

Using Artificial Intelligence Planning to Automate Science Data Analysis for Large Image Databases

Steve Chien, Forest Fisher, Helen Mortensen

Jet Propulsion Laboratory
California Institute of Technology

Edisanter Lo, Ronald Greeley

Department of Geology
Arizona State University, Tempe, AZ

Abstract

In recent times, improvements in imaging technology have made available an incredible array of information in image format. While powerful and sophisticated image processing software tools are available to prepare and analyze the data, these tools are complex and cumbersome, requiring significant expertise to properly operate. Thus, in order to extract (e.g., mine or analyze) useful information from the data, a user (in our case a scientist) often must possess both significant science and image processing expertise.

This paper describes the use of AI planning techniques to represent scientific, image processing, and software tool knowledge to automate knowledge discovery and data mining (e.g., science data analysis) of large image databases. In particular, we describe two fielded systems. The Multimission VICAR Planner (MVP) which has been deployed for 2 years and is currently supporting science product generation for the Galileo mission. MVP has reduced time to fill certain classes of requests from 4 hours to 15 minutes. The Automated SAR Image Processing system (ASIP) which is currently in use by the Dept. of Geology at ASU supporting aeolian science analysis of synthetic aperture radar images. ASIP reduces the number of manual inputs in science product generation by 10-fold.

Introduction

Recent breakthroughs in imaging technology have led to an explosion of available data in image format. However, these advances in imaging technology have brought with them a commensurate increase in the complexity

⁰This paper describes research conducted by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors would also like to acknowledge the contributions of other past and present members of the MVP team: Todd Turco, Christine Ying, Shouyi Hsiao, Darren Mutz, Alex Gray, Joe Nieten, and Jean Lorre. The authors would also like to acknowledge other contributors to the ASIP project including: Dan Blumberg(ASU), Anita Govindjee, John McHone(ASU), Keld Rasmussen(ASU), and Todd Turco.

of image processing and analysis technology. When analyzing newly available image data to discover patterns or to confirm scientific theories, a complex set of operations is often required. First, before the data can be used it must often be reformatted, cleaned, and many correction steps must be applied. Then, in order to perform the actual data analysis, the user must manage all of the analysis software packages and their requirements on format, required information, etc.

Furthermore, this data analysis process is not a one-shot process. Typically a scientist will set up some sort of analysis, study the results, and then use the results of this analysis to modify the analysis to improve it. This analysis and refinement cycle may occur many times - thus any reduction in the scientist effort or cycle time can dramatically improve scientist productivity.

Unfortunately, this data preparation and analysis process is both knowledge and labor intensive. Consider the task of producing a mosaic of images of the moon from the Galileo mission (corrected for lighting, transmission errors, and camera distortions). Consider also that our end goal is to perform geological analyses - i.e., to study the composition of the surface materials on the moon. One technique used to do this is to construct a ratio image - an image whose values are the ratio of the intensity of the response at two different bandwidths (e.g., the ratio of infra-red response and visible green response). In order to correctly be able to produce this science product for analysis, requires knowledge of a wide range of sources including:

- the particular science discipline of interest (e.g., atmospheric science, planetary geology),
- image processing and the image processing libraries available,
- where and how the images and associated information are stored (e.g., calibration files), and
- the overall image processing environment to know how to link together libraries and pass information from one program to another.

It takes many years of training and experience to acquire the knowledge necessary to perform these analyses. Needless to say, these experts are in high demand.

One factor which exacerbates this shortage of experts is the extreme breadth of knowledge required. Many users might be knowledgeable in one or more of the above areas but not in all the areas. In addition, the status quo requires that users possess considerable knowledge about software infrastructure. Users must know how to specify input parameters (format, type, and options) for each software package that they are using and must often expend considerable effort in translating information from one package to another.

Using automated planning technology to represent and automate many of these data analysis functions (p. 50 (Fayyad96)) enables novice users to utilize the software libraries to mine the data. It also allows users who may be expert in some areas but less knowledgeable in other to use the software tools to mine the data.

The remainder of this article is organized as follows. First, we provide a brief overview of the key elements of AI planning. We then describe two fielded planning systems for science data analysis. We first describe the MVP system - which automates elements of image processing for science data analysis for data from the Galileo mission. We then describe the ASIP system - which automates elements of image processing for science data analysis of synthetic aperture radar (SAR) images.

