Computer Assisted Problem Solving In Image Analysis

Fang Luo
N.J. Mulder
International Institute for Aerospace Survey and Earth Sciences (ITC),
P.O.Box 6 , 7500AA Enschede,
The Netherlands.

ABSTRACT:

The researcher / user of software for digital image analysis is confronted with huge libraries of subroutines. In order to solve a problem from a set of problems, it is, in general, not clear which subroutines should be selected, with what setting of the parameters. The authors have set out to structure and classify subroutines generally available in image processing and image analysis libraries as a first step in bottom up knowledge engineering. In order to reduce the redundancy in the large sets of subroutines, a virtual image analysis engine has to have a minimum(reduced) instruction set.Computer assisted problem analysis is approached in a top down manner. A PROLOG style specification language is developed, which allows goal directed programming. This means that the problems have to be specified in terms of relations between the components of a model.The language will check whether the number of constraints is sufficient, and if so, will solve the unknown(s). Often a search of problem space has to be performed where an optimisation criterion is required (cost function). The criterion used here is minimum cost/ maximum benefit of classification or parameter estimation.

Key words : image processing, expert system, knowledge based system, image analysis, knowledge representation, reasoning.

1 INTRODUCTION

In the image process domain, a variety of image processing algorithms have been devised to facilitate image analysis. Various software packages for image processing include many techniques advanced in the history of digital image processing. These software packages can be used efficiently in problem solving by only a few experienced people; they offer many choices of subroutines and often require a large set of parameters to be defined by the user. Choosing subroutines and parameters may prove to be quite a complex task, although expert users of such packages may find it easy.

One way to make these software packages more manageable and usable by a wider user community is to capture the knowledge of expert users in controlling these software systems. We can visualize this as an expert program that monitors the use of the software package, helps the user in understanding and controlling the package and also provides interpretation of the results produced by the user's interaction with the program. This expert system would, therefore, contain domain knowledge to be used in choosing the appropriate methods and techniques from the software package.

The following problems are encountered in designing such a system :

1. Assessment of image quality. To measure the quality of an image
is the first step in image analysis. The quality of an image is defined by its potential in providing information about a class of objects in the scene. One decision based on the evaluation of the quality is to model errors and artefacts and remove them as well as possible by inversion of the error model.

2. Selection of appropriate procedures. There are many different procedures (algorithms) for a specific image processing task. They are designed on the bases of different image models and computation schemes. One has to select appropriate procedures considering image quality, the purpose of image analysis and characteristics of the procedures.

3. Determination of optimal parameters. Many procedures have adjustable parameters, performance is heavily dependent on the values of the parameters.

4. Combination of primitive procedures. It is often necessary to combine many primitive procedures to perform a meaningful task. For example, a popular way of extracting regions from an image is to apply edge detection --> edge linking -->closed boundary detection. To attain effective combinations, knowledge about image processing techniques is required.

5. Trial_and_error experiments[13]. It is usually very hard to estimate a priori the performance of a procedure for a given image so that one has to repeat experiments by modifying parameters. The definition of a cost / performance criterion allows the use of numerical optimisation [14].

Recently, several knowledge based systems for image processing were developed to facilitate the development of image analysis processes. These incorporate knowledge engineering(KE) techniques to solve the above problems. Examples are: a consultation system for image processing[3], a knowledge-based program composition system[11,12] and a goal-directed image segmentation system[10,13]. Here we present a knowledge based method to provide users with the main functions: hypothesis generation from queries, the organisation of processing sequences [7] and the setting of parameters of subroutines.

2. DESIGN OF AN ADVISORY EXPERT SYSTEM FOR IMAGE ANALYSIS

At ITC and UT(University of Twente), there are at least six librariesof subroutines for image processing available. Each of these has its own characteristics and advantages. We categorize these subroutines according to their processing functions. We make an expert program to organize (by grouping and classification) the subroutines, and select them using heuristic information. A PROLOG style specification language is being developed to handle the problem analysis (logic of selection) and the running of compiles subroutines. This provides the user with an advisor for planning a process sequence and for setting the parameters.

There are many redundant procedures in the image processing libraries. We keep these redundant procedures but provide users with a minimum (reduced instruction) set of a non existing (virtual) machine . This requires the administration of equivalence or the reorganisation of subroutines into a more orthogonal set with appropriate parameters.

