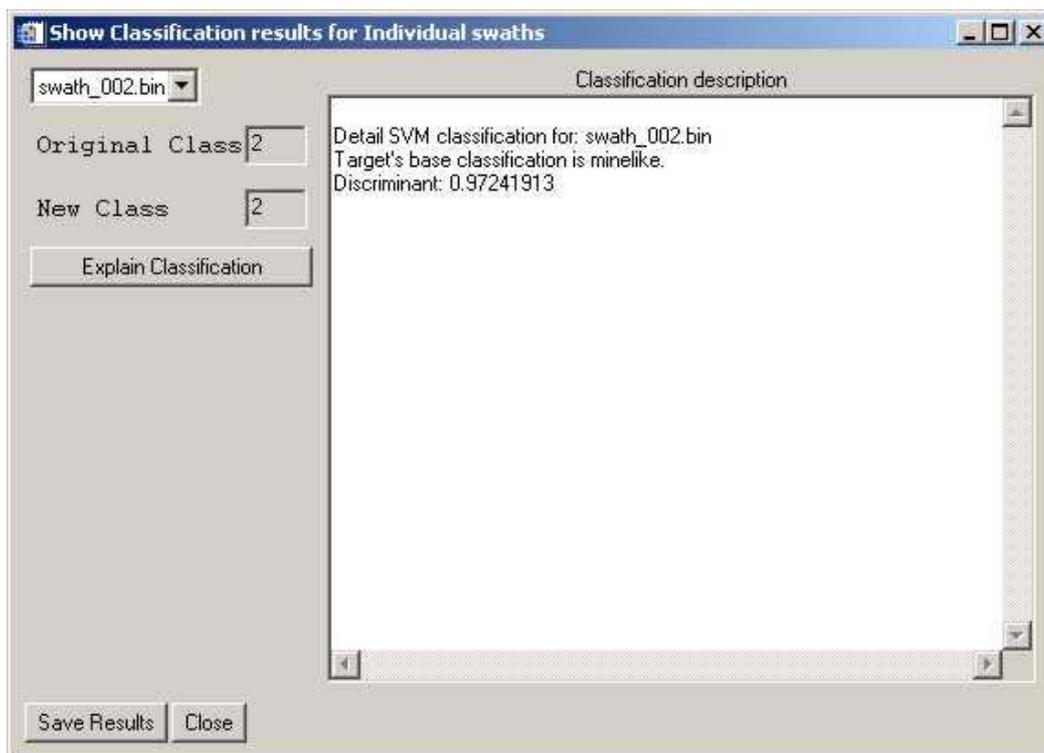


*Figure 9.* CART Tree Classification Results

### Support Vector Machine

The following information is displayed to the user for the SVM:

- Swath name

- Target Discriminant

- Indication of mine-like verses non-mine-like classification

Figure 10 depicts a window, which displays the SVM classification results.

*Figure 10.* SVM Classification Results

## Implementation of an Operator Display - Issues

There are no known issues at this time.

## Implementation of an Operator Display - Rollout

Rollout for the classification explanation display occurred in April 2004. It was modified in December 2004 with the addition of the SVM classifier.

## Implementation of an Operator Display - Recommendations for Follow-on

There are currently no recommendations pertaining to the implementation of an operator display requirement.

## Develop Multi-aspect Classification Tools (9. xi.)

This line item was dropped from the list of SIPS requirements. The implementation of template features has taken its place. The template features will be discussed in the next section.

### Develop Multi-aspect Classification Tools (9. xi.) - Recommendations for Follow-on

The following are recommendations as they pertain to the *Develop Multi-aspect Classification Tools* (9. xi.) requirement:

- Inclusion of the Develop Multi-aspect Classification Tools (9. xi.) requirement into the SIPS software product is recommended.

- Inclusion of the Develop Multi-aspect Classification Tools (9. xi.) requirement into the CAD/CAC software product is further recommended.

- The expansion of any Develop Multi-aspect Classification Tools (9. xi.) requirement, implemented within the CAD/CAC software product, into the SAC (within the Classification application) should also occur.

