

DIGITAL IMAGE PROCESSING SOFTWARE

**DIGITAL IMAGE PROCESSING SOFTWARE
DESIGN FOR ELECTRON MICROSCOPE
IMAGE ANALYSIS**

BY

MARIA ADAMCZYK, M. OF ECON.

A Project

**Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements**

**for the Degree
Master of Science**

McMaster University December 1988

(c) Copyright by name, Maria Adamczyk

MASTER OF SCIENCE (1988)
(Computation)

McMaster University
Hamilton, Ontario

TITLE: DIGITAL IMAGE PROCESSING SOFTWARE DESIGN FOR
ELECTRON IMAGE ANALYSIS.

AUTHOR: MARIA ADAMCZYK (M. of Econ., Business Academy,
Katowice)

SUPERVISORS : DR S. JARZABEK
DR B. ROBERTSON

NUMBER OF PAGES : 75

ABSTRACT

The central idea behind digital image processing is quite simple. The digital image is fed into a computer one pixel at a time. The computer is programmed to insert these data into an equation, or series of equations, and then store the results of the computation for each pixel. These results form a new digital image that may be displayed or recorded in pictorial format (specific for the particular image processing system in use) or may itself be further manipulated by additional programs. The possible forms of digital image manipulation are literally infinite. The purpose of this project is to implement some image processing techniques to facilitate the image analysis research conducted in the Institute for Materials Research at McMaster University.

ACKNOWLEDGEMENTS

I would like to express my great appreciation and sincere thanks to Dr B. Robertson for his patience, assistance, and enlightening guidance during my preparation of this project. I would also like to thank very much Dr S. Jarzabek for his assistance, advice and his encouragement towards finishing this project. I would like to present my appreciation to Dr Z. S. Basinski for his help and advice during my work on this project.

At the end I would also like to thank all the members of the department of Computer Science for the knowledge and the experience that I have gained so far during my period of study in this unit.

TABLE OF CONTENTS	PAGE
CHAPTER 1 - INTRODUCTION	- 1
CHAPTER 2 - PROCESSING BACKGROUND AND INITIAL RESOURCES	- 4
2.1 Electron microscope applications in the Institute for Materials Research at McMaster University.	- 4
2.2 Preliminary hardware and software resources.	- 7
CHAPTER 3 - CHARACTERISTICS OF IMAGE PROCESSING DESIGN	- 11
3.1 Methodology.	- 11
3.2 User interface.	- 13
3.2.1 Input / Output interface.	- 13
3.2.1.1 PIB-file processing.	- 13
3.2.1.2 LOGITECH mouse.	- 14
3.2.2 General processing interface.	- 15

3.3 Aspects of modularity in the project.	- 18
3.4 System organization.	- 23
3.5 Error handling.	- 28
CHAPTER 4 - IMAGE PROCESSING ALGORITHMS IMPLEMENTED IN THE PROJECT	- 30
4.1 Menu hierarchy.	- 30
4.2 Multiple files processing.	- 32
4.3 Single file processing.	- 35
4.4 Bit slicing operations.	- 38
4.5 Intensity profiles.	- 42
4.6 Filtering.	- 45
4.7 Binary images.	- 48
4.8 Double images.	- 52
CHAPTER 5 - FUTURE ENHANCEMENTS AND CONCLUSION	- 55
5.1 Future possibilities.	- 55
5.3 Conclusion.	- 57
Appendix A. User's manual.	- 62

CHAPTER 1 - INTRODUCTION

Pictorial information in all its varieties constitutes the most important source of our sensor data as well as the most general means of communication between people. In image processing the real images from the outside world (real scenes photographs, microscopic images, satellite images, fingerprints, and many others) are provided to the computer and processed by it.

The results of processing can be of different types: other pictures (e.g. enhanced, noise reduced, etc.), quantitative descriptions of the picture contents (e.g. number of objects, areas of cells, positions of some features, etc.), recognition decisions (e.g. name of an alphanumeric character, fingerprint classification code, etc.). [2]

Digital image processing encompasses a wide variety of techniques and mathematical tools. They have all been developed for use in one or the other of the two basic activities that constitute digital image processing, image enhancement, and image analysis.

Image enhancement accepts a digital image as input and produces an enhanced image as output; in this

context enhanced means better in some respect. This includes increasing or decreasing the contrast, removing or introducing geometric distortion, edge sharpening or smoothing, or altering the image in some respect which facilitates the interpretation of its information content (i.e. visual properties such as brightness, colour or reflectance).

Image analysis on the other hand, accepts digital imagery as input but as output produces a report of some type. [2] It might be a simple count of the number of separated features or a complex report that assigns pixel to a specific region, with all region areas calculated.

Following the great number of groups interested in picture processing a vast variety of such systems was built [1]. They are often constructed as tools facilitating basic research in digital picture processing or as practical devices for some more or less specific applications. Most image processing systems are heavily oriented towards image processing in a narrow sense i.e.: transforming one image into other images. This certainly is a necessary facility in all image processing systems and was a basic approach used in the presented design.

Because digital image processing encompasses such a variety of techniques, each application involves a sequence of processing steps. The present project gives a number of different functions organized as separate modules which can

be arranged in sequences of manipulation schemes to achieve the desired results in particular image processing case.

The environment and the design goals are described in chapter three, while the detail description of all the image processing algorithms implemented in the system is included in chapter four. In Appendix A the User's Manual is enclosed with detailed specifications for the system interface.

CHAPTER 2 - PROCESSING BACKGROUND AND INITIAL RESOURCES

2.1 Electron microscope applications at the Institute for Materials Research at McMaster University.

The basic aim of all forms of microscopy is to obtain as much information as possible from a sample of matter (the specimen) by observing its interaction with different forms of radiation (electromagnetic or particulate) in different modes (transmission or reflection, flood-beam or scanning-beam). Until quite recently the microscopist has had to be content with a qualitative, or at best a semi-quantitative, description of the specimen based on visual information.

Attempts to obtain numerical data from typical two-dimensional images of microscopical sections are not new. In the past, areas of objects on positive prints have been measured with planimeters, and optical densities of objects on photographic negatives have been measured with microdensitometers. These simple techniques, though direct, involve much tedious labour in the gathering of relatively little data.

In the last decade or so, however, there have been

tremendous advances in the use of the microscope for the acquisition of numerical data from the image.

One of the fundamental aims of Materials Science is to correlate the properties manifested by materials with their microstructures. In the Department of Materials Science and Engineering and in the Institute for Materials Research at McMaster University, advanced studies on materials have been conducted using electron microscopy. The properties of materials are directly dependent upon their microstructure in terms of such factors as: density, crystallinity, presence and distribution of defects, size, regions of different chemical composition, etc.

During the last two decades there has been a rapid advance in understanding the details of the microstructure of materials, due primarily to the extensive application of electron microscopy. Many electron optical techniques are available for determination of structure and composition of materials. For example, the diffraction contrast technique (where the angular distribution of the scattered electrons, known as an electron diffraction pattern, is analysed) provides the Materials Scientist with a powerful method for determining the nature and spatial distribution of defects, crystal and surface structure. Besides, applications of electron microscopy contribute to orientation, density and distribution analysis of microstructures, as well as clusters, shape and size analysis.

It is considered that electron microscopy has a considerable effect on production practices, modification and improvement as well as on development of new classes of materials for practical use. [3]

As the research continued at the Institute for Metals Research the demand for computer assistance in electron microscope image analysis has grown. The present design was developed to facilitate to some extent the image analysis, stressing the modular approach and facilitating system extent according to user's requirements.

The following prerequisites were considered at the stage of development:

- flexible and user friendly interface to the system;
- implementation of the most applicable image processing algorithms for electron microscopy and materials science, as specified by the user;
- simplification of input image introduction, with options for subwindows processing;
- system modularity with emphasis on system growth facilitation;
- versatile utilization of the existing image processing software and hardware;
- minimization of processing time.

2.2 Preliminary hardware and software resources.

According to user's requirements and supplied equipment the design was based on the working station configuration.

Hardware:

- 1) Computer System - IBM AT compatible
- 2) Input Device - Video Camera;
- 3) System Display Device - Monochrome Monitor;
- 4) Mouse - LOGITECH Serial Mouse;
- 5) Professional Image Board Plus (PIB-Plus) - product of Atronic International Inc.;
- 6) PIB Display Device - Analog RGB (red, green, blue colour display) monitor;

Software:

- 1) Disk Operation System - MS DOS 3.3;
- 2) Atronic PIB Image Disk;
- 3) Newly implemented software.

The IBM AT machine has been equipped with a numerical coprocessor, a 20Mbyte hard drive and a 10 Mbyte floppy disk drive for image storing. An analog RGB monitor

with high resolution has been designated as the display device for all types of images: input, intermediate (under ongoing processing), and result images (after the processing has been completed).

The Professional Image Board Plus (PIB-Plus) used in the design is meant to be a device to capture "live" images from a video camera or other source for the purpose of saving, storing, manipulating, and redisplaying those captured images. It consists of a professional board and PIB Image Disk supplying software for simple image manipulation.

The configuration in which it has been used for the presented system development was based on the Dual Monitor Scenario consisted of:

- Monochrome monitor for the system control operations, operating system functions and the presented software coordination;
- Analog RGB monitor as a dedicated PIB display device.

The software supplied with the PIB-Plus board is based on the mouse interface and provides the following modes of interaction:

- Live Mode - a PIB display monitor presents the live image that is on the camera;

- **Display Mode** - the internal PIB memory content is displayed on the monitor screen.

Besides the user has an option for display image previously stored or to capture a new image and store it as the PIB-formatted file. While in Live Mode it is also possible to adjust colour, contrast or hue levels.

The board has three levels of pixel resolution:

- 512 X 512,
- 512 X 256
- 256 X 256.

Pixel information is stored in 16-bit format, every pixel has three colour components:

- bits 0 -4 blue;
- bits 5 - 9 green;
- bits 10 - 14 red;
- bit 15 overlay.

An overlay bit decides whether the display of that pixel is white or representing image data (where the image data may be coming from external source or from video memory). This feature has been used successfully in the designed mouse interface for window marking and other parts of the design.

The on-board memory consists of 256k bytes organized in 4 switch selectable 64k memory banks. Through

I/O Mode Register setting the required banks of memory are accessible. In different resolutions image information is stored in different memory banks. In the lower resolution it is possible to maintain the whole image in 4 blocks - using just one separate screen. This ability was used to produce the stereo image display, as will be described later.

The design was implemented using Quick C programming language with some additional use of Microsoft Assembler supplied in form of a library.

