# RESEARCH ON COMPUTER-ASSISTED GENERATION OF IMAGE PROCESSING ALGORITHMS

Donald G. Bailey

Visiting Assistant Research Engineer,
Electrical and Computer Engineering Department,
University of California at Santa Barbara, CA 93106, U.S.A.

## ABSTRACT

To develop an algorithm for an image processing task a person normally should have both extensive experience in image processing and detailed knowledge of the task. Frequently those who need such algorithms have the detailed problem knowledge but not the image processing experience. We are investigating the extent to which the necessary experience can be incorporated into a system for generating image processing algorithms.

Such a system requires at least three types of knowledge: algorithm knowledge, operation knowledge, and selection rules for algorithms and operations.

We represent the algorithm knowledge as a series of candidate algorithm architectures where each architecture is applicable to a range of tasks. The architectures consist of sequences of pseudo-commands specifying the main property required at each step of the algorithm. Operation knowledge includes 1) the primary property, preconditions, and potential side effects of the operation, 2) how the operation may be invoked and what parameters are required, and 3) how the results of the operation should be evaluated. The selection rules determine how the above two types of knowledge are combined to give a useful algorithm.

## INTRODUCTION

Digital image processing consists of using a computer to apply a sequence of mathematical operations to an image of a scene or object in order to achieve some other desired result. The sequence of operations is known as an algorithm and the desired result may be an improved image (enhanced or restored), a measurement (of size, position, intensity, or object orientation), or a classification (as to whether the object is acceptable or not). There are three main requirements to successfully develop an algorithm for a particular image processing task.

The first is an interactive system for image processing which has available a wide range of operations. This is necessary since algorithm development is largely a heuristic process [Rosenfeld, 1984]. There is little or no underlying theory that may be used to determine exactly when a particular operation should be used. In practice an algorithm is frequently developed by trying out an operation. If the result of the operation is not satisfactory, another is tried in its place. This procedure is repeated until the desired result is obtained.

The second requirement is that the person developing the algorithm have an extensive background and experience in image processing, particularly in the type of application for which the new algorithm is being developed. The trial and error approach described in the previous paragraph is simplified and made practical when performed by an appropriate specialist. The specialist must have detailed knowledge of the properties of each of the available operations, experience in determining exactly which operations would be beneficial in a given situation, and ability to choose the operation best suited to the task at hand.

Finally, it essential that the specialist have a detailed knowledge of the task to which the image processing is being applied [Vogt, 1986]. This knowledge is important since it increases the probability that the right measurements will be made and that the algorithm will be robust. Without such "problem knowledge", invalid assumptions may be made about the nature of the task. There is always a danger that a resultant algorithm will work well for the specific cases that were used to develop it, but not be sufficiently general or robust to be of practical use.

With the advent of faster and cheaper computing power, image processing and related techniques are becoming more widely used in the scientific and industrial communities. People with an extensive knowledge of the application, but with little or no background in image processing, are often assigned to develop algorithms. There are two typical methods of solving this problem: to train potential users in image processing, or to train an image processing specialist in the area of application.

The users of image processing are frequently technicians, or other similarly trained members of a technical staff. Because of the heuristic nature of algorithm development, a long training period is required for a person to become experienced and proficient. A realistic estimate of the training time for a typical trainee is one full year. Training is therefore an expensive option and is practical only if there are enough applications to warrant an in-house expert.

The alternative is to consult or contract an image processing specialist to develop the algorithm. Usually this will involve at least a short training period (typically two or three weeks) for familiarization with the details of the task, before the specialist can begin. The disadvantage with this method is that there is a shortage of specialists with appropriate qualifications. When specialists are available, they usually command high salaries.

The ideal solution is to have a system which can automatically generate the required image processing algorithm from a specification of the task to be performed. Once the algorithm has been developed, the system should also suggest a preliminary design of the image processing system that would perform the required image processing task.

## RESEARCH OBJECTIVES

There are a number of stages that require research before a practical system for automatic algorithm generation can be developed. The first stage is to shift the burden of algorithm development from an image processing specialist to an intelligent computer system. Subsequent stages include eliminating the need for the user to evaluate the intermediate results of selected operations, determining how to specify the image processing task to the system, and designing the system so that it can recommend possible implementations of the algorithms it produces.

We are currently performing the research required for the first stage. The aim is to determine the extent to which an image processing system can usefully incorporate the knowledge of an image processing specialist.

## KNOWLEDGE REQUIRED

Such a system must incorporate at least three kinds of knowledge: algorithm knowledge, operation knowledge, and rules for selecting algorithm structures and operations. Problem knowledge is provided directly by the user.

