

A GEOCODED DATABASE FOR LANDCLASSIFICATION OF ERS-1 DATA

Karlheinz Knipp
Institute for Photogrammetry and Engineering Surveys
University of Hannover, FRG
ISPRS Commission III

Abstract :

A geocoded database has been designed to support investigations on geometric correction and classification of ERS-1 imagery. The database' conception is that of a "meta-database", i.e. it manages information about objects rather than storing the objects themselves.

Object are project-relevant spatial data which form different classes like geocoded images (SAR, different scanners), training-areas for classification and classification results.

Routines for the information-retrieval as well as for meta-data-editing (new, edit, delete entry) have been developed. Another module for the graphical selection of object exists, which can be used from inside the database as well as from other project-applications.

A layer is implemented around the database which may be expanded to a project-informationsystem (IMC : Integrated Meta-database in CALA¹).

KEY WORDS : Classification, Data Base, SAR

1. INTRODUCTION

In July 1992 ERS-1 (European Remote Sensing Satellite) was successfully launched into orbit. One instrument on board is AMI (Active Microwave Instrumentation), designed either to operate in the SAR (Synthetic Aperture Radar) - or scatterometer - mode. The SAR-data provide useful information for various applications, p.e. earth resource management. The project CALA (Calibration and Land Application of ERS-1 SAR Data) in general deals with various aspects of SAR-sensing : calibration, geocoding, forestry, agriculture, hydrology, etc.. The Institute of Photogrammetry and Engineering Surveys (IPI) is especially involved in the geocoding and classification part of CALA. The investigations

are titled : Determination of land-application data on the basis of geometric correction, automatic correlation and geocoded database (Fig. 1.1).

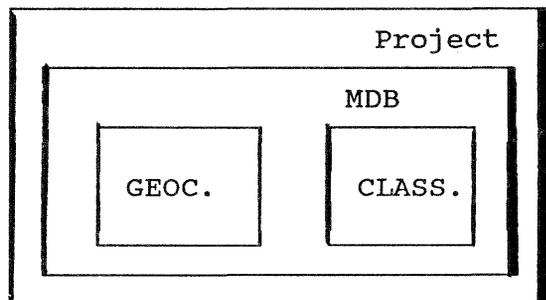


Figure 1.1 : Project - scheme

The techniques used for geometric correction are described in [Sasse, 1992]. This text deals with the design and implementation of the database. Abbreviations : DB and MDB for database and meta-database.

¹ CALA : Calibration and Land Applications of ERS-1 data

2. DESIGN

2.1 General Design

The project demands the handling of various kind of spatial data like images from several sensors, training areas and classification-results. Keeping track of all the information to the data (image-sizes, ground-coordinates, etc.) is quite tedious and leads to long listings (on paper) with which nobody can efficiently work. So we let the computer store that information in a database.

It has been decided to create an own database for the project and not to use a commercial one. Objectives for this decision has been the ability to design a system especially dedicated for the project and therefor being efficient [Lee, 1988], no need for relational calculus [Schumacher, 1991], the lack of money to purchase a commercial DBMS and the advantage to work in an homogeneous environment for the entire project. This environment is the VDA (Visual Data Analysis) software-package PV-WAVE, in which about 90% of the projects modules are developed. Especially routines like user-interfaces or analyzing features are written in WAVE, dedicated and time-consuming subroutines are added in FORTRAN or C [Sasse, 1992]. The database will be entirely developed in WAVE.

A definition of a database can be found in [Deen, 1985] : "A generalized integrated collection of data together with its description, which is managed in such a way that it can fulfil the differing needs of its users". This means more than just a collection of data. What we have to do is to define what information we want to combine with our data and how this information can be stored

efficiently. Concentrating on the first and important data we work with, the image-data, we find, that it is not possible to store all data on disc at the same time (the project-workstation has two discs with 600 MB each). This leads to the concept of a meta-database [Meissner, 1990], thus spoiling our definition from data stored together with its description. A meta-database is best described with the example of a fictive literature DB: Rather than actually storing the content of each book entered in the database just the information like author, title, short abstract, etc. is collected, i.e. the objects themselves are physically not stored in the database. With this concept it is possible to store our objects, the images, on tape or somewhere else in a network of workstations.

We also have different kind of data to work with, beneath images training areas and classification results. It may be useful integrating GCP - and DEM-data. So the objects are grouped in classes of objects (clusters).

We have to write modules to work with the database : modules for data-definition and -editing (new entry, edit entry, delete entry) and others for information-retrieval (show all objects in a specified area). These modules have to be designed class-oriented, i.e. a module has to serve its purpose for all objects in a single class.