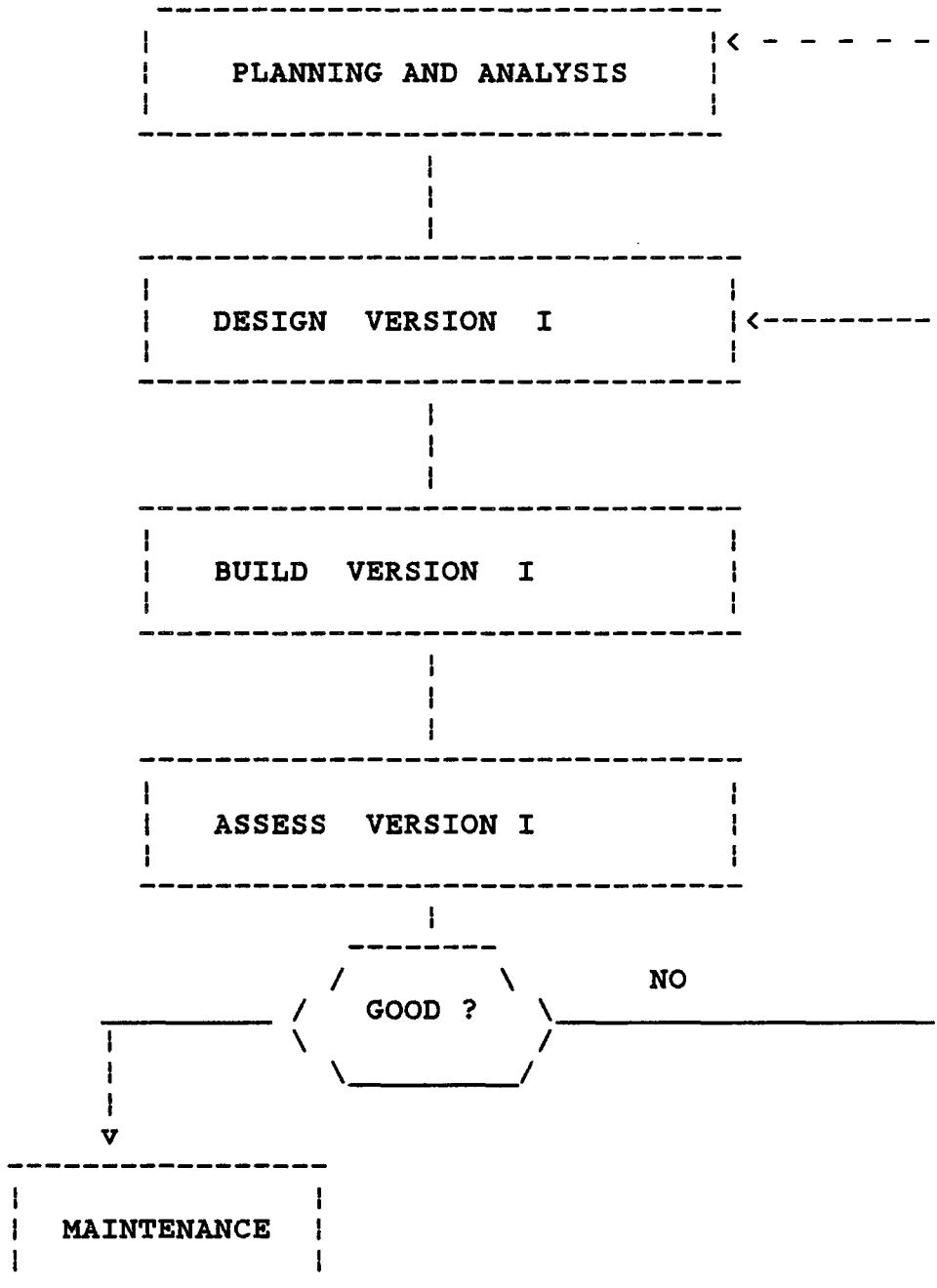
Particular features of the PIB used in the presented software are described in detail in chapter 3 with the detail description of all the implemented image processing algorithms.

CHAPTER 3 - CHARACTERISTICS OF IMAGE PROCESSING DESIGN

3.1 Methodology.

The software life-cycle encompasses all activities required to define, develop, test, deliver, operate and maintain a software product. Different models of the emphasize different aspects of the life-cycle, and no single life-cycle model is appropriate for all software products. The development process of this project was based on the method of successive versions. The initial version may be a skeletal prototype that demonstrates basic user functions and provides a framework for evolution of the product. Each successive version can incorporate the capabilities of previous versions and provide additional processing functions [5]. One of the reasons for developing a prototype is to explain various processing options to the user and to better understand user's needs.

The following figure illustrates the analysis phase followed by iterative design, implementation, and assessment of successive versions [5]. The dashed line indicates that assessment of Version I may dictate the need for further analysis before designing Version I + 1.



Originally in the presented project the user specified a basic, very limited set of functions to be performed by the system. They included linear

transformations of the image and binary slicing. According to those specifications a prototype was developed and the user was consulted. Once the simple system had been accepted new features were gradually added to the already functioning system. The final version of the design is presented in this discussion.

3.2 User's interface.

3.2.1 Input / Output interface.

3.2.1.1 PIB-file processing.

The PIB-formatted input image file - provided as one of the alternatives in the present design - can be obtained using the capture mode in the PIB software and a TV camera. In that case formatting is performed and supplied automatically. Besides the PIB-formatted file can be created as the output file in the final processing phase of the presented design.

In the PIB-file processing mode, once the user specifies the conversion mode (at present only the intensity calculation option is implemented, where the conversion is based on addition of intensity values for the three colours, the look-up table implementation being left for future implementation because of development time limitations) and the input file processing type (window or a total file) first the header is read and processed i.e.

the error checking routines are invoked to verify the PIB-format and maximum image size, then appropriate digital data is read and converted into the working file, according to the information about the file size.

3.2.1.2 LOGITECH mouse interface.

A mouse interface was provided as the additional mode to facilitate the input file specification. It was of a great concern to provide an easy way of specifying parts of an image, which were the most important from the user's point of view.

Using a mouse interface a user can traverse the monitor screen with displayed image on it and mark a rectangular shaped window for processing. Since the user can mark only the significant region for processing it results in reducing the processing time. After the user specifies the required window (there exists a possibility of remarking a new version whenever a user changes his mind) appropriate data is read directly from the PIB memory after enabling direct memory access to the board.

When the processing is terminated there exists an option to view the processed window and display it either at the original location, overlapping the old image, or at the user specified location on the screen.

3.2.2 General processing interface.

The presented design is an interactive, menu-driven image processing system. It has been developed, as mentioned before, for computer assisted analysis of electron microscope images at the Institute for Materials Research at McMaster University.

The design was based on existing facilities provided at the Institute, which implied some constraints on the design process. The designated approach was to maximize utilization of the existing image acquisition and display board and to surround it with newly developed software to fulfill best the user requirements.

The PIB-Plus board needs a digital data formatting equivalent to that which is automatically created when the original software is used. However in the presented system two modes of input image specification were implemented:

- a PIB-formatted file, which was created using the original software or was an output image from the previous program execution;
- image specification from the monitor screen using supplied mouse interface (Submenu "CHOOSE THE INPUT OPTIONS").

In both cases the user can specify either a whole input image for processing or a part of it (window). Window

specification in the PIB-formatted file option is defined by window coordinates entered by a user, while with the mouse interface the window is simply marked on the screen (Submenu "CHOOSE WORKING OPTION").

Before any digital enhancement or analysis processes are applied to the input image - reformatting is conducted to produce a 'working file'. The transformation involves two options: addition of intensity values for the three colours (red, green, blue) or look-up table implementation - left for further system enhancements (Submenu "CONVERT TO THE WORKING FILE").

The working file contains information about the image width and height as the first two parameters, and the converted data itself. At the processing time a maximum of two working files is available, since most operations are performed on a single file, with few tions - left for further system enhancements (Submenu "CONVERT TO THE WORKING FILE").

The working file contains information about the image width and height as the first two parameters, and the converted data itself. At the processing time a maximum of two working files is available, since most operations are performed on a single file, with few exceptions.

After each stage of processing the user has a choice of reusing the converted file or specifying a new

input file for further processing (Submenu "SAVING WORKING FILE"). Together with the working file name a format description is stored to prevent an incorrect use of a given working file, since for some procedures it is not necessary to convert from the PIB format and the working file is in the original format except for the header ("PIB FORMAT"). A working file constitutes an input file for all the processing options, after some procedure has been applied it becomes a 'result file'.

Most of the processing algorithms include an option for reprocessing, i.e. the working file can be reused and different options can be applied to obtain another result file. Once an 'exit' from current processing cluster has been chosen the user can decide whether to display, delete or save the result file(s) in the PIB format (Submenu "DISPLAY/DELETE OPTION").

At the display stage the working file data is converted to the display format i.e. the pixel values have to be transformed using the look-up table into the grey level values. Here, the grey level means a particular shade of intensity from the adopted grey scale. In this project a look-up table was constructed and saved in a data file for 94 shades of grey. For the colour file processing there also exist an option of colour look-up table, that contains colours in range of 0 - 204, although at the current processing stage only the black and white input image is available. The colour look-up table can also be used for

conversion of intensity variation in the black and white image into different colours. This technique is useful for several reasons. Many grey levels may be recordable on the negative but the human eye can only distinguish several of them. Since the eye can distinguish many more colours, then small intensity differences in the black and white image can be accentuated into distinct colours. Thus we can get a very effective colour map of the sample which can be more easily interpreted than a black and white image. [10] The processing structure chart is supplied to illustrate system architecture (see FIG. 1).