### ALGORITHM KNOWLEDGE:

Knowledge of the basic structure of image processing algorithms is required to guide the selection of appropriate operations. A wide range of measurement and inspection algorithms may be broken down into the following broad steps [Bailey, 1985]:

1) Designate the type of result required. The type of result required determines the kind of information that needs to be extracted from the image and thus governs the formulation of the overall strategy. Broad areas of application include: making quantitative measurements (size, length, area, position, intensity), checking for integrity or completeness (parts, features, burrs, cracks, etc), inspecting cosmetic or surface properties (stains, colors, scratches, etc), and sorting.

2) Obtain a representative image. During the initial process of algorithm development, the image should be as simple as possible while still being representative. As the algorithm is refined, a wider variety of images can be used to test the algorithm and determine its limitations. The most important aspect of the image capture step is the use of appropriate lighting.

3) Preprocess the image. Preprocessing is often used to enhance the information required from the image while suppressing information that is irrelevant. Preprocessing can frequently ensure that the more complicated operations of later stages are successful. Which of the following preprocessing operations are necessary depends strongly on the broad strategy selected at the outset, and on the images that are being preprocessed. Typical preprocessing operations are used for:
   1) Compensating for deficiencies in the image capture process. Due to various physical constraints, the image captured may not be ideal.
   2) Normalizing the image. This step permits making accurate comparisons between images, and includes intensity normalization and positional normalization.
   3) Filtering. Local filters may be used to smooth noise, or to enhance or detect edges or other features.
   4) Thresholding. This step classifies pixels according to their intensity and usually results in a binary image.
   5) Segmenting. The image is divided into meaningful parts or regions with common properties.

4) Extract the required information. The information extraction stage transforms the data from a series of fixed intensities to a more descriptive symbolic form. The image data consists of a large number of pixel values, each carrying very little information. The data are converted to a form where there are few numbers, but each number carries significant information about the object or scene being imaged. The types of operations used for information extraction include:
   1) Feature extraction. Features of the object are extracted that contain information useful for discriminating against other objects that may be encountered. This information is often a small set of numbers and may represent size, position, shape, etc.
   2) Measurement. In some applications, the information required is a measurement, while in others the measurements made are used for classification purposes. Quantities that may be measured include length, area, and intensity. When absolute measurements are required, the system should be calibrated by executing the algorithm on a similar object of known dimensions. These measurements may then be used to calibrate the result by scaling.
   3) Pattern identification and classification. At this stage, the object being imaged is classified into one of two (or more) categories based on the information extracted. This step is used for defect rejection, sorting, etc.

   These information extraction steps are very dependent on the application. Appropriate features must be selected for the discrimination indicated during the initial stages of the algorithm development process.

5) Postprocess the image or results extracted. In some circumstances, postprocessing may be required to convert the extracted information into the desired form. This may require: displaying resultant images, manipulating data for robot control, presenting special information, and recording a tally of production or defect rates.

6) Refine the algorithm. Since algorithms are initially developed using a single image, or a small range of representative images, they require testing. For this a large number of images is necessary, especially images that are marginal as far as any classifications are concerned. Various complicating factors and special cases may be introduced and the algorithm refined to accommodate them. In applications where speed is critical, the processing bottlenecks may be identified and alternative operations selected.

We have chosen to represent the above knowledge as a set of candidate algorithm architectures. This set contains the basic algorithmic structures for a range of tasks, with each structure being applicable to a small class of related tasks. These structures or architectures are derived by analyzing a range of image processing algorithms. The architecture derived from a particular algorithm may then be used to generate algorithms for similar tasks.

### OPERATION KNOWLEDGE:

The second type of knowledge is of the operations available in the image processing system. Note that the operation knowledge is represented independently of the algorithm knowledge. This makes it easy to add new operations and change existing operations without having to make changes to the algorithm knowledge base. The knowledge about each operation may be split into the following five categories:

1) Main property. The primary property of each operation is used to select a particular operation from those available. Each operation will fit into one of the algorithm steps described previously.

2) Preconditions. Some operations require that certain conditions be satisfied in the image, or other data structures before they are applicable. For example, suppose an operation (such as measuring the area by counting pixels) requires that the image be binary. If the image is not already binary, then an operation is selected that will make it binary - for example, thresholding.