## Implementation of Template Features

Template features were added to SIPS, as requested by DRDC Atlantic. These features were added in place of the multi-aspect classification tools, which were dropped from the SIPS requirements list. There are currently 90 template features. These features are broken down into categories relating the different types of possible targets. These types are as follows:

- Manta

- Vertical cylinders

- 4 types of horizontal cylinders

There are currently no templates relating to sphere-like targets. The template images are discussed in the section describing the template data file.

Template features were based in the research paper [R-7].

### Template Data File

The template data file contains all of the template images. All information in the data file is stored in floating point format. The templates are broken down into six types. For each type, there are 16 different targets. These 16 targets represent targets at different swath ranges. Each of the 16 targets has three variations that are related to the towfish altitude. Towfish altitudes are grouped as:

- Below 12 m

- 12 m-18 m

- Above 18 m

For each of the altitude variations of the horizontal cylinder templates, there exist a further 13 variations. These variations correspond to the orientation of a target. The orientations range from 0 degrees to 180 degrees, in 15 degree increments. The number of targets, altitudes and the number of azimuth variations are stored as global variables within SIPS. If the breakdown of the data file changes, these globals must be updated to reflect the new file layout.

The currently used template file is broken down as follows:

- 48 manta templates

- 48 vertical cylinder templates

- 2496 horizontal cylinder templates, grouped into four types

The data for each template is preceded by the dimensions of the template. To easily access individual templates, an index file has been created. This file functions in a similar manner to the database index file. These indices then point to the first byte of the across track dimension of each template. If the data file changes, for any reason, the index file will have to be recreated. The index file is utilized via IDL's *assoc* command, which treats the index entries as elements of an array.

### Features Added

As previously discussed, the template features are broken down by template type. For each type the following 15 features are generated:

- L1 normalization

- L2 normalization

- L1 normalization of the absolute value of the template and image

- L2 normalization of the absolute value of the template and image

- Cluster length

- Cluster length of the absolute value of the template and image

- Ratio of the number of the shadow pixels in the template shadow to the number of template shadow pixels

- Ratio of the number of highlight pixels in the template highlight to the number of template pixels

- Ratio of the number of highlight pixels in the template shadow to the number of template shadow pixels

- Ratio of the number of shadow pixels in the template highlight to the number of template highlight pixels

- Ratio of the number of highlight pixels in the template highlight to the number of template highlight pixels, using the absolute values of the template and image

- Correlation of the horizontally differenced image and template

- Correlation of the vertically differenced image and template

- Correlation of the horizontally differenced absolute valued image and template

- Correlation of the vertically differenced absolute valued image and template

## Implementation of Template Features - Planning

The work to implement the template features was broken down into two areas:

- Generation of the template features in both the DM and the bulk processor

- Integrate the new features in the classification criteria and the classification algorithms

### *Load and Pre-process Swath Image*

To calculate the template features for a swath, several pre-processing steps are taken. Firstly, a copy of the image data is loaded. Like the rest of the SIPS features, the template features will be calculated based on a small subset of the image data. This data area is roughly 9.5 m in the across track dimension. All pixels in the along track dimension are used. Care is taken such that no water column pixels are part of this data area. At this time the average altitude corresponding to the data area is calculated. In addition, the swath range to the start of the data area is calculated.

The following pre-processing steps are taken:

- The image data is byte scaled.

- The image data is normalized by subtracting the image mean value.

- The data is re-binned, so that all image values above 20% and below 95% are set to zero.

### *Feature Generation*

As previously stated, the features are grouped into six types of 15 features each. For each type, the calculated average altitude and the swath range are used to determine the base starting template. For the four horizontal cylinder types, an additional step is taken. This additional step determines the best fitting azimuth template. This is done by convoluting the swath image with the 12 different azimuth templates and the absolute value of the template. The template with the highest sumCorrTotal value becomes the best fitting template.

sumCorrTotal is defined as:

sumCorrTotal = sumCorr1 + (0.2 * sumcorr2)

Where:

sumCorr1 = convolution peak value / sum of squares

sumCorr2 = convolution peak value (using the absolute value of the template and image) / sum of squares.

Once the final template decision has been made, the template features are generated.

### *L1 and L2 Values*

The L1 and L2 values are generated by first convoluting the swath image with the template. The peak value is divided by the sums and by the sums of squares of the template values to get the L1 and L2 values respectively. The absolute values are generated using the absolute valued versions of the image and template. The convolutions are currently performed by the C dll.