2.2 Data Models

The physical model describes how the data are actually stored. In our MDB we distinguish three degrees of file-levels. There is one 1st degree file containing the list (catalog) of all

objects in the database. 2nd degree files store keywords for single classes, while 3rd degree ones are files containing the object and files storing the meta-data to the object. So to each object two files exist : one file with the actual data, p.e. the image, and another one with the information (meta-data).

The logical model hides the actual data structure from the user. Fig 2.1 shows a scheme of our logical model.

user-interfaces to control these routines. The modules are coded in WAVE. The interface should be a graphical one, mainly controlled by mouse and cursor. WAVE offers a possibility to create such interfaces with a tool called WAVE-WIDGETS. Unfortunately this tool does just work under the window-system SunView. The project needs something to work under both SunView and X-Windows (with the advantage of network-processing ability). So first we created a low-level menu-system

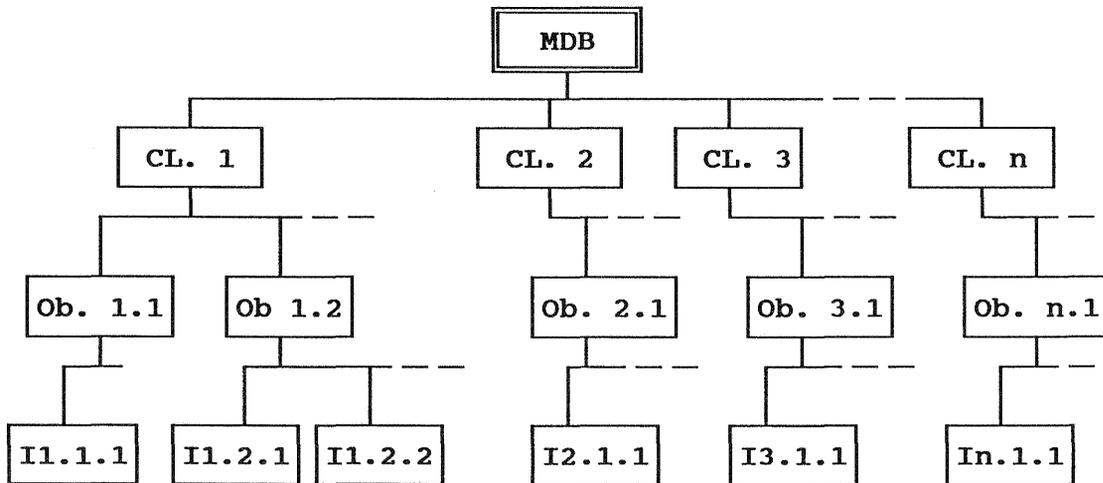


Figure 3.1 : Logical Model (Cl - cluster, Ob - object, I - Info)

Main elements are the objects, p.e. a SAR-image. Objects are arranged in clusters (classes of objects with identical data structure). All clusters together form the DB. To each object a set of meta-data is stored.

2.3 Database Modules

To work with the database we need modules to manage data and retrieve information as well as

(WaveMenu), which simply opens a window and draw some buttons. These buttons may be pushed by clicking on them with the mouse. According to which button is pressed a module, p.e. edit an entry, is called. Since all routines are programmed modular, it should be easy to adjust the entire system to a more sophisticated menu-system, which in future may be available for WAVE under X-Window. Fig. 3.1 shows an example of a menu (here IMC-main-menu)

As already mentioned the MDB is organized in clusters. So the modules for data-management and information-retrieval have to be coded according to this system : a module has at least to be valid for all objects in a class. Of course it is even better if it can be used for other classes as well.

3. IMPLEMENTATION

This chapter describes the implementations done so far. Around the MDB exists a shell offering access to the database as well as to other project-applications. This shell is called IMC (Integrated Metadatabase in Cala).

3.1 IMC

Fig. 3.1 shows the main menu of IMC. It can be divided in three parts : the database, project-applications and system maintenance. The database - part will be described later in the text.

Since concentration in the work has been laid on the database-design and -coding there are not many implementations to describe for either applications and maintenance. However the concept is already laid out: User applications are tools like image-display, format-conversion-routines, training-area definition or interfaces to project-applications like geocoding and classification. IMC-Maintenance include database-checks and backup-facilities. Also existing are the following system-tools: a HELP-facility, which displays info-texts on screen, and IMC-EXIT. Each sub-menu (no matter if database- or application - menu) will feature these two points (HELP,EXIT) plus a "return-to-higher-level" button.