3) Side-effects. Some operations have side effects or secondary properties. In some situations, these side effects may be desirable, while in others, they may be used to select

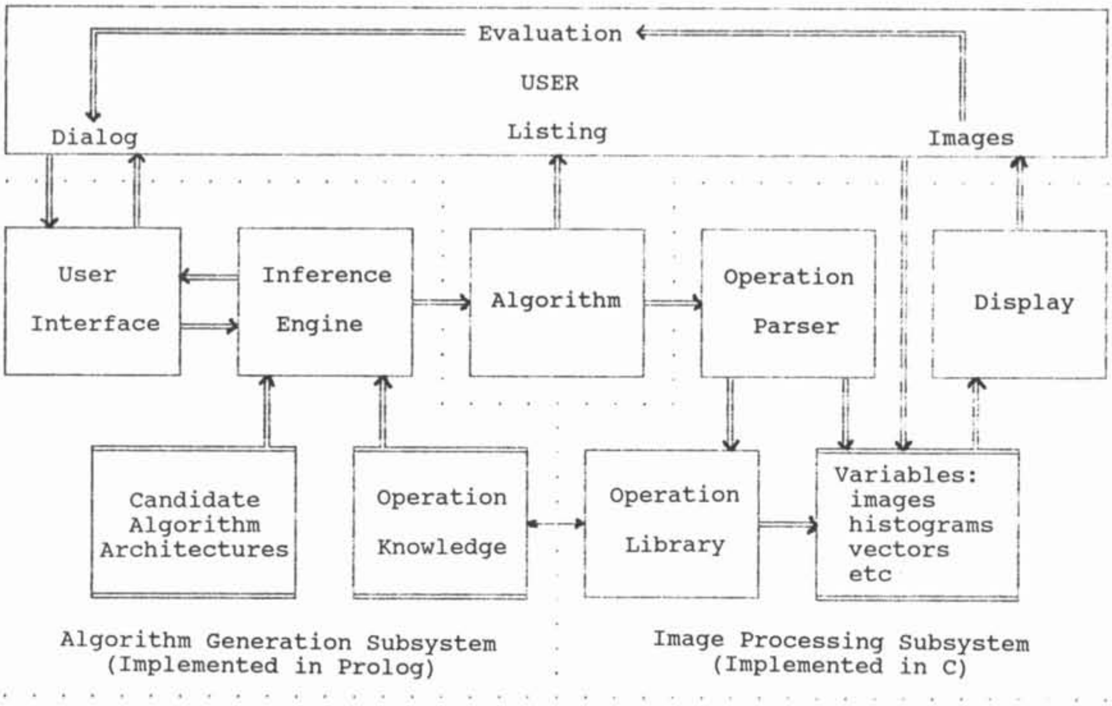


Figure 1: Block diagram of the prototype system for computer assisted algorithm generation.

from several possible operations with the same primary property. An example of the latter is selecting a filter for reducing noise. A linear filter has the side effect of blurring edge information in the image, whereas a median filter does not. In situations where edge information is critical, a median filter would be selected in preference to the linear filter.

4) Command interface. Each operation is invoked by sending a command to an image processing system. This information is used to verify correct operand types (for example, image, integer, etc) and to specify what command parameters are required (for example a threshold level in a thresholding operation).

5) Evaluation criteria. Most operations require that the result be evaluated to determine whether or not the operation was successful. For each property, including both primary and secondary properties, the aspects of the resultant image or data structure that must be evaluated need to be determined. At this stage of the project, these evaluation criteria need to be translated into questions to be asked of the user.

## SELECTION RULES:

In addition to the above areas, knowledge is required about how to select an appropriate algorithm architecture, how to capture representative images, how to select specific operations for each step in the algorithm, and how to determine any auxiliary parameters required by the operations (for example a threshold level).

In contrast to the algorithm and operation knowledge, which are sets of facts, this latter knowledge consists of rules describing how to make the selections or determinations.

## IMPLEMENTATION

A schematic of the system currently being developed is shown in figure 1. It consists of two main subsystems: the algorithm generation subsystem and the image processing subsystem. These subsystems are kept loosely coupled to enable the same algorithm generation subsystem to be used with different image processing systems. During this stage of the project, the only direct communication between the two subsystems involve commands from the algorithm generation subsystem to the image processing subsystem; all results from the image processing subsystem return via the user.

A command-based image processing system is being written in the C language. This will have a command structure very similar to that of the VAX Image Processing System (VIPS) developed at the University of Canterbury, New Zealand [Bailey and Hodgson, 1988].

The algorithm generation subsystem works as follows: by interacting with the user through a question-and-answer type of dialogue, the type of image processing task to be performed is determined. From the information provided by the user, recommendations are made on how to capture one or more representative images.

Once a representative image has been obtained, a candidate algorithm architecture can be selected according to the type of task to be performed. Associated with each architecture in the knowledge base is a set of conditions or rules describing the type of tasks for which it is applicable. By matching architecture conditions with the task description provided by the user, applicable architectures can be selected. Any additional information required to make this selection is obtained by asking questions of the user.