3.3 Aspects of modularity in the system.

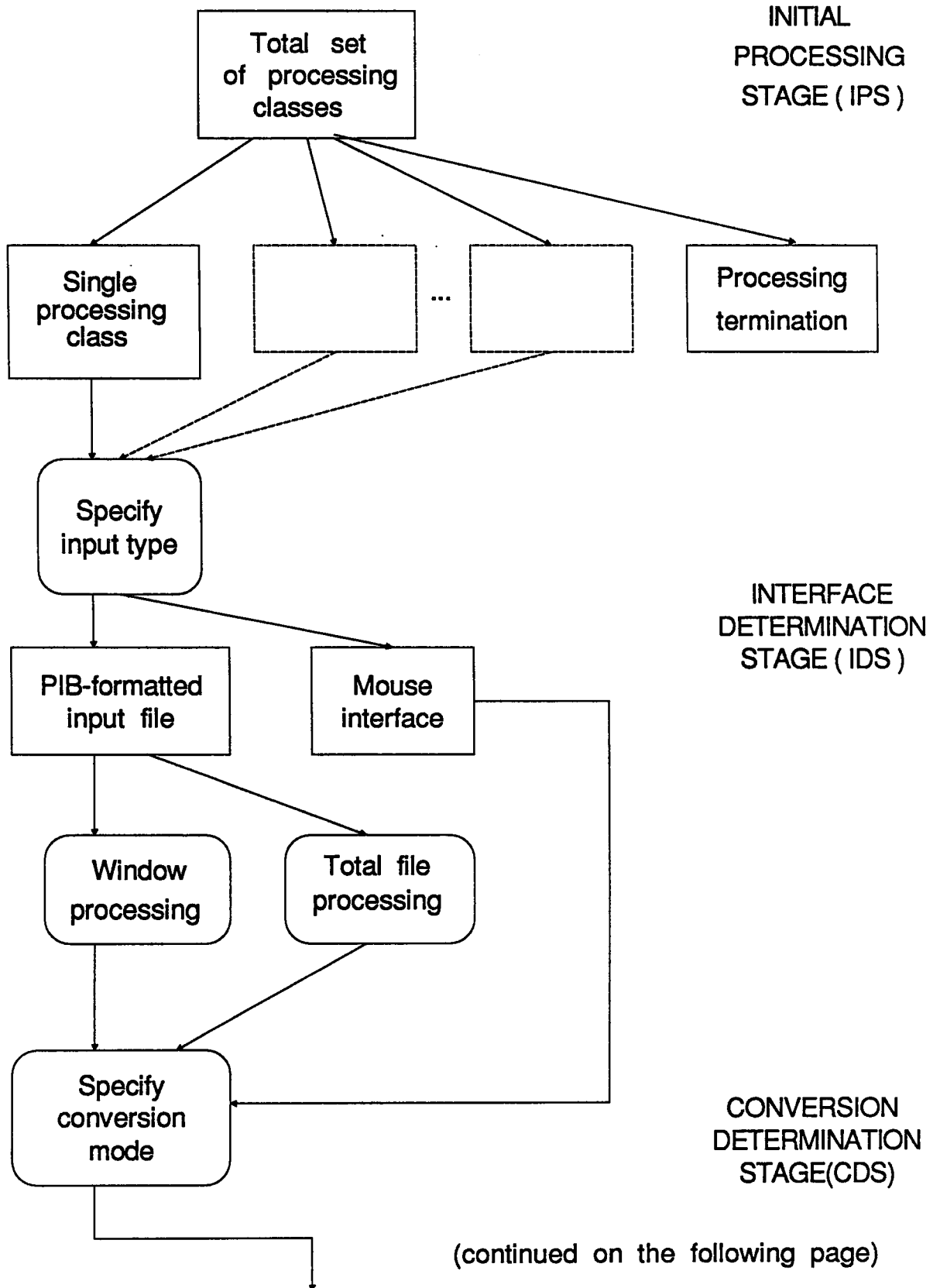
There are numerous criteria that can be used to guide the modularization of a system. Depending on the criteria used, different system structures may result. Modularization criteria include:

- the conventional criterion, in which each module and its submodules correspond to a processing step in the execution sequence;

- the information hiding criterion, in which each module hides a difficult or changeable design decision from the other modules;

- the data abstraction criterion, in which each

FIG 1. SYSTEM PROCESSING STRUCTURE CHART



module hides the representation details of a major data structure behind functions that access and modify the data structure;

- levels of abstraction, in which modules and collections of modules provide a hierarchical set of increasingly complex services;

- coupling - cohesion, in which a system is structured to maximize the cohesion of elements in each module and to minimize the coupling between modules;

- problem modeling, in which the modular structure of the system matches the structure of the problem [6].

In practice, a software system can be modularized using a single design criterion or aspects of several criteria. In the project a combination of conventional criterion and information hiding criterion was used. As the successive versions of the design were implemented a sequence of possible processing steps was widened. Thus every submodule represents a separate processing step which can be included in the user selected sequence. As the new module containing functions for a particular image processing algorithm was added the internal details of processing were hidden and the module's communication with other modules was establish through well-defined interfaces. based on data parameters passing and elements of control passing (flags).

A fundamental goal of software design is to structure the software product so that the number and

complexity of interconnections between modules is minimized. This goal can be achieved through considerations of system's coupling and cohesion. In the present design the control coupling and data coupling are the main characteristics of module interfaces. The control coupling involves passing control flags (as parameters or globals) between modules, so that one module controls the sequence of processing steps in another module. The data coupling involves the use of parameter lists to pass data items between routines.

The internal cohesion of a module as another modularization criterion is measured in terms of the strength of binding of elements within the module. The coincidental cohesion where the elements within a module have no apparent relationship to one another is the weakest and least desirable type of cohesion. The strongest cohesion, called the informational cohesion, is represented by a module containing complex data structures and several routines to manipulate the data structures. In the project the modules' cohesion is based mostly on sequential and communicational cohesion. Sequential cohesion of elements occurs when the output of one element is the input for the next element. The elements of a module possessing communicational cohesion refer to the same set of input and/or output data. Some of the modules were created on the basis of functional cohesion - grouping elements related to the performance of a single function.

3.4 System organization description.

The source code files and the object file are stored on the hard disk in the directory C:\image2. Besides, the backup copy of all the files needed for system run is provided on the floppy disk ("Image Software").

The program consists of eleven C modules and a library containing procedures written in Micro Assembler. The C modules are accompanied by the header files including data structures and constant declarations. The total list of modules constituting the design is as follows:

1) C - MODULES

- BITSlice.C
- DISPLAY.C
- DOUBLE.C
- FILTER.C
- INTENPRO.C
- MAIN.C
- MENUDRIV.C
- MOUSEDRV.C
- MULTFILE.C
- PIBTOOL.C
- SINGFILE.C

The module 'main.c' contains the driver for the main menu invoking particular submenu functions for the user-specified option. It also includes some functions used in all the other modules for menu displays and string input as well as error message display. The main module communicates only with the 'menudriv.c' module and the 'display.c' module through control parameters.

The 'menudriv.c' module contains the submenu drivers for all the image processing algorithms implemented in the project. Particular drivers maintain communication with the appropriate modules containing allotted functions through control and data parameters. Arrays containing names of working and result files are global and indices to those arrays are passed as parameters indicating the required working or result file.

The 'mousedrv.c' module contains all the functions implemented for mouse interface with the PIB-memory. The main function 'run_mouse' is invoked for the input image specification and all the parameters required for window identification are returned in the data structure 'window'. This module also contains a function 'set_point' used at the display stage for specifying the output image location.

The module 'singfile.c' contains all the procedures for opening and converting a single input file into the working file in both options of processing (window, total file).

A special way of reading and converting the image data from the input file (or window) was implemented if the input file size exceeds the size of memory segment. On the PC the maximum size of a single data item cannot exceed 64K. Since the screen can contain up to 512 pixels in width, the maximum number of rows for the array containing input data at the stage of processing was set to

63. To meet this constraint the input data is read in 'chunks' of 64K, converted and written into the working file. The number of required 'chunks' is calculated prior to the processing based on the input image height using function 'set_limits'. Two variables of array type were declared as globals for holding the input and intermediate data 'chunks' ('windrow', 'row'). These arrays are used at every processing stage when the data is processed and transferred from one file to the other, i.e. from the input file to the working file, from the working file to the result file, from the result file to the PIB memory etc.

The same algorithm is used at every processing stage when the whole working or result file is processed. The module 'singfile.c' also contains a single function used for linear transformations of a single working file.

The module 'multfile.c' covers all the procedures for opening and converting two input files into the working files as well as the function for linear transformations of multiple files.

The module 'bitslice.c' includes functions used for single bit and three LSB slicing operations. Separate functions were developed for PIB-formatted input file and for the PIB memory input because the conversion into the working file is not performed in this processing algorithm and the image data is read directly from the input not from the working file.

The module 'intenpro.c' covers all the functions

for histogram calculation and display. Separate functions were implemented for the single line data input and conversion both for the PIB-formatted input file and the PIB-memory input. This module also includes functions for calculation of simple statistics such as the mean, median, maximum and minimum values of intensity which accompany histogram displays.

The module 'double.c' contains all the functions used for the stereo vision effect. Two functions for writing the input images into the separate screens of the PIB-memory and the coordinating function for interchanging display of those images are provided.

The module 'filter.c' contains functions for mean, median and directional filters application. All these functions communicate with assembler routines for convolution matrix application. Data parameters are passed to the assembler routines indicating the source and the result data object. The reading from the working file and writing into the result file is performed in the C function.

The module 'display.c' contains all the functions used at the output processing stage. Functions for display on the PIB screen and for creating an output file in the PIB-format are provide separately. The routines for intensity range checking as well as functions used for intensity values scaling are also included in this module.

The module 'pibtool.c' contains routines written in C which were provided with the PIB hardware to facilitate programming interface with the board.

2) HEADER FILES

- GLOBAL.H
- EXTGLOB.H
- CONST.H

The header file 'global.h' contains data structure declarations used in the program. Some text constants and integer constants are included in the header file 'const.h'. All the required external declarations are included in the header file 'extglob.h'.

3) ASSEMBLER SOURCE FILES

- MEAN.ASM
- MEDIAN.ASM
- FILTERS.ASM
- REGIONS.ASM
- CALCINTEN.ASM

The assembler source files contain functions for filter applications ('mean.asm', 'median.asm', 'filters.asm'), binary image calculation ('region.asm') and data conversion from the PIB-format to the working file ('calcinten.asm'). The object files created from the assembler source file are linked with the C object files in the form of a library " ASM.LIB ".

The Quick C makefile utility is used for specifying all the

source modules and libraries as well as linking options for object file creation. All the C modules and the library "ASM.LIB" are listed in the makefile "IMAGERUN".

Two input files have to be available at the time of the system run - "TABLE1" and "TABLE2". They contain the look-up table data for PIB-format => working file conversions.

3.4 Error handling

Error handling could be divided into:

- input error processing stage;
- intermediate error processing stage.

At the input error processing stage, in case of PIB-formatted input file, first of all the PIB format restrictions are checked. The file has to have a header and its size can not exceed the maximum screen resolution. The information about the input file size is read from the header and appropriate checking is performed.

An error message is printed if:

- the specified input file does not exist;
- the given file does not meet the PIB requirements;
- the window size exceeds the file size, in the case of the window processing option;

- in the case of multiple file processing the files are of different size;

The error handling mechanism at the intermediate processing stage is based on the menu hierarchy. Whenever an error occurs at any of the lower level the user is informed about the error type and asked if he/she requires continuation. If so, the error-indicator is passed up the menu tree and processing restarts from the main menu; otherwise the program terminates.

At the processing stage the working file formats are compared. Since for 'Bit slicing operations' function the conversion to the working file is not necessary the original pixel values remain. It would be recognized as a conflict if two specified by the user working files had different formats (PIB FORMAT and WORKING FILE FORMAT).

In the case of incorrect input during the menu item selection the menu is always redisplayed to enable input correction.

CHAPTER 4 - IMAGE PROCESSING ALGORITHMS IMPLEMENTED IN THE SYSTEM.