An example of elements of a reduced instruction set for image analysis is based on the treatment of a complete image as one object in a register. Typical Operations are : Copy(from image register_A with shift(dx,dy), multiply with a constant and accumulate result in image register_B) , Pack bytes from(A,B,C) into X, sort X, generate Frequency_of_coincidence(X -> A,B,C), Select maximum frequency (likelihood).

In order to select the best sequence of subroutines and their parameters we must define evaluation functions in order to enable standard optimisation software [14] to select procedures, using minimum cost / maximum benefit .The value of the evaluation function is the weighted sum of all factors (e.g. the intention of a user, length of path, time consumption, software uniform etc).

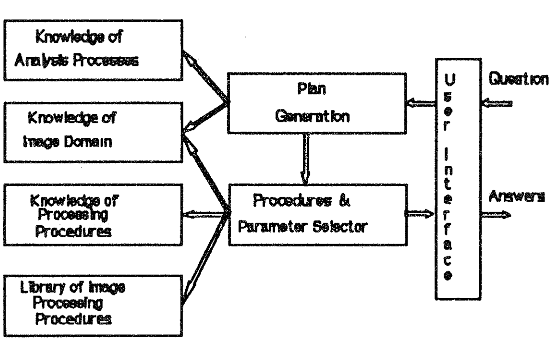

Figure 1. Architecture of the system

Figure 1. illustrates the general architecture of the system. It consists of a reasoning engine, the knowledge about image processing techniques, a library of image processing procedures and a database of characteristics of the input and processed image data; procedures in the library are applied to analyze the image and the result is stored in the database. The reasoning engine uses the knowledge about image processing techniques and characteristics of the image data for reasoning.

To develop the expert system for image processing, it is very important to create a programming environment. As we know, there are two kinds of programming style. One is the imperative style which tells the machine exactly what to do,such as FORTRAN, C, etc. Another is the declarative style which only describes the domain problem and lets the machine take over the problem solving, such as PROLOG, LISP etc. So far PROLOG is a rather successful language in AI research. But it is not satisfactory enough to develop an expert system for image processing, because its computation capability is too low to meet the needs of computation in image processing. It is essential to develop an application language which has both powerful capability of describing and of solving problems.

Although the construction of a general knowledge-based image processing scheme is a very long-range goal, the appropriate combination of state-of-the-art techniques can solve a class of problems in a specific domain. Under the given computational environment at ITC, we develop a knowledge based system which is able to automatically plan the processing sequence and select the arguments for the user, using PROLOG for the advisory/planning part and linking it with compiled subroutine libraries. In section 3, we present a knowledge representation scheme in the system. In section 4, we describe the reasoning process.

3. KNOWLEDGE REPRESENTATION

To create programs that have "intelligent" qualities, it is necessary to develop techniques for representing knowledge. Unlike people, computers do not have the ability to acquire knowledge on their own. Any knowledge they contain about the world has been explicitly provided in the form of data and knowledge structures.

Knowledge structures are usually closely tailored to specific problem areas which are called problem domains. The domain is defined by the set of relevant information required to solve a specific problem. For a case study we selected a software package named SPIDER which consists of over 400 FORTRAN subroutines for various image processing algorithms.

The knowledge used by the system comes from three aspects:

(1) The knowledge of image processing algorithms.

This deals with the usage of algorithms, condition for arguments and the range of the parameters. We can find this sort of knowledge in the manual of a software package.

First of all, we divided SPIDER subroutines into groups, and found their common parameters for each group. Figure 2. illustrates the relation between subroutines. From the figure, we find that frames are rather suitable to represent these subsets of procedures.

Using figure 2. as a guide for frame layout, we can embed the usage of the algorithm, conditions for arguments and data types into frames.

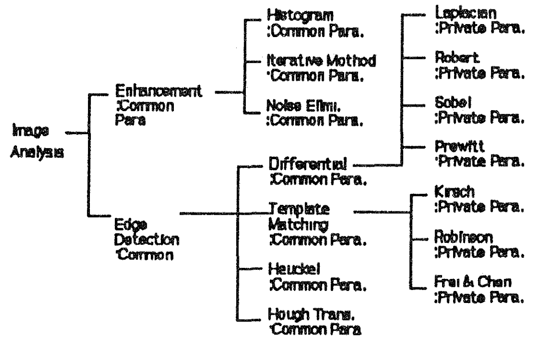

Figure 2. the hierarchy of subroutines in SPIDER

Frame Title: ETOC1
Frame Usage: ETOC1(IP,TH,JRT,ISX,ISY,ITH,IRO)
Parent: Edge Detection
Comment: " Detecting straight lines in an image"