The principle contributions of this article are twofold. First, we identify software tool reconfiguration as an area where AI planning technology can significantly extend KDD capabilities. Second, we describe two systems demonstrating the viability and impact of AI planning on the KDD process.

Artificial Intelligence Planning Techniques

We have applied and extended techniques from Artificial Intelligence Planning to address the knowledge-based software reconfiguration problem in general, and two applications in science data analysis (e.g., data mining) in specific. In order to describe this work, we first provide a brief overview of the key concepts from planning technology¹.

Planning technology relies on an encoding of possible actions in the domain. In this encoding, one specifies for each action in the domain: *preconditions*, *postconditions*, and *subactivities*. Preconditions are requirements which must be met before the action can be taken. These may be pieces of information which are required to correctly apply a software package (such as the image format, availability of calibration data, etc). Postconditions are things that are made true by the execution of the actions, such as the fact that the data has been photometrically corrected (corrected for the relative location of the lighting source) or that 3-dimensional topography information has been extracted from an image. Substeps are lower level activities which comprise the higher level activity. Given this encoding of actions, a planner is able

¹For Further details on planning the user is referred to (Pemberthy92; Erol94)

to solve individual problems, where each problem is a current state and a set of goals. The planner uses its action models to synthesize a plan (a set of actions) to achieve the goals from the current state.

Planning consists of three main mechanisms: sub-goaling, task decomposition, and conflict analysis. In sub-goaling, a planner ensures that all of the preconditions of actions in the plan are met. This can be done by ensuring that they are true in the initial state or by adding appropriate actions to the plan. In task decomposition, the planner ensures that all high level (abstract) activities are expanded so that the lower level (subactivities) are present in the plan. This ensures that the plan consists of executable activities. Conflict analysis ensures that different portions of the plan do not interfere with each other.

The Multimission VICAR Planner (MVP)

MVP (Chien96) partially automates generation of image processing procedures from user requests and a knowledge-based model of VICAR image processing area using Artificial Intelligence (AI) automated planning techniques. In VICAR image processing, the actions are VICAR image processing programs, the current state is the current state of the image files of interest, and the specification of the desired state corresponds to the user image processing goals.

The VICAR environment (Video Image Communication and Retrieval²) (LaVoie89) supports image processing for: JPL flight projects including VOYAGER, MAGELLAN, and GALILEO, and CASSINI; other space imaging missions such as SIR-C and LANDSAT; and numerous other applications including astronomy, earth resources, land use, biomedicine, and forensics with a total of over 100 users. VICAR allows individual processing steps (programs) to be combined into more complex image processing scripts called procedure definition files (PDFs). The primary purpose of VICAR is to enable PDFs for science analysis of image data from JPL missions.

An Example of MVP Usage

In order to illustrate how MVP assists in VICAR planetary image processing, we now provide a typical example of MVP usage to ground the problem and the inputs and outputs required by MVP. The three images, shown at the left of Figure 1 are of the planet Earth taken during the Galileo Earth 2 flyby in December 1992. However, many corrections and processing steps must be applied before the images can be used. First, errors in the compression and transmission of the data from the Galileo spacecraft to receivers on Earth has resulted in missing and noisy lines in the images. Line fillin and spike removals are therefore desirable. Second, the images

²This name is somewhat misleading as VICAR is used to process considerable non-video image data such as MAGELLAN synthetic aperture radar (SAR) data.