### *Cluster Length*

The cluster length is found by first:

- Re-binning the convolved image (all values below 85% of max are set to zero)

- Regions are labelled by using *idRegions8*

- Cluster length returned as difference of the max and min y values of the labelled region

### *Pixel Count Ratios*

The ratios are generated by first centering the template over the convolution peak value and creating a series of counts. The counts are generated in the C dll, using the function *shadowHighlight_count*. The following values are returned in a vector of *unsigned shorts,* named *count:*

- Number of image shadow pixels

- Number of template shadow pixels overlapping image shadow

- Number of image highlight pixels

- Number of template highlight pixels overlapping image highlight

- Number of template shadow pixels

- Number of template highlight pixels

- Number of template highlight pixels overlapping image shadow

- Number of template shadow pixels overlapping image highlight

The following ratios are generated using the counts:

- \# of template shadow pixels overlapping image shadow / \# of template shadow pixels

- \# of template highlight pixels overlapping image shadow / \# of template highlight pixels

- \# of template highlight pixels overlapping image shadow / \# of template shadow pixels

- \# of template shadow pixels overlapping image highlight / \# of template highlight pixels

The last ratio is generated using absolute valued versions of the image and template:

- \# of template highlight pixels overlapping image highlight / \# of template highlight pixels

### *Correlations of the Horizontally and Vertically Differenced Images*

These features are generated by first, filtering the image and template with a Gaussian filter. The differenced images are created in the C dll using the correlate_edges_diff function.

To create the horizontal value, the horizontally differenced image is convolved with the horizontally differenced template. The final value is then the peak value / sum of the template values.

To create the vertical value, the vertically differenced image is convolved with the vertically differenced template. The final value is then the peak value / sum of the template values.

The process is repeated for the absolute valued versions of the image and template. Figure 11 depicts the Database Management window displaying template features.
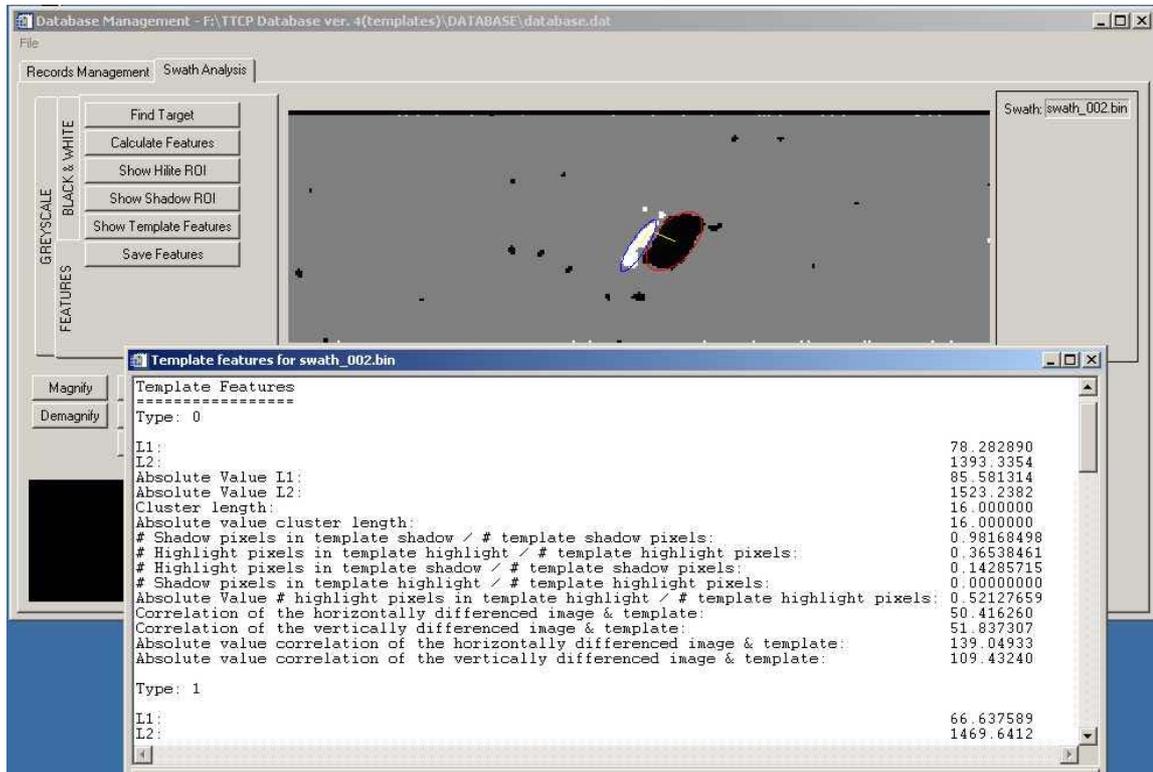