/* private parameters slots */
Slot Name: JRT
Data Type: Dimension
Horizontal: ITH
Vertical: IRO
Comment: "2-dimension histogram of (ρ,θ)"
Slot Name: ITH
Data Type: Integer
Upper Range: 32768
Lower Range: 1
Default: 180
If-Needed: Manual
Auto
Comment: "Number of quantization for θ"

Slot Name: IRO
Data Type: Integer
Upper Range: sqrt(ISX+ISY) * 1.414
Lower Range: 0
Default: 260
If-Needed: Manual
Auto
Comment: "Number of quantization for ρ"

Table 1: Structure of the Hough Frame

The frame concept , Marvin Minsky[4] , consists of dividing knowledge up into specified categories. Frames function like forms. They are often implemented as form-like data structures in which the information in a given category is hierarchical.

Like the entries in a form, frames can have numerous slots or places where information can be stored. Another important feature provided with frames is the fact that the slots can have default values or procedures. This means that it is not necessary to describe in detail all of the facts about a given object.

A frame consists of a slot for storing general information about the frame itself, such as titles and usage. The "parent" field contains the name of the frame that references this one. The level field indicates the level of hierarchy of this frame. In addition, each frame has slots. The specialization of slots is used to establish a property inheritance hierarchy among the frames, which in turn allows information about the parent frame to be inherited by its children. After a particular frame has been selected to represent the current situation, the primary process in a frame-based reasoning system is often just filling in the details called for by its slots, and the data type will be checked. Some parameters are directly inherited. If there are no specifications, the default value can be used, or the attached If-Needed procedure can be used to decide. Table 1. shows thestructure of a Hough frame. There is no limit to the number of slots that a frame can have. Slots can also have support fields or attributes. These fields help define and describe the value of the slot. For example, the limit for numeric slot values is given by the upper and lower range values. The default value is the value used if no other explicit information is available. To each frame there are some specific slots. Default and inherited values are relatively inexpensive methods of filling in slots; they do not require powerful reasoning processes. Thesemethods account for a large part for the power of frames. When the needed information must be derived, attached procedures provide a means of specifying appropriate methods. This representation allows for more flexibility and greater accessibility to the knowledge.

(2) The knowledge of plan generation.

Although there is much knowledge about image processing, by way of image analysis strategies, for example to detect the edge in an image, there is an optimal solution in SPIDER: Sobel (differential), Kirsch (template matching type), Frei & Chen and Hueckel. From our experiment, Sobel , Kirsch, Frei & Chen may give basically the same type of image. Kirsch consumes more time, Hueckel's algorithm has the advantage of providing an equation for the edge-line pattern detected inside the area of analysis; this equation can define the location of the edges and lines within a subpixel resolution, which could be used for registration purposes for processing a scene. It is better to use a line-following algorithm in combination with the edge-line detection program in order to obtain a continuous type. We incorporate the above knowledge as gained from the experiment into our system as heuristic information to guide our search.

Many analysis strategies have been proposed to increase the performance of image analysis. Here we use such a scheme to represent the image analysis strategies. We describe an image processing procedure by a function. The result of applying procedures O to image D is denoted by O(D), thus the sequential composition of procedures can be described as

$$O_n(O_{n-1}(\ldots O_2(O_1(D))))$$

where D denotes an input image and O1,O2,...On-1,On functions represent image processing procedures. These functions are successively applied in this order: the innermost function is applied first to produce the data for the second function and so on. We omit arguments of the functions representing parameters of procedures.

(3) The application domain knowledge.

It provides a model, linking man defined attributes to the physical properties measured by remote sensing. One important aspect of this sort of models is that only a measure of performance needs to be defined such that the system can modify its strategy during processing (optimisation algorithms, model inversion, parameter estimation) in order to reach a mimimum cost solution.

For systems with a well defined goal, performance functions can be defined that measure the distance from the existing state to the goal state at any point in time. Systematic search leads then to the finding of an optimal the path towards that goal.

4. REASONING

From the previous section, we have defined the domain of knowledge and representation method for the system. In this section, we will be concerned with the order in which rules are selected.

Usually, the reasoning in a knowledge based system is done at two levels.

