

BLOCK ADJUSTMENT ON PERSONAL COMPUTERS

Dipl.-Math. Hermann Klein
Photogrammetry, Software, Consulting
Lehenbühlstr. 59, 7253 Renningen
West Germany
Presented Paper for Commission III

Introduction:

The enormously increased efficiency concerning computing time, available disk space and the simultaneously drop down of costs have made the IBM PC an interesting computer system to use for both scientific research and engineering applications. Having developed the computer programs PATM for block adjustment with independent models and PATB for block adjustment with bundles, it was now of interest to transfer these programs to personal computer systems. This paper deals with the necessary investigations, the the effort to develop real comfortable and userfriendly PC programs and the capabilities of these PC versions.

The IBM PC family:

The first IBM Personal Computer was introduced in the fall of 1981. Later, as PC sales zoomed beyond all expectations, this PC became recognized as the standard for all serious desktop computers.

From the programmer's side of view, all members of the PC family consists of a processor, memory chips and several circuit chips. All these main components are located on the system board. Other important parts are located on extension boards, which may be plugged into the system board.

The central processing unit (CPU) of the PC is the Intel 8088 16-bit microprocessor. It controls the computer's basic operations by sending and receiving control signals, memory addresses and data from one part of the computer to another along a network of interconnecting pathways called the bus. The microprocessor is tied to at least 64K bytes of memory, some built in ROM programs, such as BASIC and the ROM-BIOS, and some very important support chips.

The microprocessor cannot control the entire computer without some help - nor should it. It delegates certain functions to other chips, the CPU is free to attend to its own work. Some of these chips control the external devices, for example the disk drive, the display screen and the keyboard, or control the flow of information throughout the internal circuitry.

All members of the PC family are designed to work with additional processors such as the math coprocessor. These optional chip helps to reduce and to speed up the workload of the main CPU. The math coprocessor performs floating point operations up to 50 times faster than can be achieved with the main processor.

When the PC family was introduced in 1981, it used one main type of storage device: the standard 5¼ - inch floppy disk, which was double sided and soft sectorred and stored only 160K bytes.

Since 1981 IBM has increased the disk storage capacity and has added hard disks with 10 and more megabyte storage capacity.

Since 1984 the Intel microprocessor 80286 is used in the advanced personal computer PC/AT. It is a true 16-bit microprocessor that uses a full 16-bit data bus and is able to allow multitasking and virtual memory storage. In 1986 the 32-bit Intel microprocessor 80386 was introduced.

The enormously increased efficiency concerning computing time, available disk space and the simultaneously drop down of costs have made the IBM PC an interesting computer system to use for both scientific research and engineering applications.

Architecture of the IBM-PC:

Hardware : microprocessor: Intel 8088,80286,80386
 math coprocessor: Intel 8087,80287,80387

 support chips: 8259 interrupt controller
 8237A DMA controller
 8284A clock generator
 8255 peripheral interface

 address bus: transmits addresses
 data bus: transmits data

 memory : ROM (permanent) : BIOS
 BASIC
 DIAGNOSTICS
 DISPLAY MEMORY

 RAM : up to 640K bytes addressable

Requirements to FORTRAN compilers for PCs:

The success and significance of the PC family has made the development of programs for it or the transfer of already available programs to it very important. Although other high-level languages as for example C are available and recommendable due to their structural capabilities, the programming language FORTRAN has the most widespread use for engineering applications. Especially programs, developed on other computer systems and designed for easy transfer between different systems, have to stick to FORTRAN because of its strict standard. The main requests to a Fortran compiler used on PCs are:

For easier programming, easier transfer of programs and for portability Fortran compilers should cover the full implementation of ANSI standard Fortran-77 designated ANSI X3.9-1978.

Due to the restricted memory size of 640K bytes addressable memory under the MS-DOS operating system, the transfer of very big programs request for a loader providing the possibility of overlays, especially of overlays in more than one level.

A Fortran compiler configured for the DOS operating system should support DOS-device names and DOS-path names.