**Figure 11.** *Database Manager Displaying Template Features*

## Implementation of Template Features – Issues

Table 5 presents the issues of significant, and notable, relevance to the implementation of the Implementation of Template Features.

| Issue | Resolution |
|---|---|
| Sometimes the calculated cluster length was 0.0. It should be at least 1.0. | The port of *mlabel3* was replaced with the more stable *idRegions8*. |
| Cluster length was reporting fractional values. | Cluster length was mixed up with differenced values. |
| Horizontal and Vertically difference values were reporting integer value instead of fractional values. | Reversed cluster lengths with differenced values. |
| Wrong templates being loaded. | An off by one error occurred when returning the template number to load. A minus one was applied to the template number. |

*Table 5.* Template Features Issues

## Implementation of Template Features - Rollout

Initial rollout of the first batch of 75 template features occurred in October 2004. The efforts of COGS were then diverted to assisting in the integration of the basic SIPS features into CAD/CAC. Before the remaining features were rolled out in late November 2004, the underlying logic and source code of the classification criteria and algorithm training interfaces were re-factored. This will be discussed in a later section.

## Implementation of Template Features - Recommendations for Follow-on

The following are recommendations as they pertain to the implementation of template features requirement:

- Possible inclusion of templates-related configuration options in the SIPS configuration file. Additional configuration options could be the number of altitude categories or the number of azimuth categories for the template targets.

- The addition of more in-depth error checking of the templates data file.

- The establishment of a formal file format for the templates data file. This would aid in the error checking of the data file.

- The additional implementation of the template features in CAD/CAC.

- The expansion of any template features implementation into the SAC (within the classification application) should also occur.

# Re-factoring Classification Criteria Source Code

The underlying source code of the classification criteria and the algorithms training interfaces required substantial changes to easily accommodate the changes to the number of target features.

## Re-factoring Classification Criteria Source Code - Planning

Initial planning focused on how to improve and simplify the data structures relating to the storage of training and testing target features. Originally, because of the possibility of targets without highlights, target shadow and highlight features were stored in separate data structures. These data structures were arrays of structures specific to highlight and shadow features.

It was decided to amalgamate the highlight and shadow features into a single 2-dimensional array of target features. Columns would be features, with the last column holding the target class. The rows would be targets. A second 2-dimensional array will be used as storage for target feature populations.

## Re-factoring Classification Criteria Source Code - Implementation

The following data structures were retired from use:

- Sh_features (shadow feature values)

- Sh_pops (shadow feature populations)

- Hi_features (highlight feature values)

- Hi_pops (highlight feature populations)

- Populations (highlight and shadow features)

Features are now stored in a 2-dimensional array of doubles, with one row per target. The last column stores the target classification.

The features array is indexed using feature indices. Each feature used in SIPS is assigned an index value. These index values are read in from the config.txt file at SIPS startup. Indices are stored in the *FEATURECODES* global variable as an array of FCODES structures. The size of the array is equal to the number of features currently used in SIPS. The global variable TOTALFEATURES contains this value. The structure layout:

- Feature – Actual feature name

- Code – Code to indicate the feature and its grouping

- IDX – Feature storage index

Features are currently separated into three groups: Highlight, Shadow and Template. A feature's grouping can be determined from the code. Highlight feature codes contain "_h", shadow feature codes contain "_s" and the template feature codes contain "_t" plus a number

indicating what type of template. Both the feature values array and the populations array may be indexed using the feature indices.

The feature codes may be modified by the user with the "Update Feature Codes" interfaces, under the utilities menu.