1. Analysis plan generation first reasons out an appropriate plan to guide the analysis of a given image. The reasoning engine uses characteristics of the image and knowledge about standard image analysis processes to generate the plan.

2. procedure selection and parameter adjustment. The reasoning at this level instantiates the analysis plan into a special subgoal. Procedures are selected and optimal parameter values are determined. If the derivatives of cost functions to parameters or procedure selection are not available then selections are done through trial and error, i.e. the system performs image analysis by applying promising procedures, and evaluates the analysis results for the discovery of a minimum cost solution.The following example describes this situation.

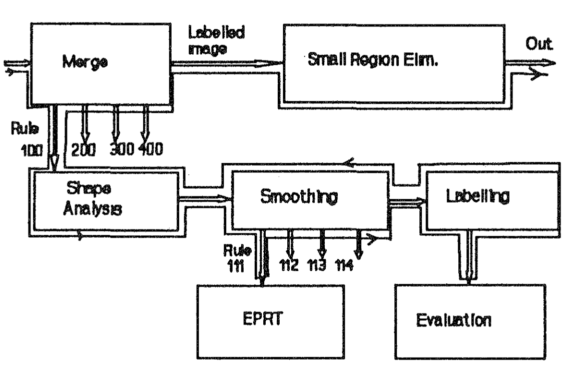

Figure 3. the search path for optimal merging

Firstly, the system produces a process sequence for merging region as following steps:

(1) smoothing the image using an edge preserving filter

(2) assigning labels to connected components

(3) measuring the length of perimeter of a region (the number of boundary elements).

(4) producing evaluate the cost of alternative merging of regions and select the minnimum cost strategy.

After plan generation, some procedures and parameters must be selected. In step (1), the operator of edge-preserving-smoothing is selected. The following example depicts the situation:

RULE 111:
IF noise must be reduced AND edge must be preserved
THEN run EPRT
Where EPRT is the name of the edge preserving operators in SPIDER.

In step (2), although the system does not need to choose the labelling method, a threshold has to be defined. We choose the adaptive threshold option[8] for region labelling. The thresholding proceeds as follows: Given a grid size N, the input image is divided into NxN windows. For each NxN window, the statistics (average and standard deviation) within the windows are calculated. If the standard deviation within the window is smaller than the standard deviation of some background patch, then there is no object within that windows; if the deviation is greater, then we label the object.

RULE 120:
IF $St.Dev._{windows} >= St.Dev._{back.}$
THEN $THRESHOLD_{window} = Ave._{window} - St.Dev._{back.}/2$

In (4) the system measures the length of perimeter of each region P, then measures the common length W between two regions (R1,R2) on which the difference of value is less than a thresholding value $\Theta1$, and the common length B between regions. Rule 130 131 will merge such two adjacent regions iteratively.

RULE 130
IF: 1 the difference over a border[9] is LOW
2 $W/B > \Theta1$
THEN: merge R1,R2

RULE 131
IF: $W/min \{P1,P2\} > \Theta1$
THEN: merge R1,R2

When more than 2 rules can be used , we use specificity ordering. It means that the more conditions a rule has, the higher matching priority it has.

In the system, we apply the depth-first search for forming a processing strategy. We arrange the most promising potential solution for each sort of process as a default path.

It is assumed that the definitions of initial states, procedures and goals are all fixed, thus determining a search space; the question then is how to search the given space efficiently. The techniques for doing so usually require additional information about the properties of the specific problem domain beyond that which is built into the state and procedure definition. Information of this sort is heuristic information. The measure by which the promise of a node is estimated is called an evaluation function. The next node will be selected according to the criterion of minimum cost / maximum benefit.The setting of parameters depends on the evaluation results of previous processing or experience.

In our system, frame-based knowledge representation provides flexiblity and inheritance of common knowledge over a set of subroutines and parameter values.

In execution, firstly, we try to find some key factors affecting the performance of a processing algorithm, then to specify the different values of factors to a specific problem-solving, we can store these promising values into the legal values slots (see section 3). Our representation structure allows us to store these values in advance. On the other hand, we can also store less important factors into default slots.

A user will pay more attention to the specific application problem instead of to the know-how about problem-solving, i.e. he is interested in the final result(state), but not in how to reach this. We provide users with all possible final states for choosing. In the system, we organize those states and some intermediate layer states to form a search space. On each state node there are different direct outputs according to input and state. We give different weights to the state nodes in order to arrive at a final node along a minimum cost path. Tracing of the problem space search provides an explanation facility, prviding answers to questions of the types why(X)? and why_not(Y)?.