4.1 Menu hierarchy.

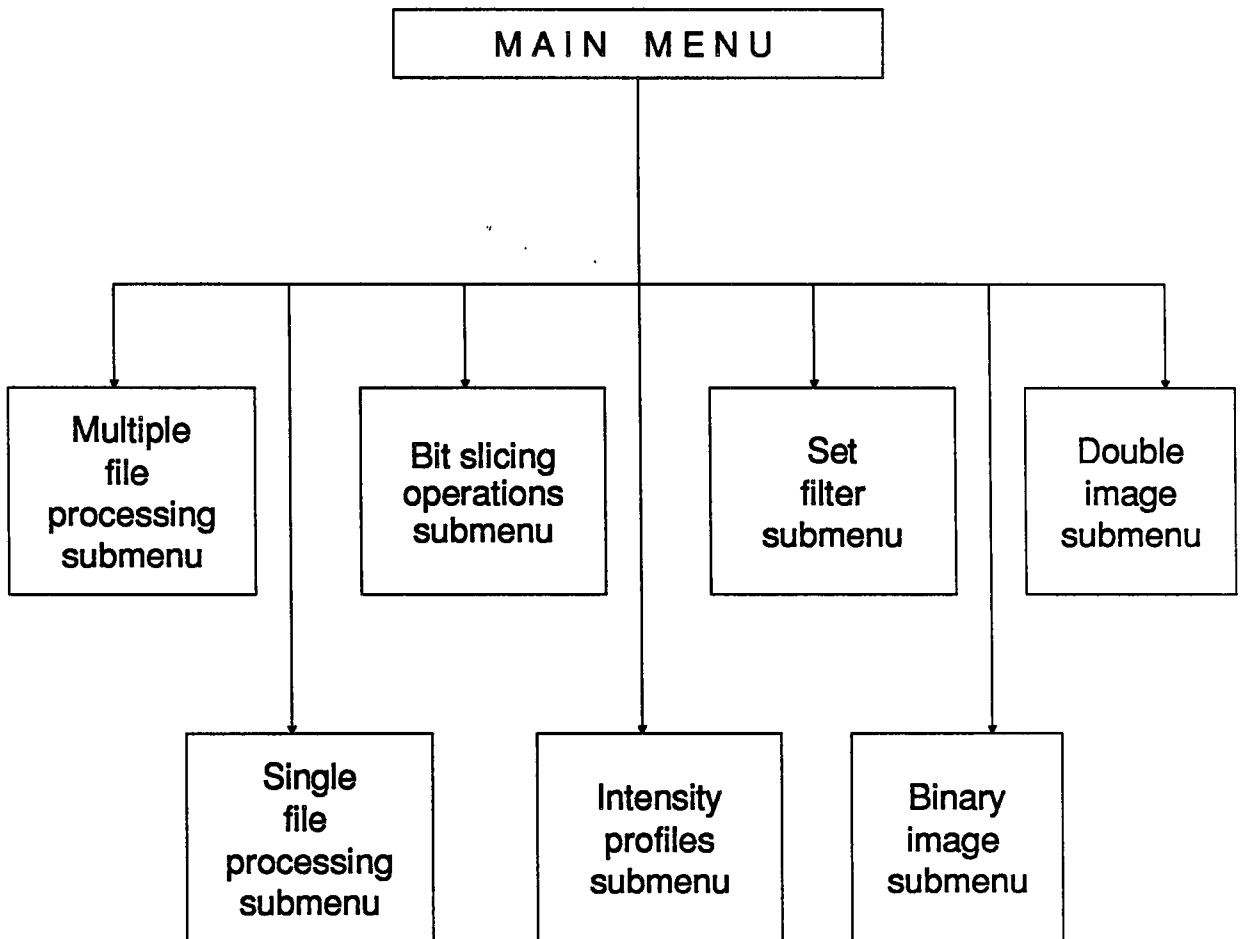
The opening menu is displayed after the system invoking command is entered. It contains entries for all the image processing functions available to the user. Single letter input was implemented for all the menu option indicators. Both upper case and lower case letters are accepted (see FIG 2.).

The following display presents the opening menu:

```
-----  
|                                     |  
|                               MAIN  MENU                               |  
|  
| T - Operations with two files  |  
| S - Operations with single file|  
| L - Bit slicing operations     |  
| I - Intensity profile         |  
| F - Set the filter            |  
| B - Binary image              |  
| D - Double image              |  
| E - Exit                      |  
|  
|-----|
```

Particular options invoke the sub-hierarchical menu for the specified processing function. To facilitate

FIG 2. MAIN MENU TREE STRUCTURE



user's orientation while traversing the menu-tree a subtitle has been added to indicate the immediate option up the menu-tree in the son-parent relation.

Option 'Exit' in this stage of processing causes termination of the program, while at every lower level it would step up in the menu hierarchy.

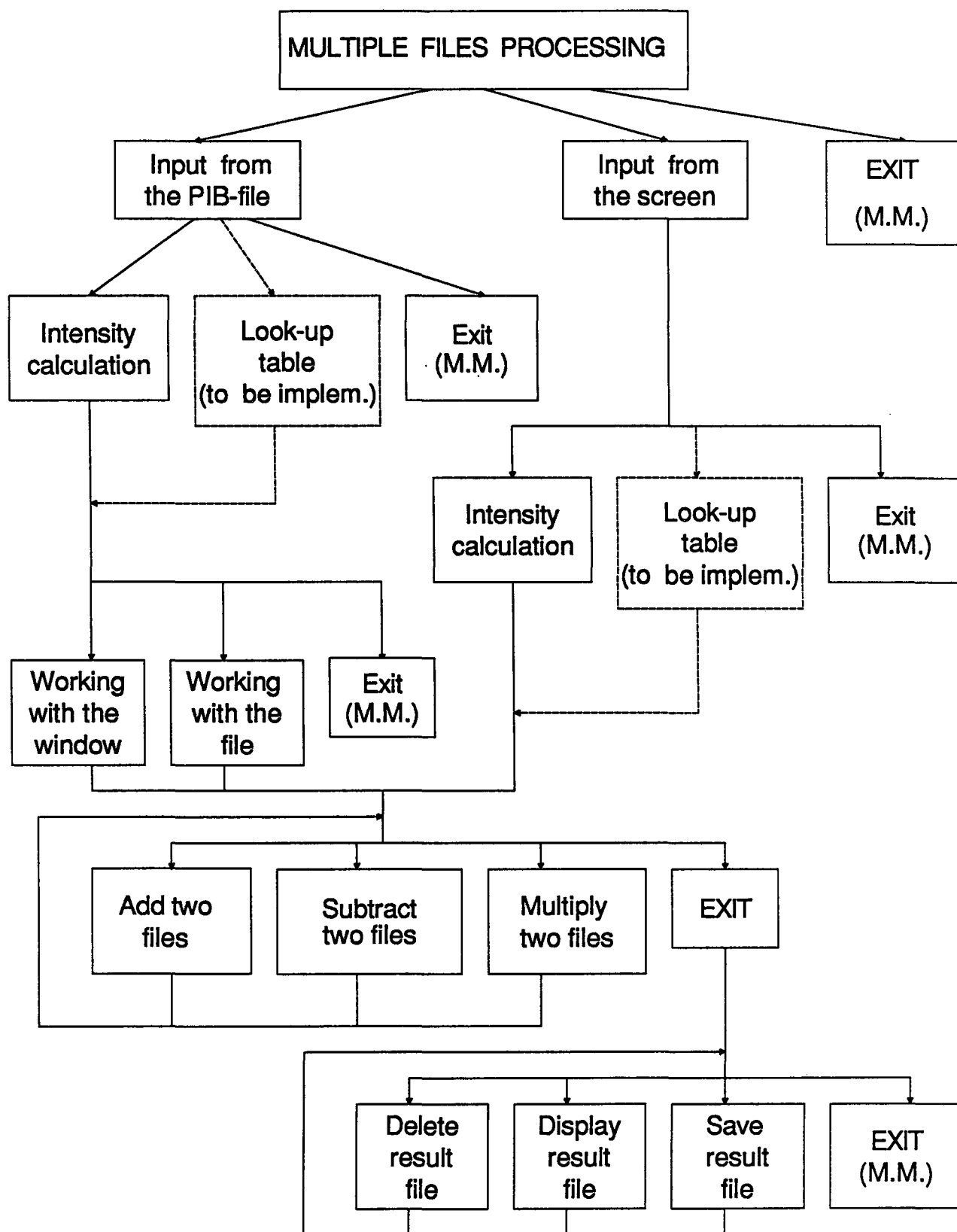
Schemes displaying the menu-tree and submenu-trees are provided for processing phases illustration.

4.2 Multiple files processing.

The multiple file processing options involve linear transformations of two working files (see FIG 3.). The following display appears on the screen when this option is selected from the 'MAIN MENU':

```
-----  
|                                     |  
|           OPERATIONS WITH TWO FILES |  
| (Submenu  of the MAIN MENU)        |  
|                                     |  
|   A - Add files                    |  
|   S - Subtract files               |  
|   M - Multiply files               |  
|   E - Exit                         |  
|                                     |  
|-----|
```

FIG 3. MULTIPLE FILES PROCESSING SUB - MENU TREE STRUCTURE



Every time a user specifies an operation to be performed the working files are checked and, if only one working file or no working file has been prepared, the user is prompted for the input file names. Afterwards, the result file name has to be specified. Until the 'Exit' option is chosen, the user can create more result files for different operation combinations.

Multiple image operations require special consideration on the same file size and format - all the appropriate tests are performed using information included in the headers of the input image files.

The smallest and the largest value of the individual pixels, after the application of a given operation to the working file, must be contained within the given (accepted by the system) grey scale. This is taken into account at the display stage of processing when the user has options to apply some scaling operations.

Multiple file processing can be useful in some cases for the noise reduction. For example, when multiple images which differ only in statistical uncertainty of the intensity values are available then, by adding them one can achieve the overall noise reduction if the noise component exceeds one least significant bit. Of course it is then necessary to divide the resulting image by the number of images summed to reduce the intensity values to the allowed range. On the other hand, subtraction of two

images can be used to assess the degree of change that has taken place.

Multiplication of two images is rarely performed in practice, although it is a useful one if an image of interest is composed of two or more distinctive regions, and if the analyst is interested only in one of them. It is also useful if the image detector is nonuniform in sensitivity.

Only some examples of possible applications are specified here for purpose of illustration.