The feature populations are now stored in a 2-dimensional array of integers. The row indicates virtual class and the column indicates feature. The feature populations are accumulated by examining a target's features. The value -999.0 is used to indicate a missing value. Currently, missing values should be limited to the highlight features. The algorithm for determining feature values is as follows:

> For each virtual class
>
>> For each feature
>>
>> - Get vector of feature values spanning all targets
>>
>> - Determine how many targets do not have -999.0 for feature
>>
>> - Store amount in cell (virtual class number, feature index)

The population values are only used to calculate the class means, standard deviations and the cross-covariance matrices, thus they are not stored, and they are not needed in algorithm training.

## Re-factoring Classification Criteria Source Code - Issues

There are no known issues at this time.

## Re-factoring Classification Criteria Source Code - Rollout

Rollout of this work was completed in November 2004.

## Re-factoring Classification Criteria Source Code - Recommendations and Follow-on

There are currently no recommendations pertaining to the re-factoring classification criteria source code requirement.

# Implementation of SIPS Feature Extraction Routines in the CAD/CAC System

At the Technical Review Meeting on 20 July 2004, hosted by MDA, it was decided that the feature extraction routines from SIPS would be implemented in the CAD/CAC system.

### Implementation of SIPS Feature Extraction Routines in the CAD/CAC System - Planning

The planning involved determining how to make the feature extraction routines most easily available for inclusion into CAD/CAC. It was decided that the routines would be ported to C and placed in the SIPS dll file for access by CAD/CAC. The CAD/CAC system would then utilize the dll as part of normal operations.

It was decided that any routines ported would pass all arguments directly, instead of using the argv/argc interface, which IDL uses.

### Implementation of SIPS Feature Extraction Routines in the CAD/CAC System - Implementation

The following feature extraction routines and tools were either ported to the C dll or re-factored to be usable by CAD/CAC:

- getRegionList
- fit_ellipse
- object height
- object length
- length from ellipse
- convexity
- lacunarity

The fit_ellipse routine is base on the fit_ellipse routine by Dr. David Fanning, Ph.D. of FANNING SOFTWARE CONSULTING (http://www.dfanning.com). It is copywrited as open source.

The chain hull algorithm, used by convexity, was taken from the website: softSurfer (www.softsurfer.com). It is copywrited 2001. The copywrite allows free usage and modification.

## Implementation of SIPS Feature Extraction Routines in the CAD/CAC System – Issues

Table 6 presents the issues of significant, and notable, relevance to the implementation of the Implementation of SIPS Feature Extraction Routines in the CAD/CAC System.

| Issue | Resolution |
|---|---|
| The convexity tool in IDL made extensive use of built-in IDL routines. An extensive rewrite was required. | The call to IDL's QHULL was replaced with the *chain_hull* algorithm. IDL's contour was replaced by the function *getPerimeterPixels*. The perimeter is calculated by *perimChain8*. |

*Table 6: CAD/CAC Porting Issues*

## Implementation of SIPS Feature Extraction Routines in the CAD/CAC System - Rollout

Rollout of the release of CAD/CAC that incorporates the feature extraction routines into the base product commenced in January 2005 to the RMTC located at the DRDC Atlantic lab.

## Implementation of SIPS Feature Extraction Routines in the CAD/CAC System - Recommendations for Follow-on

The following are recommendations as they pertain to the implementation of feature extraction tools in CAC/CAC requirement:

- The additional implementation of the template features in CAD/CAC

- The expansion of the implementation of the feature extraction routines into the SAC (within the classification application) is also recommended

# Report Conclusion and Recommendations for Follow-on

The following subsections presents the summary of work performed and the conclusions rendered resulting from the execution of the second year of the SIPS and RMS TD software development contract.

Further to this, the following subsections also present the reader with general recommendations for potential follow on taskings should funding be available in the future.

## Report Conclusion

The beginning of year 2 produced the completion of the software Amendment 001 to the original contract. This software amendment was for the implementation of the CAD/CAC software package into the MDA-developed SAC software in use by the Route Survey offices of both coasts.

The finalization of the Implementation of the CAD/CAC software in the upgraded Route Survey Data Analysis Facility (RSDAF) SAC software (New Requirement) was rolled out in May 2004 to the RMTC facility in preparation for the Spring 2004 trials at Canadian Forces Base (CFB) Esquimalt.