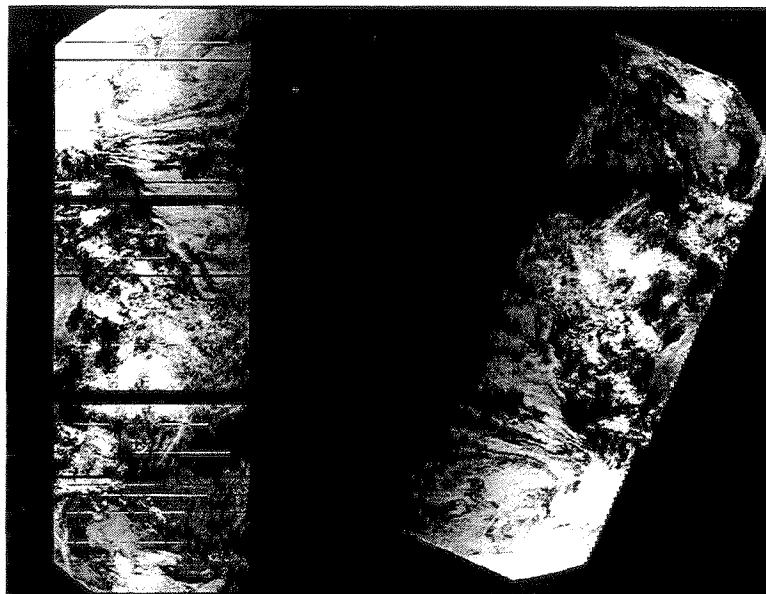


Figure 1: Raw and Processed Image Files

should be map projected to correct for the spatial distortion that occurs when a spherical body is represented on a flat surface. Third, in order to combine the images, we need to compute common points between the images and overlay them appropriately. Fourth, because we are combining multiple images taken with different camera states, the images should be radiometrically corrected before combination.

MVP enables the user to input image processing goals through a graphical user interface with most goals as toggle buttons on the interface. A few options require entering some text, usually function parameters that will be included as literals in the appropriate place in the generated VICAR script. Figure 2 shows the processing goals input to MVP. Using the image processing goals and its

- | | |
|--------------------------------------|---------------------------------|
| radiometric correction | pixel spike removal |
| missing line fillin | uneven bit weight correction |
| no limbs present in images | perform automatic navigation |
| display automatic nav residual error | perform manual navigation |
| display man nav residual error | map project with parameters ... |
| mosaic images | smooth mosaic seams using DN |

Figure 2: Example Problem Goals

knowledge of image processing procedures, MVP constructs a plan of image processing steps to achieve the requested goal. This plan is translated into a VICAR script which, when run, performs the desired image corrections and constructs a mosaicked image of the three input files. The finished result of the image processing task is shown at the right in Figure 1. The three original images now appear as a single mosaicked image, map projected with missing and corrupted lines filled in.

To further continue this example, shown in Figure 3 is a code fragment to perform portions of image navigation

Conceptual Steps	VICAR Code
get initial navigation information	IBISNAV OUT="file_listNAV" PLANET=target_C_10 + PROJECT="GLL" SEDR=@RIMSRC FILENAME="file_listLIST"
construct initial overlap pairs	!! Construct initial overlap pairs MOSPLOT MOSPLOT inp="file_listNAV" n=lines_0_6 ns=samples_0_6 project="GLL" ! mos_overlap is just a holder for the overlap plot. dcl copy printrom.zpl mos_overlap dcl print/rotated mos_overlap
refine initial overlap pairs	!! Refine initial overlap pairs edibis EDIBIS INP="file_listOVER"
find previous tiepoint file (if present)	!! Manmatch mosaic file list !! If there is no existing tiepoint file.... !! Check if a tiepoint file exists !! The following code is in written VMS !! LOCAL STR STRING INIT = "" LET _ONFAIL = "CONTINUE" ! Allow the pdf to continue !! If a file is not found DCL DEASSIGN NAME DCL DEFINE NAME %\$SEARCH("file_list.TP") LOCAL STR STRING TRANSLONG NAME STR LET _ONFAIL = "RETURN" ! Set PDF to return on error
use manmatch program to construct or refine tiepoint file	IF (STR = "") MANMATCH INP=("file_listNAV" "file_listOVER") + OUT="file_list.TP" PROJECT="GLL" SEDR FILENAME="file_listLIST" !! If an old tiepoint file exists... !! The old tpfile is part of input and later overwritten. ELSE MANMATCH INP=("file_listNAV" "file_listOVER" "file_list.TP") + OUT="file_list.TP" PROJECT="GLL" SEDR FILENAME="file_listLIST"
use tiepoints to construct OM matrix	!! OMCOR2 OMCOR2 INP=("file_listNAV" "file_list.TP") PROJECT="GLL" GROUND=@GOOD OMCOR2 INP=("file_listNAV" "file_list.TP") PROJECT="GLL" GROUND=@GOOD

Figure 3: Sample VICAR Code Fragment

³ for a Galileo image ⁴. The higher-level conceptual steps (i.e., plan steps) are shown at the left and the corresponding VICAR code is shown at the right. In this case the overall user goal is to navigate the image. The other subgoals (and steps) are necessary to support this goal.