4.3 Single file processing.

Linear processing of a single working file was implemented as one of the first applications. It involves a user-specified constant and a working file with the following menu options:

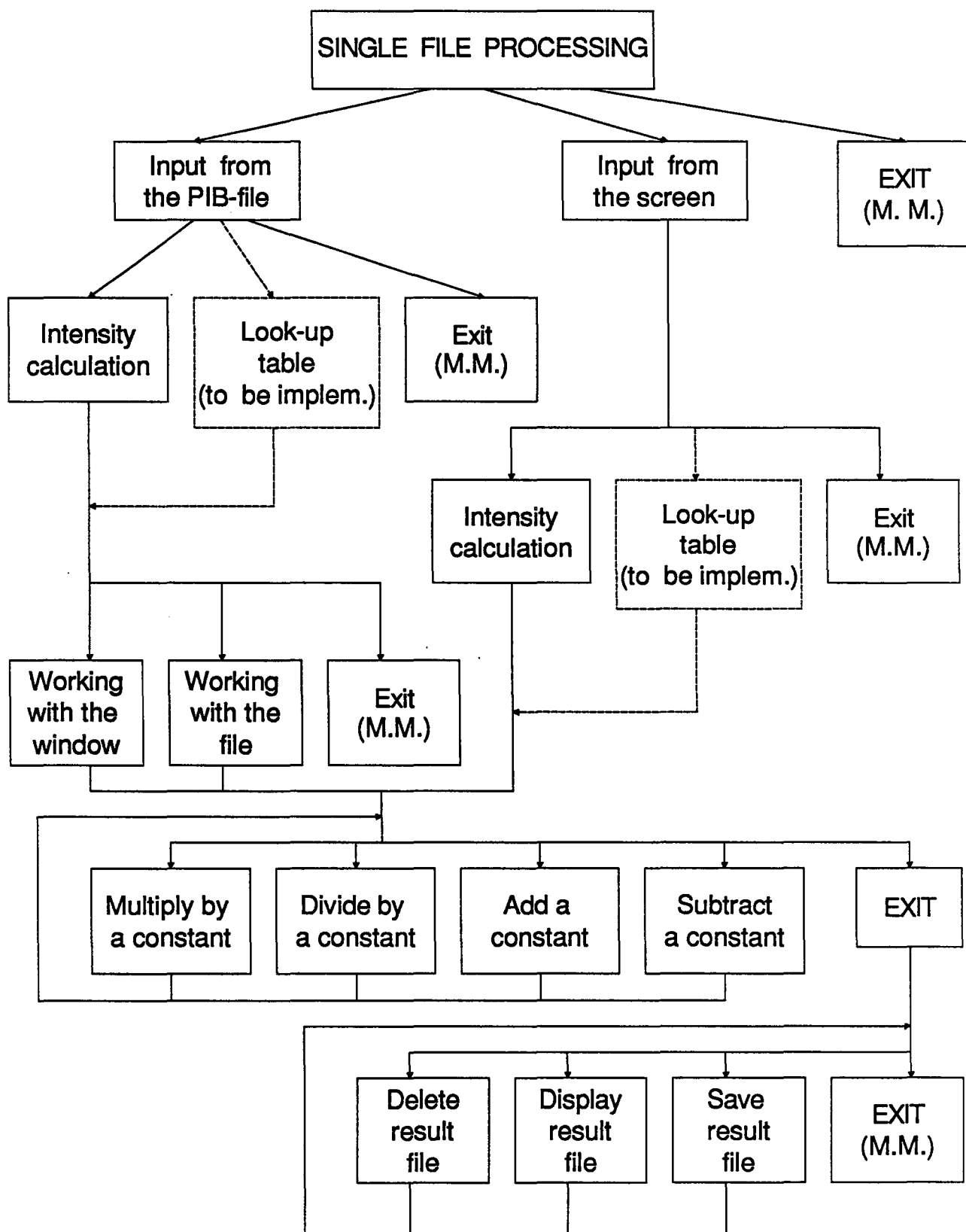
```
-----  
|                                     |  
|          OPERATIONS WITH SINGLE FILE          |  
| (Submenu of the MAIN MENU)                   |  
|  
| D - Divide by a constant                     |  
| M - Multiply by a constant                   |  
| A - Add a constant                           |  
| S - Subtract a constant                      |  
| E - Exit                                     |  
|  
|-----|
```

The indicated menu appears on the screen until the user chooses 'Exit' option so that more than one operation can be applied to particular input image. The user will be notified whether the operation had caused pixel values to go over the grey scale top level or below the low level at the display stage. At this time, additional options are supplied for scaling, upon user's request (see FIG 4.).

All the operations are performed on the pixel by pixel basis. Once a particular operation has been applied to the working file it becomes a result file under the user-specified name. A list of all result files is presented to the user after a 'Single file processing' sub-menu module is left. The result file can be viewed by the user on the PIB-monitor, or it could be saved in the PIB-format for further processing. There also exists an option for result file deletion.

Adding and subtracting a constant is a valuable operation for example, for contrast enhancement. The overall effect of this technique is to brighten up an image which is underexposed (too dark) or to darken an otherwise over-bright image. The contrast enhancement operation may involve combination of linear transformations applied to the image. One can perform, for example, constant subtraction to perceive the small variations in intensity values better. If this is followed by multiplication, the

FIG 4. SINGLE FILE PROCESSING SUB-MENU TREE STRUCTURE



differences will be exaggerated. The choice, as always, is left up to the user to select.

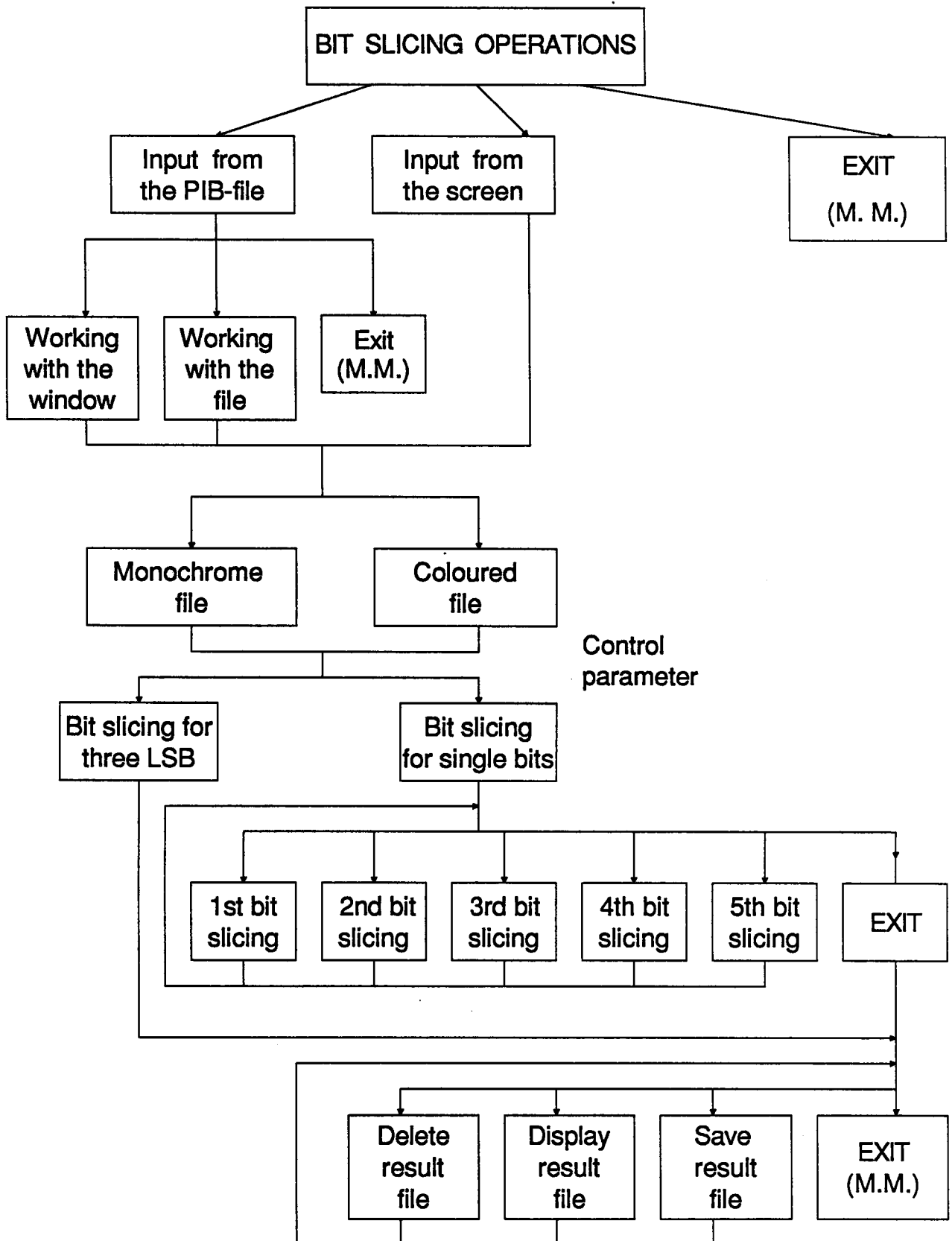
4.4 Bit slicing operations.

As described in chapter two, the pixel information is stored in 5-bits per colour format. It appeared to be very important for the user to find out how many bits carry significant information.

The function 'Bit slicing operations' was created to provide analysis of information significance for particular bits (0 - 4) separately and for the 3 least significant bits together (see FIG 5.). The term 'bit slicing' is used to indicate particular bit analysis (slice) for information content i.e. whether it is carrying any information or not.

To perform these operations the user has to specify if the input image is monochrome or colour, because in case of the colour input image the bit slicing is carried over all three colours, to produce a result file for all the three colours analysed separately. After this information has been provided to the system the user is faced with the following display:

FIG 5. BIT SLICING OPERATIONS SUB - MENU TREE STRUCTURE



BIT SLICING OPERATIONS

(Submenu of the MAIN MENU)

- S - Single bit slicing
- T - Bit slicing for three LSB
- E - Exit

When the option for three LSB is chosen processing progresses right away, on the other hand in case of the single bit processing option the following menu is displayed:

SINGLE BIT SLICING

(Submenu of the BIT SLICING)

- F - First bit slicing
- S - Second bit slicing
- T - Third bit slicing
- R - Fourth bit slicing
- V - Fifth bit slicing
- E - Exit

Here the analyst has a choice between particular bits testing. The result file created through bit slicing for a single bits is a binary file with white colour indicating all the pixels having a given bit set, and black otherwise.

In case of three LSB analysis the result file contains three colours combinations:

- red for the first LSB,
- green for the second LSB,
- blue for the third LSB,

creating a mosaic of colours for overlapping bit settings. The result files can be viewed by the user or they can be saved in the PIB-format.

The main purpose of bit slicing operation is to distinguish the significance of noise in the image. The image frame can be described as the sum of a signal information and random noise information, if the noise occurs at all. If the noise content for the particular bit slice proves to be significant, the user can decide to integrate image frames to achieve noise reduction.

By adding several frames of the same image, the noise fraction can be averaged off and thus the signal to noise ratio can become substantially enhanced.

Analysis for the three LSB might be considered as the first step of the significance analysis, while stepping through single bit slices could be undertaken to determine the number of images needing to be summed in the noise reduction process.

4.5 Intensity profiles.

This function provides a range of histogram calculations and presentations for single lines (vertical or horizontal) or for the whole image, with all the combinations of input from the PIB-formatted file or the PIB-monitor, subwindows and a mouse interface (see FIG 6.).