3.2 Meta-database

3.2.1 Meta-data

Up to now just one class of objects has been completely defined - cluster 1 : geocoded images. It is already mentioned that the database is designed to handle meta-data, i.e. the object itself may be stored on a tape and therefore be not accessible for p.e. a quick-look. Since this is rather inconvenient when dealing with visual data we integrated the concept of an image-chip. For each image a minification of 128 by 128 pixels is created (the larger of both dimensions - columns or rows- defines the reduction-scale; before resizing a low-pass-filter with a kernel-size according to the reduction-factor is applied to the image). This image-chip is added to both the image (as a footer) and to the information stored for the image. The following shows a list of information collected for each object of this class :

- object name
- physical place where object is stored
- dimensions
- ground-resolution
- test-area
- ground-coordinates (image-corners and corners of image-information)
- sensor
- image-source for geocoding
- date of recording image-source
- comment to object
- image-chip

This list may be frequently expanded. The information is accessed via keywords (p.e. area for test-area) so that there is no need to store them in a strictly formatted list.

3.2.2 The Modules

On the higher level -the main menu in IMC - the database-

applications are already divided into Information-retrieval, Data-management and Maintenance (refer to Fig. 3.1).

Besides the obligatory HELP-, RETURN- and EXIT-functions the Information-menu offers two ways of selecting an entry (one by selecting from the catalog, the other by a graphical display of all objects in a specified test-area) and a facility to display all information stored to an object (Fig. 3.2). Both selection-tools may also be used to get an overview what entries are stored in the MDB. The graphical one demonstrates the modular philosophy in IMC : a) it is defined as a function to be called by any other Wave-routine and therefore open for acces from both inside and outside the MDB and b) it again features the tools just mentioned : select by catalog and display information. The latter (Fig. 3.2) offers the possibility to either store the display as an image (in SunRaster format) or send it to a postscript-printer.

For Datamanagement all necessary modules have been implemented, a) enter new entry, b) edit entry and c) delete an entry. Before storing edited information or removing an entry a backup of the old information is created.

Maintenance of the database consists so far of simple consistency checks : Each entry in the catalog is checked for existence of its meta-data and vice versa. If an inconsistency is detected recovering via the backup-information can be tried.

4. CONCLUSIONS

The designed database serves its purpose as a project-oriented information-system. It is open

to be called from application-routines outside the MDB as well as to serve as interface to the applications.

Next steps will include the definition of more clusters. Right now work is being done to install modules for the definition of training-areas for the classification. The defined areas will not be stored in respect to an image but to the ground-coordinate-system.

Literature :

- Deen, S.M. (1985) : "Principles and Practice of Database Systems"
MacMillan, Southampton.
- Lee, Y.C. (1988) : "An Introduction to Digital Mapping"
Lecture notes no. 56
Department of Surveying Engineering,
University of New Brunswick 1988.
- Meisner, R. (1990) : "Entwicklung eines Informationssystems fuer Fernerkundungsanwendungen"
Ludwig-Maximilians-Universitaet
München.
- Sasse, V. (1992) : "Object Space Reconstruction for ERS-1 Stereo SAR"
Submitted paper to ISPRS Commission II
Washington 1992.
- Schumacher, H. (1991): "Untersuchungen zur ueberwachten Klassifikation von Fernerkundungsaufnahmen"
Wissenschaftliche Arbeiten der Fachrichtung Vermessungswesen an der Universitaet Hannover.