Trials support to DRDC Atlantic for the Spring 2004 SIPS/RMS TD sea trials was an additional amendment to the contract. The proposal for this amendment was submitted to the customer, and the customer then turned this proposal into an official Contract Amendment 002 against contract no. W7707-032060/ 001/HAL.

MDA provided the support necessary to DRDC Atlantic for the Spring 2004 SIPS/RMS TD sea trials. These trials occurred during the period of May-June 2004 with both pre-trials and trials support taskings being accomplished.

The finalization and implementation of the Implement Three Classification Algorithms (9. vi.) requirement, including the SVM portion of the Implement Three Classification Algorithms for use with SIPS requirement, was accomplished in year 2 of the contract.

As part of a Technical Progress Review Meeting, a re-prioritization of the RTM occurred. Requirement 9. x. was pushed up in priority for CAD/CAC, while requirements 9. iv., 9. v., 9. vii. and 9. viii. were lowered in priority. It was also determined that requirement 9. ix. is satisfied within the existing CAD/CAC detection code developed under Contract Amendment 001.

Other areas of progression during the course of the second year of this contract are as follows:

- Additional improvements to the of feature extraction aspect of the SIPS software product were rendered

- Development of a GUI into the SIPS software product occurred

- A work plan for mid-term software maintenance was approved and implemented

- Implementation of template derived features were added into the SIPS software product

- COGS completed an upgrade to IDL version 6.1

- Product build releases and general upkeep/maintenance to the CM server (Fireball) continued throughout year 2 of the contract

- Software maintenance to the RMTC located at the DRDC Atlantic lab also occurred throughout the second year of the contract

Lastly, the implementation of the Work Plan for Feature Extraction into the CAD/CAC software occurred throughout the fall of 2004, with rollout commencing in January 2005.

## Recommendations for Follow-on

The following are general recommendations as they pertain to SIPS Development and Implementation of Classification Tools with Sidescan Sonar Image Database:

- The examination of the IDL virtual machine as a way to make SIPS more portable

- The establishment of a naming convention for the different input and output files that are utilized by SIPS

- The amalgamation of all of the configuration options into one interface or an external setup utility

- The addition of parent classes to the SIPS class list (parent classes would be cylinder, manta, sphere, etc.)

- The inclusion of the "Implement 3 classification algorithms (9. vi.)" requirement into the, as-delivered, CAD/CAC software product is recommended.

- Consideration should be given to the identification and revision of those design, software and interface specifications, developed under the RMS TD contract (contract no. W8477-8-0010/001/SV), which DRDC Atlantic considers value-added publications

# References

1. Vincent L. Myers, Decision trees for computer-aided detection and classification (CAD/CAC) of mines in sidescan sonar imagery, DRDC Atlantic TM 2002-144.

2. Quidu, I. Malkass, J.PH., G. Burel and P. Vilbe, Mine classification using a hybrid set of descriptors, In Oceans 2000 Conference Proceedings, Vol. 1, pp. 291-297, MTS/IEEE, New Hampshire, USA.

3. Ronald T Kessel, Decision Using sonar speckle to identify regions of interest for mine detection, DRDC Atlantic.

4. N. Cristianini and J. Shawe-Taylor, "An introduction to Support Vector Machines and other kernel-based learning methods", Cambridge University Press, 2000.

5. Ronald T. Kessel, "Using speckle to identify regions of interest and form mine detection, 2002. (The lacunarity scan was derived from this paper.)

6. G.J. Dobeck, J.C. Hyland and L. Smedley, Automated detection/classification of seamines in sonar imagery, Proc. SPIE-Int. Soc. Optics, vol. 3079, pp. 90-110, 1997. (The matched filter algorithm was based on this paper.)

7. John Fawcett, Computer-aided detection and classification of minelike objects using template-based features, Proceedings of Oceans 2003, San Diego, September 2003. (Template features were based on this research paper.)