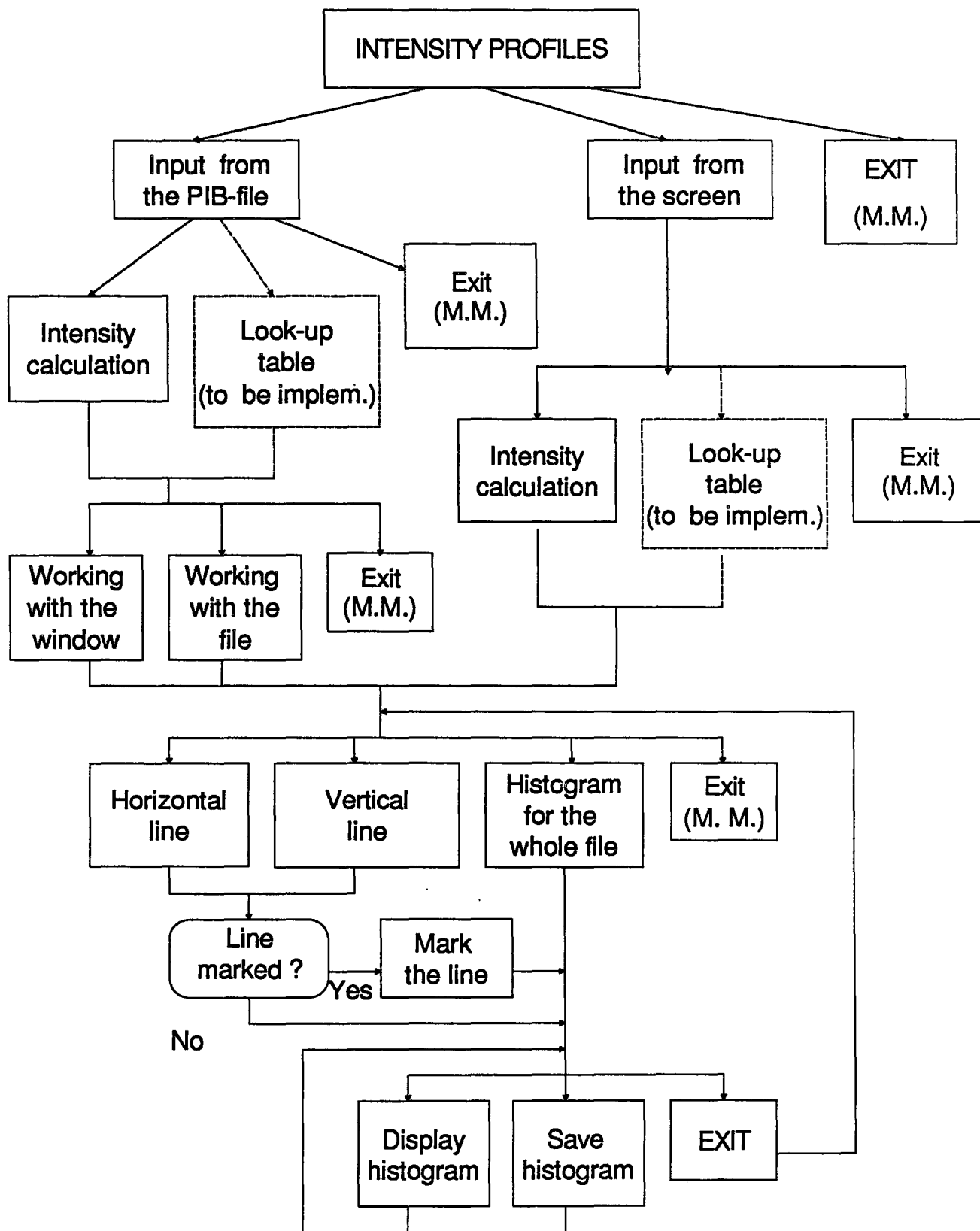
After the user specifies all the necessary input options the following menu display appears:

```
-----  
|                                     |  
|          HISTOGRAM CALCULATIONS    |  
| (Submenu of the INTENSITY PROFILES)|  
|                                     |  
|   V - VERTICAL LINE HISTOGRAM     |  
|   H - HORIZONTAL LINE             |  
|   T - TOTAL FILE CALCULATIONS     |  
|   E - Exit                         |  
|                                     |  
|-----|
```

A histogram in this case is a plot of the number of pixels occurring for each of the brightness values. It is displayed in the horizontal manner for horizontal line or the whole file, and vertically for vertical lines. The user also has an option to display the histogram in the top/bottom or left/right part of the screen, not overlapping the portion of an image being analyzed.

In case of single line analysis it is also possible to mark the given line for which the histogram is produced. It is performed using the overlay bit setting, thus

FIG 6. INTENSITY PROFILES SUB - MENU TREE STRUCTURE



preserving the original image. This particular feature improves greatly user's ability to relate histogram analysis to a given line.

Together with the histogram being displayed on the PIB-monitor, some descriptive statistical information is provided on the operator monitor. It contains mean, median, maximum and minimum values calculated for the given input image.

Very often in case of total file processing, the raw count of certain brightness values becomes difficult to display. When this occurs the scaling is conducted according to the maximum number allocated for the histogram bar ordinate in the histogram display. Thus together with the statistics also the value of the scaling factor is provided.

As already mentioned, this function is usually performed as the first operation applied to the input image since it is a good representation of the information content of the whole image, utilized in all the other processing phases. It provides the user with an appreciation of the quality of the original data (i.e. whether it is low in contrast, high in contrast, or multimodal in nature).

The histogram display is particularly useful in determining the threshold values used in functions described in the section related to binary image

calculations. By analysing the particular region of an image the user can determine the boundaries for intensity values used for binary image formation.

4.6 Filtering.

For a better understanding of the approach used in implementing functions for image filtering, some introduction is necessary. A characteristic of images called spatial frequency, defined as the number of changes in brightness value per unit distance for any particular part of an image, [6 p.638] refers to the "roughness" of the tonal variations occurring in an image. Image areas of high spatial frequency are tonally "rough". That is, the grey levels in these areas change abruptly over a relatively small number of pixels. "Smooth" image areas are those of low spatial frequency, where grey levels vary only gradually over a relatively large number of pixels.

Low-pass filters are designed to emphasize low frequency (large area changes in brightness) and deemphasize the high frequency components of an image (local detail).

High-pass filters do just the reverse. They emphasize high frequency, and deemphasize the low frequency information.

The filter applications are based on convolution, where a moving window is established, which contains an

array of coefficients or weighting factors (referred to as operators or kernels) [6]. The kernel is moved throughout the original image and the pixel value at the center of the kernel in a second (convoluted) output image is obtained by multiplying each coefficient in the kernel by the corresponding pixel data in the original image and adding all the resulting products. This operation is performed for each pixel in the original image. In this project, 3 X 3 convolution matrices were used.

In the presented design a user has the following options while entering 'Set the filter' sub-menu:

```
-----  
|  
|          SET THE FILTER  
|  
| (Submenu  of the MAIN MENU)  
|  
| M  - MEAN FILTER  
| D  - MEDIAN FILTER  
| C  - CONVOLUTION MATRICES  
| E  -  Exit  
|  
-----
```

The mean and the median filter are examples of low-pass, in other words smoothing, filters. The effect is to reduce the overall variability of the image and lower its local contrast. At the same time those pixels that have larger or smaller values than their neighbourhood average are respectively reduced or increased in value, so that local detail is lost - in this way the noise components are also blurred.

In case of the mean or the median filter all the necessary calculations are performed right away by calling routines written in Microassembler. This step was undertaken to speed up the processing, since the calculations are very time consuming, especially for a large input images. While developing the programs for mean and median filters, optimized algorithm versions were implemented, which significantly lower the processing time [6].

The option for 'Convolution matrices' involves another sub-menu for more detailed filter specification:

```
-----  
|  
|          CONVOLUTION MATRICES  
|  
| (Submenu  of the SET THE FILTER)  
|  
|   D - Directional filters  
|   I - Isotropic filters  
|   U - User specified filter  
|   E - Exit  
|  
|-----
```

Filters specified in the first two groups belong to the high frequency filtering domain. As mentioned before the result of applying those filters is to sharpen (de-blur) the image and accentuate linear features, opposite functions to those which low-frequency filters perform.

An edge is considered as a discontinuity or sharp change in the grey scale value at a particular pixel point [7 p.260]. Edge enhancement delineates linear features and edges.

By choosing option 'Directional Filters' a set of five convolution masks is presented to the user. Most of them involve subtraction of the pixel values within a window from a multiple of a central pixel, thus accentuating or sharpening edges without regard to edge directions.

Under 'Isotropic filters' option, the analyst has again a set of four convolution matrices to perform edge enhancement.

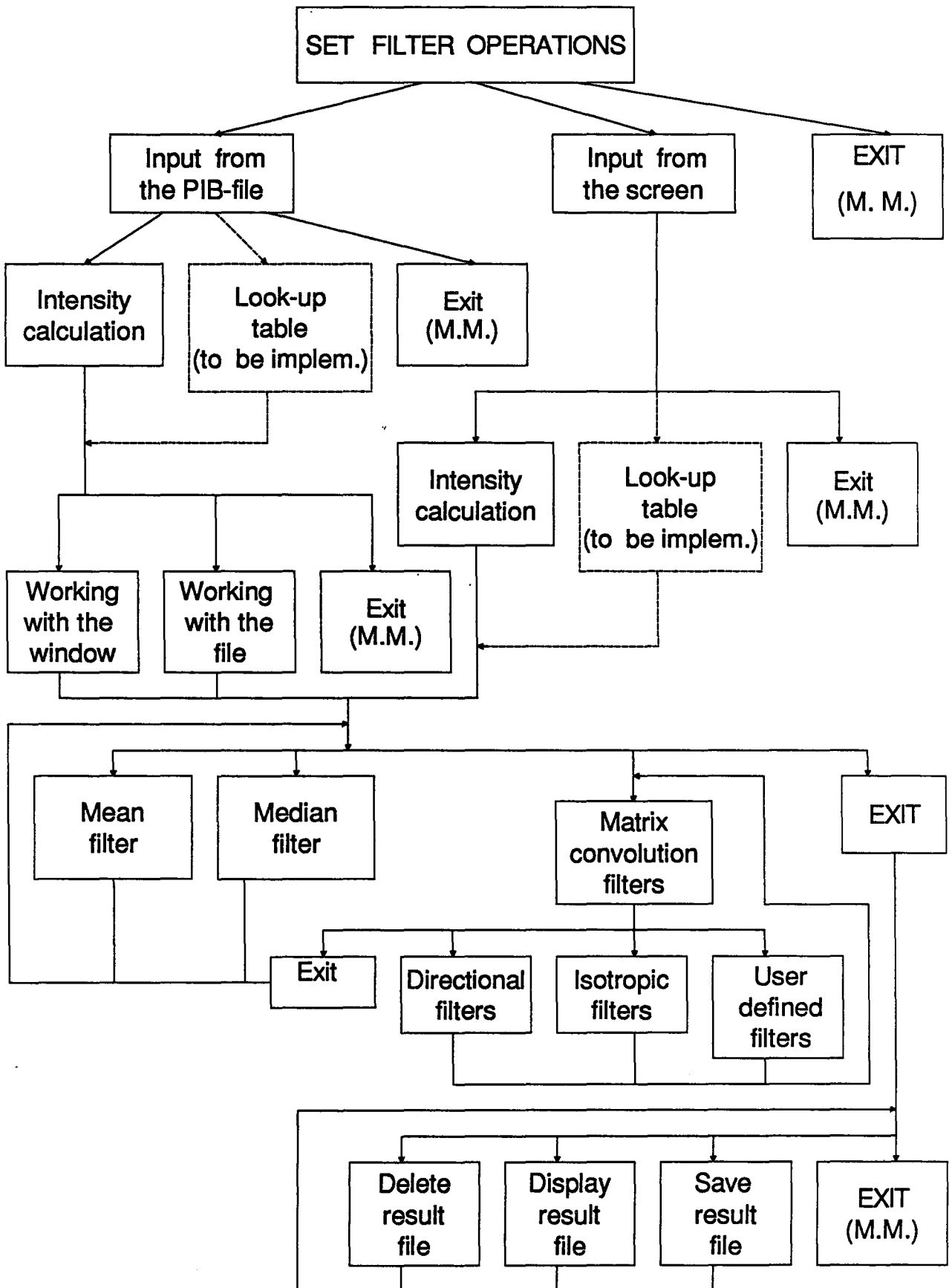
But by applying these masks edge enhancement in particular directions is obtained. The user can achieve the required "direction" of edge enhancement by selecting appropriate mask.