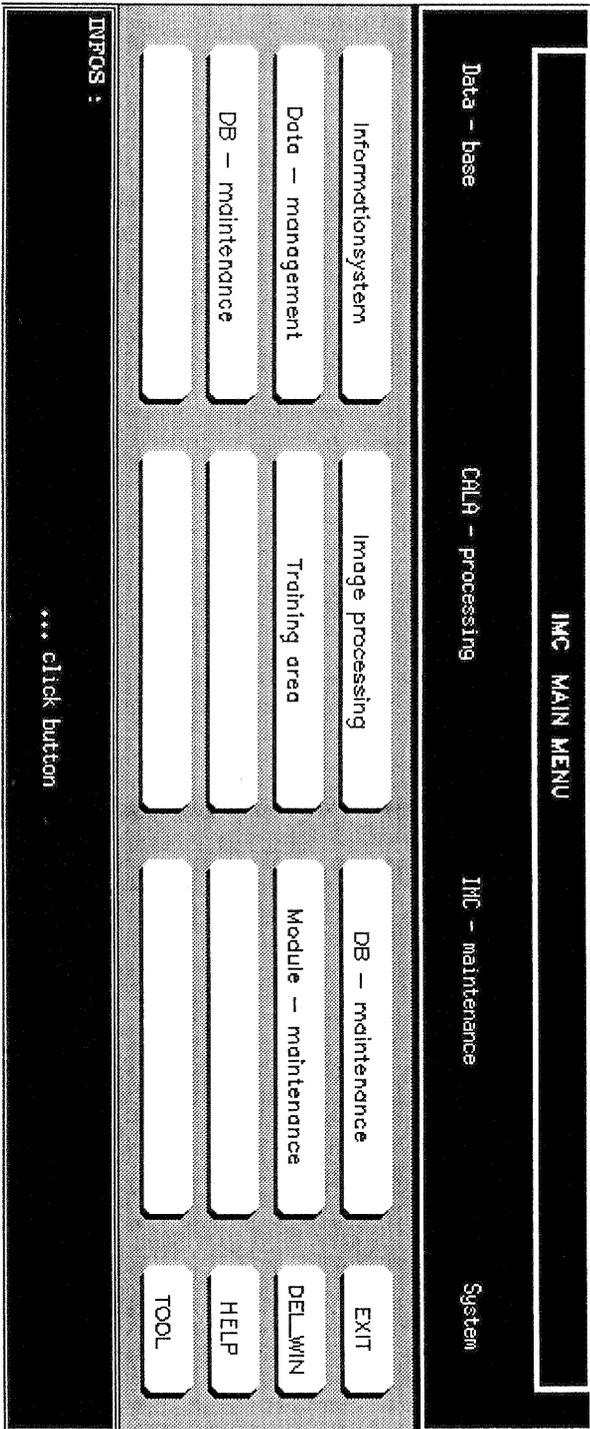


Figure 3.1 : Main Menu of IMC

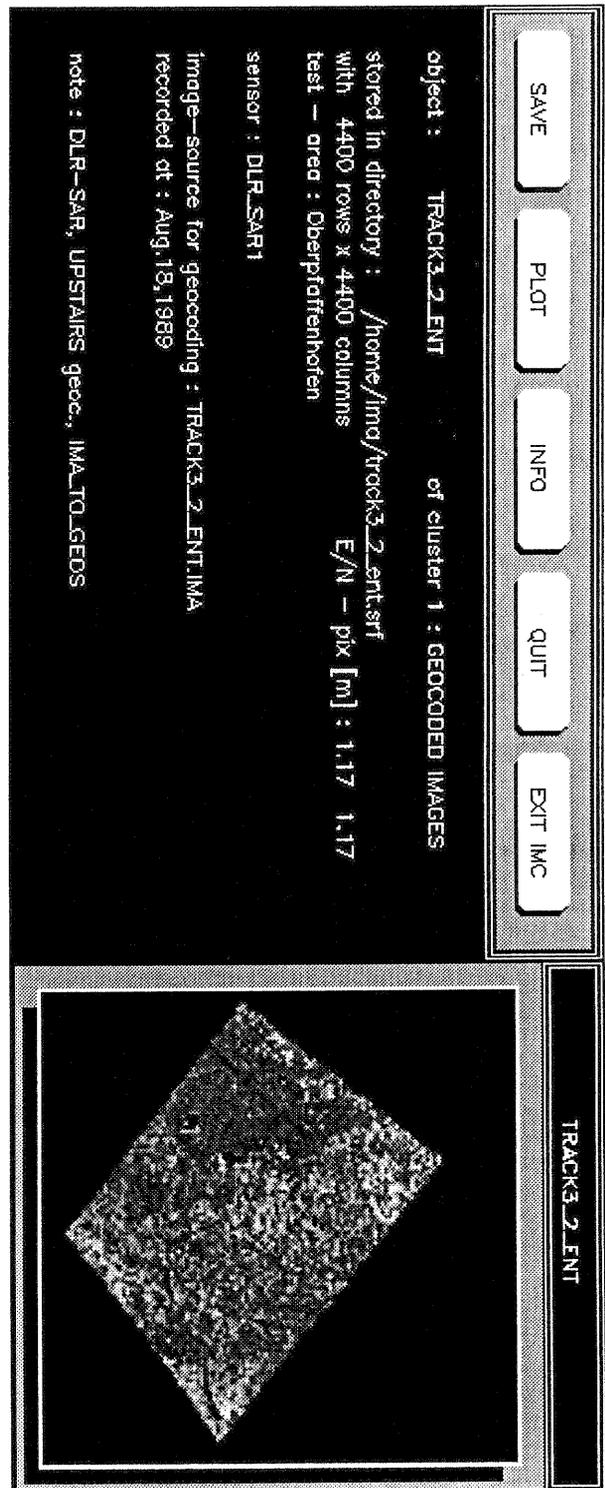


Figure 3.2 : Information-Display in IMC