Thus MVP allows the user to go directly from high

³Image navigation is the process of determining the matrix transformation to map from the 2-dimensional (line, sample) coordinate space of an image to a 3-dimensional coordinate space using information on the relative position of the imaging device (spacecraft position) and a model of the target being imaged (e.g., the planetary body).

⁴This code was generated by MVP.

<p>LHS</p> <p>G₁= mosaicking goal present</p> <p>C₀= null</p> <p>C₂= an initial classification has not yet been made</p>	<p>G_R =</p> <p>1. local correction, 2. navigation 3. registration 4. mosaicking 5. touch-ups</p> <p>C₁ = these subtasks be performed in order 1. 2. 3. 4. 5. protect local correction until mosaicking problem class is mosaicking</p> <p>N = mosaicking</p>	<p>operator GALSOS</p> <pre> :parameters ?infile ?ubwc ?calc :preconditions the project of ?infile must be galileo the data in ?infile must be raw data values reseau are not intact for ?infile the data in ?infile is not raw data values missing lines are not filled in for ?infile ?infile is radiometrically corrected the image format for ?infile is halfword ?infile has blemishes-removed if (UBWC option is selected) then ?infile is uneven bit weight corrected if (CALC option is selected) then ?infile has entropy values calculated </pre>
---	--	---

Figure 4: A Task Reduction Rule from MVP

level image processing goals to an executable image processing program. By insulating the user from many of the details of image processing, productivity is enhanced. The user can consider more directly the processing goals relevant to the end science analysis of the image, rather than being bogged down in the details such as file format, normalizing images, etc.

MVP does not always fully automate this planetary imaging task. In typical usage, the analyst receives a request, determines which goals are required to fill the request, and runs MVP to generate a VICAR script. The analyst then runs this script and then visually inspects the produced image(s) to verify that the script has properly satisfied the request. In most cases, upon inspection, the analyst determines that some parameters need to be modified subjectively or goals reconsidered in context. This process typically continues several iterations until the analyst is satisfied with the image product.

Task Reduction in MVP

MVP represents VICAR processing and science data analysis knowledge in the form of task reduction rules. For example, Figure 4 shows a decomposition rule for the problem class *mosaicking with absolute navigation*. This rule states that if mosaicking is a goal of the problem and an initial problem decomposition has not yet been made, then the initial problem decomposition should be into the subproblems local correction, navigation, registration, mosaicking, and touch-ups and that these subproblems must be solved in that order.

Operator-based Planning in MVP

MVP also uses operator-based planning techniques (Pemberthy92). An operator-based planner uses models of actions in a domain to achieve goals from an initial world state. In the VICAR domain the actions (operators) are image processing steps, initial state the initial image file state, and the goals the processing request.

In operator-based planning, an action is represented in terms of its preconditions (required to be true before an action can be executed), and its effects (true after an action is executed). For example, the GALSOS program to radiometrically correct Galileo image files is represented as shown below.

When constructing a plan to achieve a goal G₁, a planner will consider those actions which have G₁ as an effect (thus considering GALSOS to achieve a radiometric correction goal). In order to use GALSOS, MVP must also ensure that the preconditions of the operator are met, in a process called *subgoaling*. MVP must also ensure that operators in the plan do not undo conditions required by other parts of the plan - this is performed in a process called *conflict analysis*⁵.

One novel aspect of the VICAR domain is that considerable search in planning is not at the *program selection level* (which corresponds to operator selection in the planning process) but rather at the *program option selection level* (which corresponds to selecting the appropriate operator effect after the operator has been selected). In order to efficiently handle this type of search, we have integrated a constraint reasoning mechanism which allows the planner to reason about compatible and incompatible program option settings in a least-commitment fashion (see (Chien96) for details).