5. SUMMARY

In this paper, we present a knowledge based method to solve some problems in image analysis. We embed human understanding and experience about subroutines in image processing packages into the system as a kind of prior knowledge to guide the setting of parameter, and organize the knowledge about techniques of image processing to plan the process sequence. Although the system cannot produce anything new to users, our aim is to develop a more intelligent system as an expert system for image processing. We regard the expert system for image processing as a new flexible software environment for developing image analysis. It facilitates the development of image analysis for users . At the same time, the increasing knowledge of image analysis can be represented in the system to enlarge the systems knowledge base and to expand application areas of image processing.

In this section we would like to present some problems to be solved.

(1). Evaluation of the analysis result. To go to a next step in processing, we need to evaluate the result of the last step. This analysis result will be represented in the system as a kind of heuristic knowledge or analytical cost function. In our method, a minimum cost evaluation function is used.Alternatives for cost function definition are being investigated.

(2). Description of 'visual' information. Our basic strategy is to model the relation between three dimensional scenes and two dimensional images on the basis of physical systems analysis. However, some knowledge, like that of a trained picture interpretor, is difficult to represent. To facilitate the inclusion of this kind of unstructured knowledge we are going to develop a friendly interface for users to assist the description of visual information in terms of geometric, radiometric, dynamic models.

(3). The integration of RS with GIS can be achieved in a natural way by storing likelihoods and prior probabilities in a (new)GIS, let a knowledge based system generate hypotheses from the GIS and evaluate them against evidence derived from RS data. This goal of integration is approached through an overall research plan "model based image analysis" being executed at ITC and the UT in cooperation with members of the Dutch society for pattern recognition and image processing.

REFERENCES

[1] A.Barr and E.Feigenbaum, The Handbook of Artificial Intelligence,Vol. I, Stanford, CA: HeurisTech Press,1981.

[2] SPIDER USER'S MANUAL, Joint System Development Corp., Tokyo,1982.

[3] Katsuhiko Sakaue and Hideyuki Tamura, Automatic Generation of Image processing Programs by Knowledge-based Verification.

[4] Keith Weiskamp and Terry Hengl ,Artificial Intelligence Programming with Turbo Prolog, John Wiley & Sons, Inc. 1988.

[5] X.S.Cheng,Design and Implementation of an Image Understanding System: DADS, Ph.D Thesis, Delft University of Technology, Netherlands,1990.

[6] Mulder N.J. and H.P.Pan, Panorama: An Experimental Image Understanding System, in Proceedings ISPRS VI Symposium, Wuhan, China ,May 1990.

[7] Mulder N.J. and K.Schutte, Model Based Image Analysis: Using Solid Modelling and Ray Tracing, Symposium Remote Sensing & Space, Hat Yai, Thailand, Jan. 1992.

[8] Sharon A. Stansfield, ANGY: A Rule-Based Expert System for Automatic Segmentation of Coronary Vessels From Digital Subtracted Angiograms IEEE Trans. Pattern Anal. & Machine Intell., Vol.PAMI-8,No.2, pp188-199, March 1986.

[9] Martin D. Levine and Ahmed M. Nazif, Dynamic Measurement of Computer Generated Image Segmentation, IEEE Trans. Pattern Anal. & Machine Intell., Vol. PAMI-7, No.2 pp155-164, March 1985.

[10] David M.Mckeown, Rule-Based Interpretation of Aerial Imagery, IEEE Trans. Pattern Anal. & Machine Intell. Vol. PAMI-7, No.5 pp570-585,Sept. 1985.

[11] Takashi Matsuyama, Knowledge-Based Aerial Image Understanding System and Expert Systems for Image Processing, IEEEE Trans. Geoscience and Remote Sensing, Vol. GE-25, No.3,PP305-316,May 1987.

[12] Takashi Matsuyama, Expert System for Image Processing Knowledge-Based Composition of Image Analysis Processes, CVGIP, Vol.48, pp22-49,1989.

[13] A. Nazif and M.Levine, Low-Level Image Segmentation: An Expert System, IEEE Trans. Pattern Anal. & Machine Intell. Vol.PAMI-6, No.5 pp555-577, 1984.

[14] J.J.McKeown, D Meegan and D Sprevak, "An Introduction to Unconstrained Optimisation", Adam Hilger, 1990, ISBN 0-7503-0025-6.