The third option allows the analyst to specify a filtering mask at the processing time, which makes the system more flexible (see FIG 7.).

4.7 Binary images.

Before the 'Binary image' function is invoked, the analyst usually inspects an image histogram to decide upon the possible thresholding ranges for image

FIG 7. SET FILTER SUB-MENU TREE STRUCTURE



segmentation. Segmentation of an image entails the division or separation of the image into regions of similar attribute [9 p.534] - in our case intensity value.

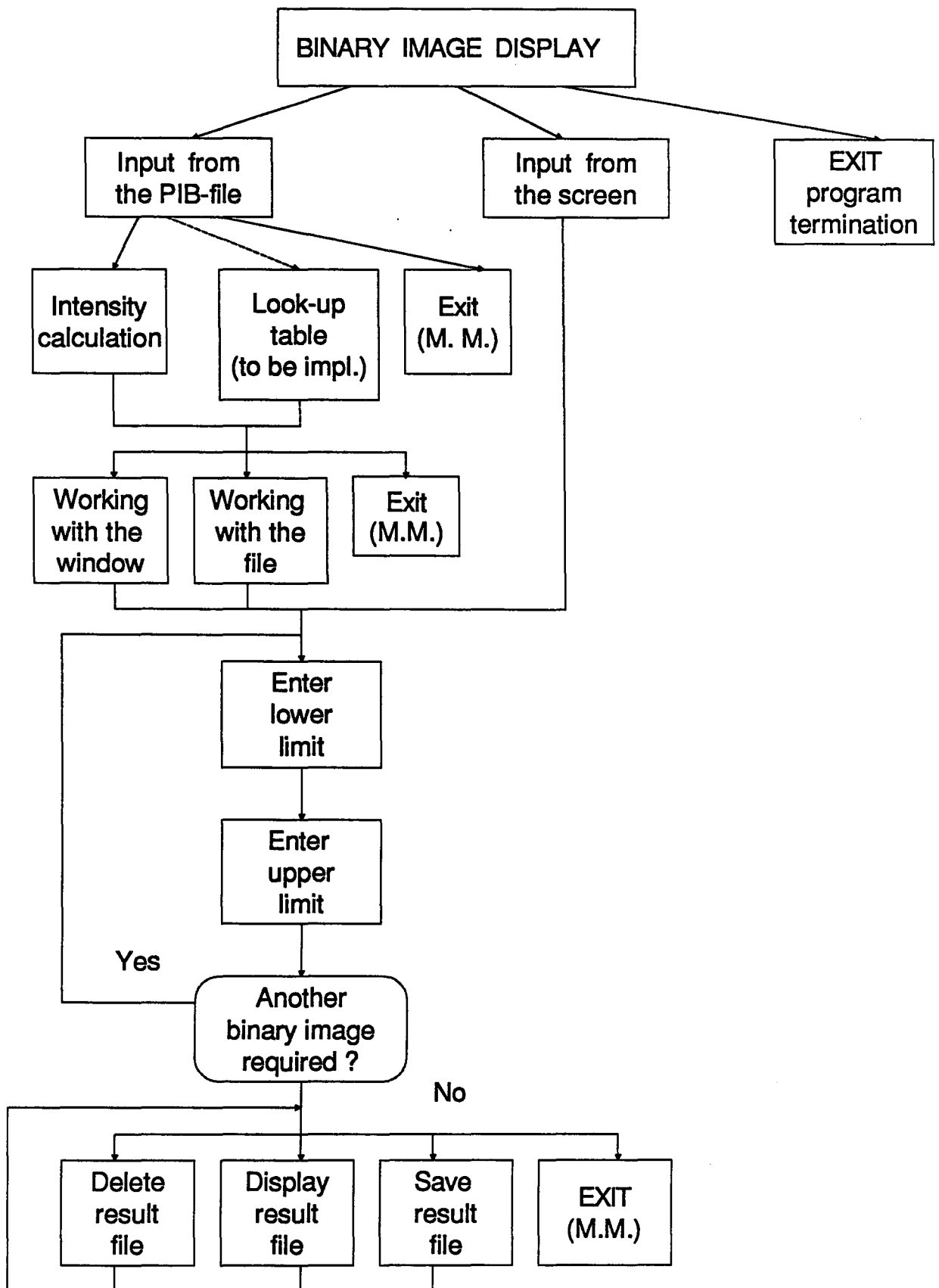
According to the user-specified lower and upper bounds the image segmenting is performed - the values within the boundaries are set to white colour and the others are set to black (see FIG 8.). While performing the testing operation also the 'white pixels' counting process is conducted, so that at the end the analyst is presented with the total binary region area in terms of numbers of pixels.

Binary region finding might be a very useful operation for preparing binary mask for an image, for example. Once the mask is ready, multiple file operations can be used to eliminate parts of an image which are not important to the analyst. This can also serve as an example of combination of operations applied to a given image, making use of the functions present in the design.

Binary images are also produced to obtain features, which are going to be quantified, i.e. involving element counting, individual area calculations, shape determination and quantitative description of the degree of clustering. Binary images are the first stage of this long processing phase.

While describing the system functions, as mentioned before, only very limited and simple examples are given. It is left up to the user, with training in image

FIG 8. BINARY IMAGE SUB - MENU TREE STRUCTURE



processing to decide upon operations to be performed dependent on the image and on the information sought.

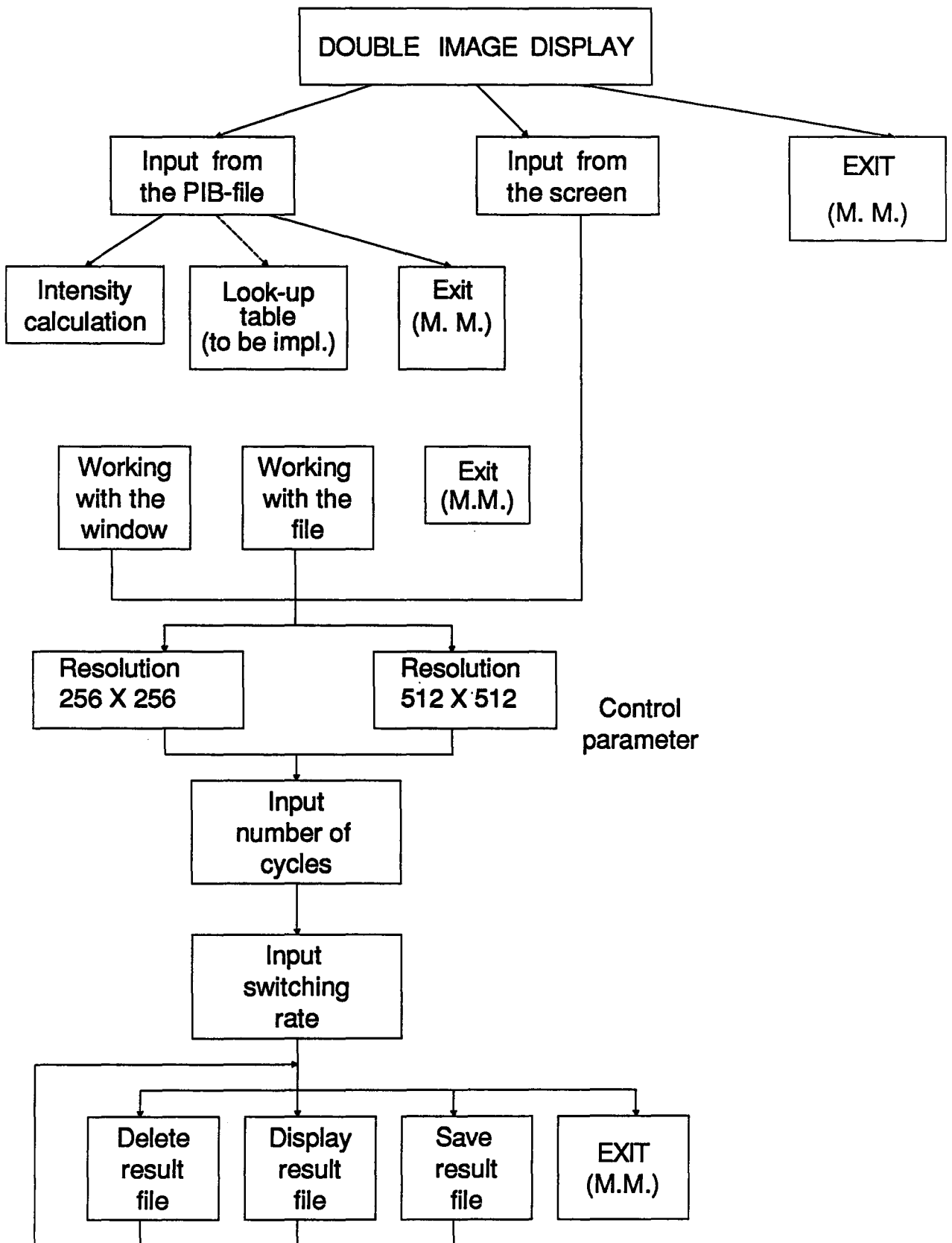
4.8 Double images.

The PIB-Plus hardware provides abilities to display images at the resolution 256X256 in two separate screens (screen 0 and screen 1). This feature was used in creating the 'Double images', which generate a stereo vision effect, as described below. Two input images are first written into the first and second screen, after enabling access to portions of the on-board memory by setting the appropriate I/O registers on the board (see FIG 9.).

Afterwards just by switching the display from one screen to the other the user can view alternate images. The switching rate, as well as the number of full switching cycles is set by the user.

When the two images represent the same object, but are taken from the different angles, then one has a three dimensional impression of the object. This function opens up the possibility of image depth analysis and perception of objects location within the image when viewing the two images at a certain rotation rate. The switching rate is entered by the user, who can decide on its value upon previous examination.

FIG 9. DOUBLE IMAGE SUB-MENU TREE STRUCTURE



CHAPTER 5 - FUTURE ENHANCEMENTS AND CONCLUSIONS.

5.1 Future possibilities.

While developing the image processing system a great emphasis was put on systems modularity. It was considered to be very important for further system's expansion. As mentioned before the 'successive versions' method was used for system development. A subsequent function was added after a previous function had been completed. The same approach is recommended for further development, since only the most important algorithms were implemented.