An Example of Subgoaling in VICAR Image Processing To illustrate how the operator-based planning process performs subgoaling, consider the subgoal graph illustrated in Figure 5.⁶ In this case the user has selected the goal that the images be navigated using manual methods and that the archival navigation information for the image should be updated. The decomposition planner has access to the knowledge that in order to navigate the image, the operational goal is to construct an OM matrix which defines the transformation from (line, sample) in the image to some known frame of reference (usually the position relative to the target planet center). The planner knows that in order to compute this matrix it must have a tiepoint file, the project of the image, and the image files formatted into a mosaic file list. In order to produce a tiepoint file for the goal specification of manual navigation, the planner uses the MANMATCH program. The MANMATCH program in turn requires a refined overlap pairs file, the project of the images, the initial predict information, and again a mosaic file list.

⁵Because subgoaling and conflict analysis in operator-based planning are not unique to MVP, we have only briefly sketched their key elements. For a more detailed treatment of operator-based planning algorithms the reader is referred to (Pemberthy92).

⁶The VICAR code previously shown in Figure 3 is taken from this example.

The refined overlap pairs file can be constructed using the EDIBIS program, but this requires a crude overlap pairs file based on an initial predict source. This crude overlap pairs file in turn requires the default navigation method, and the latitude and longitude of sample image files. The rest of the graph is generated similarly. This subgoal graph is generated in response to the particular combination of user goals and the state of the selected image files.

Example of Resolution of Goal Conflicts in VICAR Image Processing To illustrate how the operator-based planning process resolves interactions between steps, consider the (simplified) image processing operators shown in Figure 6. The relevant operators to achieve the goals of missing line fillin, spike removal, and radiometric correction for Voyager and Galileo images are shown below. When constructing a plan to achieve these goals, depending on the project of the image file (e.g., either Voyager or Galileo), MVP determines the correct program to use because the preconditions enforce the correct program selection.

However, determining the correct ordering of actions can sometimes be complex. In this case, the correct order to achieve the goals of line fillin, spike removal, and radiometric correction is dependent upon the project of the file. In the case of Voyager files, ADESPIKE (spike removal) requires raw pixel values and FICOR77 (radiometric correction) changes pixel values to correct for camera response function – thus FICOR77 removes a necessary condition for ADESPIKE (raw pixel values). This interaction can be avoided by enforcing that ADESPIKE occurs before FICOR77. Additionally, VGR-FILLIN requires binary EDR header on the image file, and ADESPIKE removes the binary EDR header, thus ADESPIKE removes a necessary condition for VGR-FILLIN. This interaction can be avoided by requiring VGRFILLIN to be executed before ADESPIKE. Thus in the VOYAGER example the only legal execution order is VGRFILLIN, ADESPIKE, FICOR77.

In the Galileo case, GALSOS undoes missing line fillin (the goal achieved by the GLLFILLIN operator). Thus in order to avoid undoing this processing, GLL-FILLIN must be applied after GALSOS. Additionally, GALSOS requires raw pixel values, and ADESPIKE alters the pixel values, so ADESPIKE removes a necessary condition for GALSOS. This interaction can be avoided by requiring that GALSOS occurs before ADESPIKE. This determination of correct programs to use and correct execution order is shown in Figure 7.

This simple example illustrates some of the interactions and context-sensitivity of the VICAR image processing application. All of these interactions and context sensitive requirements are derived and accounted for automatically by MVP using the operator specification, thereby allowing plan construction despite the presence of complex interactions and conditions.

Impact of the MVP system

User reports indicate that MVP reduces effort to generate an initial PDF for an expert analyst from 1/2 a day to 15 minutes and reduces the effort for a novice analyst from several days to 1 hour. This represents over an order of magnitude in speedup. The analysts also judged that the quality of the PDFs produced using MVP are comparable to the quality of completely manually derived PDFs.

The Automated SAR Image Processing (ASIP) System

ASIP automates synthetic aperture radar (SAR) image processing based on user request and a knowledge-base model of SAR image processing using AI automated planning techniques. SAR operates simultaneously in multipolarizations and multifrequencies to produce different images consisting of radar backscatter coefficients (s0) through different polarizations at different frequencies. ASIP enables construction of an aerodynamic roughness image/map (z0 map) from this raw data – thus enabling studies of Aeolian processes.