Each candidate architecture contains a directed graph which represents the basic structure of the algorithm. Each node in the graph specifies the main property of the operation required for that step in the algorithm. Following is an example architecture derived from the algorithm for detecting blemishes on the surface of kiwifruit [Bailey, 1985].

```
architecture("BLEMISH",
  "Surface blemish detection",
  [n(1,[0],[2],        f(image,segmentation),
                       "Locate object"),
   n(2,[1],[3],        f(image,normalization),
                       "Correct for intensity variation"),
   n(3,[2],[4],        f(image,modelling),
                       "Unblemished object"),
   n(4,[2,3],[5],      f(image,comparison),
                       "Compare object with model"),
   n(5,[4],[6],        f(image,segmentation),
                       "Locate blemishes"),
   n(6,[5],[0],        f(image,measurement),
                       "Measure blemish")]).
```

Each step, or node, of the selected candidate architecture is filled with specific operations. For a particular node, several operations with appropriate properties may be selected. These are then ranked according to how well the preconditions match the current state of the image, and the most appropriate operation is selected. The selection of a particular operation may result in a search for other operations to satisfy the remaining preconditions, and to determine other parameters.

Each operation that can be performed by the image processing subsystem is represented by a frame. Following is an example of the THRESHOLD operation:

```
operation("THRESHOLD",
   f(image,segmentation),
   "THRESHOLD",
   [p("<image>",    pimage,   main_io,   none),
    p("<level>",    pint,     req_in,    ar("FIX_THRESH")),
    pend],
   [r(separate,      f(object,intensity),
                     f(background,intensity))],
   [f(image,binary)],
   "Have we segmented the image correctly?").
```

The fields or slots in this frame are: the title of the operation (THRESHOLD), the main property (image segmentation), the command to be sent to the image processing system, and the parameters required by the command. These parameters are:

<image>: This is the main parameter (the image to be thresholded), the one represented by the links in the architecture.

<level>: This refers to the threshold level, an integer required by the command. Its value may be determined by invoking the FIX_THRESH secondary architecture.

The next field in the operation refers to the preconditions. For the THRESHOLD command to be successful, the object intensity and background intensity must be separate. The side-effects are that the THRESHOLD operation gives a binary image. The last slot in the frame contains the evaluation criteria.

The most suitable operation is selected from those that are applicable, and a command is sent to the image processing subsystem to perform that operation on the sample image. The results are evaluated by the user in response to specific questions from the system. This evaluation determines if the operation has moved the state of the algorithm to the desired goal and determines if the side effects of the operation are critical. To correct for the side effects, if necessary, other operations may be used, either before or after the operation in question.

Some operations may cause secondary architectures to be invoked. For example, the thresholding operation described above requires a threshold level as a parameter. There are a number of different threshold selection methods, each of which is applicable in different circumstances. In this case, the purpose of the secondary architecture FIX_THRESH is to use rules on threshold selection to determine which method is most appropriate in any given application, and then to generate any image processing commands necessary to calculate the threshold level.

The operation selection process described in the previous paragraphs is repeated for each node in the candidate algorithm architecture. If a suitable operation is not found for a particular node, the algorithm generation subsystem backtracks to the previous node to consider alternative operations. Once the algorithm is completed, that is an operation is found for each of the nodes in the candidate algorithm architecture, the algorithm is provided to the user.

## SUMMARY AND CONCLUSIONS

This paper describes one possible approach for making it easier for those that are not specialists in image processing to develop algorithms for tasks in which they are interested. This is achieved by incorporating some of the necessary image processing expertise into the image processing system used for algorithm development.

We are currently developing a prototype system to be used to test the ideas presented here. At present, we have completed the framework shown in figure 1, and are investigating ways of practically using the information in the knowledge bases. Our initial results appear promising, however, there is currently very little knowledge contained in the system. One of the next tasks is to expand the knowledge bases to include more algorithm architectures and operations.

The system described also shows promise as a teaching tool. If a suitable user interface is provided, the users are able to learn the image processing techniques employed by the system as they develop algorithms for their tasks.

## ACKNOWLEDGEMENTS

Dr Bailey was supported in part by a John R. Templin Travelling Scholarship. The research described in this paper was funded in part by a seed grant from the Non-linear Science Group of the University of California.

## REFERENCES

BAILEY D.G. (1985), 'Hardware and software developments for applied digital image processing', Ph.D. thesis, Electrical and Electronic Engineering Department, University of Canterbury, New Zealand.

BAILEY D.G., and HODGSON R.M. (1988), 'VIPS - a digital image processing algorithm development environment', Image and Vision Computing, accepted for publication.

ROSENFELD A. (1984), 'Image analysis: problems, prospects and progress', Pattern Recognition, vol 17, pp 3-12.

VOGT R.C. (1986), 'Formalized approaches to image algorithm development using mathematical morphology', Vision '86 Conference Proceedings, pp 5/17 to 5/37.