There exists a wide range of other applications which might be developed within the given environment. In particle counting, for example the image features are reduced equally and approximately isotropically (i.e. eroded) to eliminate isolated noise pixels before counting the features. Equal and approximately isotropic enlargement (i.e. dilation) of image features can then be used to restore the area of these features for area determination of the set of counted features. The demand for computer assistance is quite dynamic and it probably will increase further so the system extension could be quite large.

The program listings are documented in a

standardized way to improve future analysis. Every procedure has a header with information about its purpose, parameters and modifications resulting from its performance.

Besides development of new functions it is realized that the present design could be improved in many ways. It was mostly because of the time limitations that not everything could have been completed as it was designed. Also some technical problems with the hardware limited the presented implementation. Many functions provide only some image preprocessing abilities and could serve as a basis for further development.

The following indications are given to facilitate future continuation:

- 1) First of all the user interface could be improved by introducing menu containing icons to facilitate the user's interaction;
- 2) The system could be made more PIB-software independent by implementing the image capture through the TV-camera straight from the system menu;
- 3) The mouse interface could accommodate marking of different subwindow shapes, not just rectangular one;
- 4) Some calculation optimization for particular directional and isotropic filters could be implemented;

5) The histogram display could be implemented in such a way that the original image would be automatically restored when the histogram display is no longer needed (for example by saving the underlying data before histogram is displayed).

However the constraint of processing speed in the present working station environment (even operating with 12MHz CPU speed) is the most cumbersome one. Thus the code optimization to achieve processing time minimization should be considered as the most important factor for all further system expansion.

5.2 Conclusion.

The image analyst must evaluate several important image processing considerations prior to extracting meaningful information from a given image. Usually the analyst begins by computing fundamental statistics and viewing histograms to determine an image quality. Based on these observations the image is then preprocessed to reduce distortions in the data. Various enhancements may then be applied to the corrected data for improved visual analysis or as input to further digital image processing.

The present design was implemented to assist the user in electron microscope analysis. It contains

functions for image processing and enhancement as well as statistical description. As mentioned before it was not meant to be a complete and exhaustive implementation but a basic and easy to expand system.

The user manual attached as Appendix A should allow for better understanding of the system functioning and should solve the possible problems occurring at the time of system execution.

REFERENCES

- [1] "UNIVERSAL DIGITAL IMAGE PROCESSING IN EUROPE - A COMPARATIVE SURVEY ", PAN, Z. KULPA, WARSAW, JAN 1981;

- [2] "DIGITAL IMAGE PROCESSING OF REMOTELY SENSED DATA", R. M. HORD, ACADEMIC PRESS, NEW YORK, 1982;

- [3] "PRACTICAL ANALITICAL ELECTRON MICROSCOPY IN MATERIALS SCIENCE ", PHILIPS ELECTRONIC INSTRUMENTS INC., D. B. WILLIAMS, NEW YORK 1984;

- [4] "ANALYTICAL AND QUANTITATIVE METHODS IN MICROSCOPY", CAMBRIDGE UNIVERSITY PRESS, Edited by G. A. MEEK, Y.H. ELDER, NEW YORK, 1977;

- [5] " SOFTWARE ENGINEERING CONCEPTS", R. FAIRLEY, MCGRAW - HILL INC., NEW YORK, 1985;

- [6] " REMOTE SENSING AND IMAGE INTERPRETATION", T. M. LILLESAND, JOHN WILEY & SONS INC., NEW YORK, 1987;

- [7] " COMPUTER PROCESSING OF REMOTELY SENSED DATA ",
P. M. MATHER, PAUL M.MATHER INC., NEW YORK, 1987;
- [8] " INTRODUCTORY DIGITAL IMAGE PROCESSING ",
J. R. JENSEN, PRENTICE - HALL, NEW JERSEY, 1986;
- [9] " DIGITAL IMAGE PROCESSING ", V. K. PRAT,
NEW YORK, JOHN WILEY & SONS INC., 1978;
- [10] " COMPUTER PROCESSING OF ELECTRON MICROSCOPE
IMAGES", P. W. HAWKES, SPRINGER - VERLAG, NEW
YORK, 1980;
- [11] " PROFESSIONAL IMAGE BOARD PLUS ", REFERENCE
MANUAL, ATRONICS INTERN. INC., 1988;
- [12] " MICROSOFT QUICK C", REFERENCE MANUAL, MICROSOFT
CORPORATION, 1984-87;
- [13] " MICROSOFT MACRO ASSEMBLER ", REFERENCE MANUAL,
MICROSOFT CORPORATION, 1987;
- [14] " LOGITECH MOUSE - USER'S MANUAL ", REFERENCE
MANUAL, LOGITECH INC., 1987.

FIGURE CONTENT

FIGURE 1. SYSTEM PROCESSING STRUCTURE CHART.

FIGURE 2. MAIN MENU TREE STRUCTURE.

FIGURE 3. MULTIPLE FILES PROCESSING SUB-MENU TREE STRUCTURE.

FIGURE 4. SINGLE FILE PROCESSING SUB-MENU TREE STRUCTURE.

FIGURE 5. BIT SLICING OPERATIONS SUB-MENU TREE STRUCTURE.

FIGURE 6. INTENSITY PROFILES SUB-MENU TREE STRUCTURE.

FIGURE 7. SET FILTER SUB-MENU TREE STRUCTURE.

FIGURE 8. BINARY IMAGE SUB-MENU TREE STRUCTURE.

FIGURE 9. DOUBLE IMAGE SUB-MENU TREE STRUCTURE:

USER'S MANUAL

1. System execution description.

Presuming that the DOS environment variables are set for accessing programs in the subdirectory the following command invokes the main menu:

```
IMAGERUN      'RETURN'
```

The main menu and submenu structures are described in thesis chapter 3. A standard format for all the menu displays has been adopted, as was shown in chapter three where all the menu structures were discussed. For some applications the " Y/N "-question format was used and description of actions undertaken for those questions will be presented at this point.

1.1 Working file inspection and determination stage.

In every function performed by the presented design a working file is used. As mentioned before the working file, including data converted from the PIB format, can be used in many processing applications. After the working

file has been processed by the specified function the following display appears on the screen:

THE FOLLOWING ARE THE WORKING FILES WHICH WERE CREATED

1. First working file name.
2. Second working file name.

Do you want to delete any of them ? Y/N

- answer 'Y' or 'y' will cause file deletion and the user will be prompted for file number (working file number one or two) :

Enter the file number:

- answer 'N' or 'n' indicates that the user might want to reuse the working file. To confirm this assumption the following question will be displayed:

Do you want to use the working file in further calculations?

Answer 'N' or 'n' implies new input file specification for the following processing steps (new selection from the main menu). If the answer was 'Y' or 'y' the following menu will be displayed:

|
| SAVING OPTIONS |
|
| F - SAVE FIRST WORKING FILE |
| S - SAVE SECOND WORKING FILE |
| B - SAVE BOTH WORKING FILES |
| E - EXIT |
|
|-----

As mentioned before two working files are specified for multiple file processing and double images, otherwise single working files are created. If the option for both files is chosen from the above menu then the user will be confronted with further questions about the required file at the beginning of the next specified processing function. Selection of a working file which does not exist will cause an error and the user will be notified about it at the next processing step (whenever the user-specified algorithm will require access to the working file).

Also if the working file specified for reprocessing does not have an appropriate format (PIB-format confused with the working file format or vice versa) the user will be notified about it and he will have to enter a new input file.

1.2 Result file inspection stage.

The result files list is presented at the end of each submenu processing termination stage. Before the program control is returned to the main menu the user has a chance to display or save the result files produced at the current processing stage. However all the result files created by different functions are preserved, in case the user might want to view all the result files at the end of processing.

The following menu is displayed at the end of a single processing function (submenu tree in the menu hierarchy):

```
-----  
| SAVE/DELETE OPTIONS FOR THE RESULT FILES |  
| D - DELETE THE FILE |  
| P - DISPLAY THE FILE |  
| S - SAVE THE FILE AS THE PIB FILE |  
| E - EXIT |  
-----
```

After the user selects the appropriate action the list of all result files created so far is displayed in the following format:

THE FOLLOWING ARE THE RESULT FILES WHICH WERE CREATED

1. First result file name.

2. Second result file name ... and so on until the result file list is exhausted.

.
. .
.

The 'SAVE/DELETE OPTIONS FOR THE RESULT FILE ' menu will be displayed over until the user chooses option 'Exit' which will move the program control to the main menu.

If the 'Delete ...' option was chosen then the result file list is rearranged so only the existing files are displayed.

1.3 Display /store processing stage.

Working file manipulations can cause some pixel values to go over the adopted intensity range. Thus every time the user wants to display some result file on the screen or save it in the PIB-format, appropriate actions are undertaken to make sure the implemented intensity range is preserved by all the pixels. Otherwise some unpredictable values would be assigned to the pixels at the time of output mapping of intensity values.

If the result of the checking operation shows some expansions over the current scale (grey or colour) the user will see a message of the following form:

VALUES IN THE WORKING FILE ARE OUT OF THE RANGE !!!

Acceptable range of values is from |_| to |_|

Values in the file are from |_| to |_|

Do you want to scale them before writing the pib file
/ displaying the file ? Y/N

If the user answers 'N' or 'n' then he will be confronted
with the following message for reassurance of his
decision:

YOU WILL SEE SOME UNPREDICTABLE COLOURS IN YOUR PICTURE !!!

Don't you really want to change your mind ? Y/N

If the users confirms his decision the specified result
file will be displayed (or saved, as required). On the
other hand if the user wants to make some corrections the
following menu will be displayed :

```
-----  
|                                     |  
|           FILE UPDATE OPTIONS       |  
| M - MULTIPLY BY A FACTOR           |  
| A - ADD AN OFFSET                   |  
| D - DIVIDE BY A FACTOR             |  
| E - EXIT                            |  
|                                     |  
|-----|
```

According to the user's choice of correction one of the following lines will be displayed as a prompt for the constant value:

Enter the factor (mult):

Enter the offset : (for addition/subtraction)

Enter the scaling factor (division):

The correction operation is performed and if the values within the working file are still out of the adopted range then the user is asked again to specify the operations. However he might not want to do so (simply answer 'no' to both of the initial questions).

Every time the image is displayed on the PIB-monitor or, is saved as the PIB file, a description of the image in terms of minimum and maximum value of intensities is provided on the system control monitor. The description also contains a statement about the intensity values, whether they were contained within the accepted range or not. The format of the description is as follows:

PICTURE DESCRIPTION

Values within the given range

or

Values out of the given range (if corrections were not applied)

Minimum value |_!

Maximum value |_!

At the display stage the user can either indicate the location on the screen where he wants to place the result image or whether the result image is to be displayed at the original location, preserving the coordinates from the input file and translating them into the screen coordinates. The latter applies only to the images specified at the input stage as the PIB file, since for the screen input the window of the image is read directly from the PIB memory and coordinates are not known. All the functions presented in the menu have been implemented, except for the look-up table given as the conversion option from the input to the working file. That has been indicated on the menu display. If this option is indicated before its implementation the error message will be displayed and the program control will be moved to the main menu.