# List of symbols/abbreviations/acronyms/initialisms

| | |
|---|---|
| ASCII | American Standard Code for Information Interchange |
| CAD/CAC | Computer-Aided Detection/Computer-Aided Classification |
| CFB | Canadian Forces Base |
| CM | Configuration Management |
| COGS | Centre of Geographical Sciences |
| DM | Database Manager |
| DRDC | Defence R&D Canada |
| GUI | Graphical User Interface |
| IDL | Interactive Data Language |
| IPT | Integrated Project Team |
| MDA | MacDonald Dettwiler and Associates Ltd. |
| MLO | Mine-Like Object |
| RBF | Radial Basis Function |
| RMS | Remote Minehunting System |
| RMTC | Remote Minehunting Tactical Center |
| ROI | Region of Interest |
| RSDAF | Route Survey Data Analysis Facility |
| RTM | Requirements Traceability Matrix |
| SAC | Sonar Analysis Console |
| SIPS | Sonar Information Processing System |
| SOR | Statement of Requirement |

SVM           Support Vector Machine

TD             Technology Demonstrator

# Distribution list

## Internal Distribution

David Hopkin, Mine and Torpedo Defence Group, 9 Grove St.,
Dartmouth, Nova Scotia  B2Y 3Z7

Anna  Crawford, Mine and Torpedo Defence Group, 9 Grove St.,
Dartmouth, Nova Scotia  B2Y 3Z7

Juri Sildam, Mine and Torpedo Defence Group, 9 Grove St.,
Dartmouth, Nova Scotia  B2Y 3Z7

John Fawcett, Mine and Torpedo Defence Group, 9 Grove St.,
Dartmouth, Nova Scotia  B2Y 3Z7

Library (6)

## External Distribution

NDHQ/DRDKIM

Troy Kennedy, McDonald Dettwiler and Associates Ltd., Suite 60, 1000 Windmill Road,
Dartmouth, Nova Scotia  B3B 1L7

LCdr  J.T. Hewitt, DMRS 3-2, NDHQ, 101 Colonel By Dr., Ottawa, Ontario  K1A 0K2

<table>
<tr><td colspan="3" align="center">**DOCUMENT CONTROL DATA**<br>(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)</td></tr>
</table>

| | | |
|---|---|---|
| 1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)<br><br>Defence R&D Canada – Atlantic, PO Box 1012, Dartmouth, NS  B2Y 3Z7 | 2. SECURITY CLASSIFICATION<br>(overall security classification of the document including special warning terms if applicable).<br><br>UNCLASSIFIED | |

| |
|---|
| 3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).<br><br>Development and Implementation of Classification Tools with Sidescan Sonar Image Database – Final Report |

| |
|---|
| 4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)<br><br>Troy Kennedy, Arthur Abbott |

| | | |
|---|---|---|
| 5. DATE OF PUBLICATION (month and year of publication of document)<br><br>August 2005 | 6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc).<br><br>42 (approx.) | 6b. NO. OF REFS (total cited in document)<br><br>7 |

| |
|---|
| 7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).<br><br>Contract Report |

| |
|---|
| 8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).<br>Defence R&D Canada – Atlantic<br>PO Box 1012<br>Dartmouth, NS, Canada  B2Y 3Z7 |

| | |
|---|---|
| 9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).<br><br>11CL | 9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).<br><br>W7707-032060/001/HAL |
| 10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)<br><br> DRDC Atlantic CR 2005-054 | 10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.) |

| |
|---|
| 11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)<br>( X ) Unlimited distribution<br>(    ) Defence departments and defence contractors; further distribution only as approved<br>(    ) Defence departments and Canadian defence contractors; further distribution only as approved<br>(    ) Government departments and agencies; further distribution only as approved<br>(    ) Defence departments; further distribution only as approved<br>(    ) Other (please specify) |

| |
|---|
| 12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected). |

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This report describes the associated activities pertaining to the second year of the 2-year "Development and Implementation of Classification Tools with Sidescan Sonar Image Database" contract (contract no. W7707-032060/001/HAL). This report is to provide the reader with a summary of the progress against the requirements of contract no. W7707-032060/001/HAL as it pertains to the technical topics of discussion.

This publication will provide the reader with detailed technical progress and implementation activities associated with the satisfaction of requirements progressed during the second year of this 2 year contract. Additionally, this report will provide the reader with information gathered throughout the second year of this 2 year contracts, in addition to presenting a Requirements Traceability Matrix (RTM) of the most current status of the requirements of the contract.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

sidescan sonar, classification

This page intentionally left blank.

## Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

## R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE **R&D** DÉFENSE

**www.drdc-rddc.gc.ca**