Studies of Aeolian Processes

The aerodynamic roughness length (z0) is the height above a surface at which a wind profile assumes zero velocity. z0 is an important parameter in studies of atmospheric circulation and aeolian sediment transport (in laymans terms: wind patterns, wind erosion patterns, and sand/soil drift caused by wind) (Greeley87a; Greeley87b; Greeley91). Estimating z0 with radar is very beneficial because then large areas can be mapped quickly to study aeolian processes, as opposed to the slow painstaking process of manually taking field measurements(Blumberg95). The final science product is a VICAR image called a z0 map that the scientists use to study the aeolian processes.

Planning to Generate Aerodynamic Roughness Maps

ASIP is an end-to-end image processing system automating data abstraction, decompression, and (radar) image processing sub-systems, and intelligently integrates a number of SAR and z0 image processing sub-systems. Using a knowledge base of SAR processing actions and a general-purpose planning engine, ASIP reasons about the parameter and sub-system constraints and requirements. In this fashion ASIP extracts needed parameters from image format and header files as appropriate, relieving the user of having to know about these aspects of the problem. These parameters, in conjunction with the knowledge-base of SAR processing steps, and a minimal set of required user inputs (entered through a single graphical user interface (GUI)), are then used to create the processing plan. ASIP represents a number of processing constraints. For example, ASIP represents the fact that only some subset of all

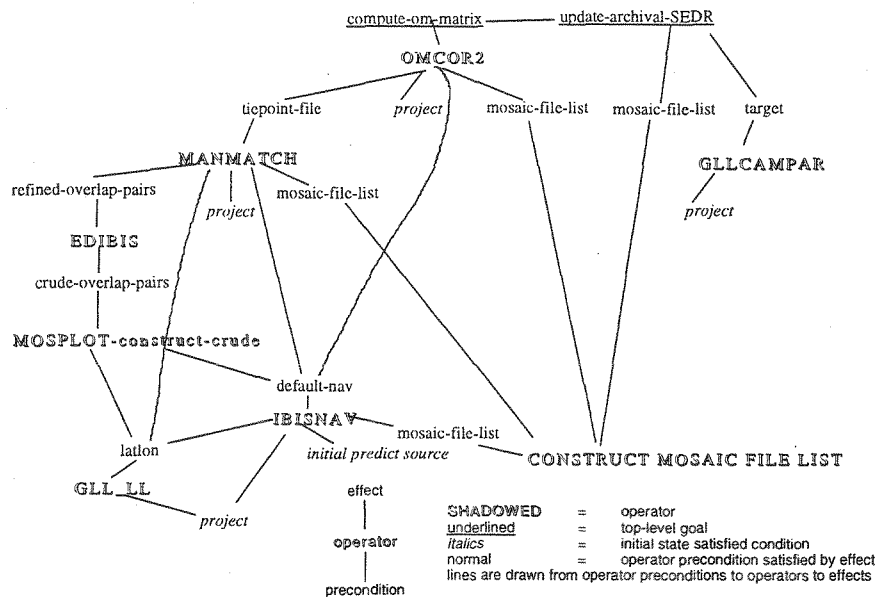


Figure 5: Subgoal Graph for Manual Relative Navigation of Galileo Image Files

Operator	VGRFILLIN	GLLFILLIN	ADESPIKE	FICOR77	GALSOS
Preconditions	VGR image EDR present	GLL image	(GLL image) or ((VGR image) and (raw values))	VGR image	GLL image raw pixel values
Effects	missing lines filled in.....		spike removal not raw values	radiometric corr. blemish removal not raw values	radiometric corr. reed-solomon overflow corr. saturated pixel corr. not missing line fillin

Figure 6: Simplified Operator Definitions

Goal	VICAR Program		Execution Order	
	Voyager	Galileo	Voyager	Galileo
fillin missing lines	VGRFILLIN	GLLFILLIN	VGRFILLIN	GALSOS
remove spikes	ADESPIKE	ADESPIKE	ADESPIKE	GLLFILLIN
radiometric corr.	FICOR77	GALSOS	FICOR77	ADESPIKE

Figure 7: VICAR Programs and Execution Order to Resolve Conflicts for Voyager and Galileo Data

possible combinations of polarizations are legal (as dependent on the input data). ASIP also represents image processing knowledge about how to use polarization and frequency band information to compute parameters used for later processing of backscatter to aerodynamic roughness length conversion - thus freeing the user from having to understand these processes.

Figure 8 shows an aerodynamic roughness length map of a site near Death Valley, California generated using the ASIP system (the map uses the L band (24 cm) SAR with HV polarization). Each of the greyscale bands indicated signifies a different approximate aerodynamic roughness length. This map is then used to study aeolian processes at the Death Valley site.

Since the ASIP system has been fielded, it has proven to be very useful in the use of generating aerodynamic roughness maps with three major benefits. First, ASIP has enabled a 10 fold reduction in the number of manual inputs required to produce an aerodynamic roughness map. Second, ASIP has enabled a 30% reduction in CPU processing time to produce such a map. Third, and most significantly ASIP has enabled scientists to process their own data (previously programming staff were required). By enabling scientists to directly manipulate that data and reducing processing overhead and turnaround, science is directly enhanced.

Conclusions

This paper has described knowledge-based reconfiguration of data analysis software using AI planning techniques. This represents an important area where AI planning can significantly enhance KDD processes. As evidence of this potential, we described two fielded planning systems that enhance KDD: the MVP system, which automates image processing to support Galileo image data science analysis; and the ASIP system which automates production of aerodynamic roughness maps to support geological science analysis.

References

- D. Blumberg and R. Greeley, "Field Studies of Aerodynamic Roughness Length." *Journal of Arid Environments* (1993) 25:39-48.
- S. A. Chien and H. B. Mortensen, "Automating Image Processing for Scientific Data Analysis of a Large Image Database," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (8): pp. 854-859, August 1996.
- K. Erol, J. Hendler, and D. Nau, "UMCP: A Sound and Complete Procedure for Hierarchical Task Network Planning," *Proceedings of the 2nd International Conference on AI Planning Systems*, Chicago, IL, June 1994, pp. 249-254.
- U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Magazine*, Vol 17 No. 3, Fall 1996, pp. 37-54.
- F. Fisher, E. Lo, S. Chien, R. Greeley, "Automated SAR Processing for Science Analysis using Artificial Intelligence Planning Techniques," Submitted to the 1997 International Conference on Image Processing, Santa Barbara, CA, November 1997.
- Y. Iwasaki and P. Friedland, "The Concept and Implementation of Skeletal Plans," *Journal of Automated Reasoning* 1, 1 (1985), pp. 161-208.
- R. Greeley and J.D. Iversen, "Measurements of Wind Friction Speeds over Lava Surfaces and Assessment of Sediment Transport." *G.R.L.* 14 (1987):925-928.
- R. Greeley, P.R. Christensen, and J.F. McHone. "Radar Characteristics of Small Craters: Implications for Venus." *EMP* 37 (1987):89-111.
- R. Greeley, L. Gaddis, A. Dobrovolskis, J. Iversen, K. Rasmussen, S. Saunders, J. vanZyl, S. Wall, H. Zebker, and B. White. "Assessment of Aerodynamic Roughness Via Airborne Radar Observations." 1991, *Acta Mechanica Suppl.* 2, 77-88.
- A. Lansky, M. Friedman, L. Getoor, S. Schmidler, and N. Short Jr., "The Collage/Khoros Link: Planning for Image Processing Tasks," *Proceedings of the 1995 AAAI Spring Symposium on Integrated Planning Applications*, March 1995, pp. 67-76.
- S. LaVoie, D. Alexander, C. Avis, H. Mortensen, C. Stanley, and L. Wainio, *VICAR User's Guide, Version 2*, JPL Internal Document D-4186, Jet Propulsion Lab., California Institute of Tech., Pasadena, CA, 1989.
- J. S. Pemberthy and D. S. Weld, "UCPOP: A Sound Complete, Partial Order Planner for ADL," *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, Oct 1992.
- M. Stefik, "Planning with Constraints (MOLGEN: Part 1)," *Artificial Intelligence* 16,2(1981), pp. 111-140.



Figure 8: Aerodynamic Roughness Length Map Produced Using ASIP