Available Fortran compilers:

The author started in December 1985 to transfer the block adjustment programs PATM and PATB to IBM compatible personal computers. At that time mainly three different FORTRAN compilers have been available:

1. Digital Research Fortran-77, Version 4.1 : DR
2. Microsoft Fortran 77, Version 3.20 : MS
3. Ryan-McFarland Fortran-Compiler, Version 1.0 : RM

At that time only DR and RM covered the full implementation of ANSI standard Fortran-77 designated ANSI X3.9-1978. MS conformed only to the ANSI X3.9-1978 subset Fortran requirements. For example it did not allow substrings of character variables and did not include the full capabilities of some I/O statements like OPEN or INQUIRE.

RM Fortran did not support any overlay. With MS Fortran overlays have been only possible in one level. DR Fortran was supposed to accept overlays in several levels, but there was obviously a bug within the overlay-manager which stopped execution of the program immediately running under DOS 3.0 and later versions when the first overlay was called.

Path names of files have not been accepted for DR and MS Fortran during compilation and have not been accepted during linking of programs by the DR linkage editor, but all three accepted already DOS-path names during execution of the programs.

The size of the generated load modules differed remarkable. The load module generated by DR Fortran was about 60% bigger than the one generated by RM Fortran, and due to the bug within the overlay linkage editor it was too big for execution with 640K bytes of memory.

Meanwhile new version of all three Fortran compilers are available covering the full implementation of Fortran-77 and all three provide linkage editors to enable nested overlays in the form of a tree structure with multiple levels.

Benchmark tests:

There have been many discussions about benchmark tests. It is quite clear that a specific benchmark test, which don't take into consideration all other things, leads to restricted statements. But the problem of block adjustment is a typical Fortran application, a lot of number crunching, matrix operations and a lot of input and output operations for the intermediate storage of results. Therefore a matrix multiplication of 2 matrices of the size 100 x 100 and a program for writing and reading unformatted data to the hard disk was chosen for the test. The following tables show the results:

Multiplication of two 100 x 100 matrices (double precision):

compiler:	time in sec on XT (ratio)	time in sec on AT (ratio)
DR-F77	552.7 (8.0)	124.1 (1.8)
MS-FOR	276.3 (3.9)	77.4 (1.1)
RM-FOR	200.4 (2.8)	70.8 (1.0)

The results of the test can't be considered as absolute values. One has to take into account that the time for matrix multiplications for the AT depends to a high degree on the frequency of the math co-processor. For the test an AT with a 6-MHZ co-processor was used. With the nowadays available 20-MHZ 32-bit co-processors the execution time on the PC 386 will be about 5 times faster and therefore easily reach the speed of minicomputers like the HARRIS 100 or the MV8000 and the SUN 3/75 and not far from others like the MICROVAX II and the HP 900.

I/O of unformatted read and write statements to hard disk:

compiler:	Byte/sec on XT (ratio)	Byte/sec on AT (ratio)
DR-F77	1005 (1.0)	3020 (3.0)
MS-FOR	4475 (4.5)	11850 (11.8)
RM-FOR	5490 (5.5)	15600 (15.5)

Comparing the transfer of data, one has to take into account that the 40 MByte hard disk used with the PC/AT is about twice as fast than the 10 MByte hard disk of the PC/XT.

Actual transfer of the programs to PC:

The benchmark tests have shown that the Ryan McFarland Fortran compiler seems to be the fastest for execution of this type of arithmetic operations and for data transfer to hard disk required for solving problems with this big amount of data as it is with block adjustment. Therefore the Ryan McFarland Fortran compiler was used for the actual transfer of the mainframe versions PAT-MR and PATB-RS to IBM compatible personal computers.

The first step was the adaptation of the programs to strict ANSI standard Fortran-77 designated ANSI X3.9-1978. Besides

this some more modifications, with respect to peculiarities of PC systems, have been made to optimize the execution time of the CPU and the transfer of data to hard disk. The resulting PC versions are named in the following as PATM-PC and PATB-PC.

Use of the BIOS services:

The wisest approach for programming normally is to write all programs in a high-level language. But PC users are different people than mainframe users, they request for better and more comfortable user interfaces to communicate with the programs. These user interfaces deal with input from keyboard, input and output from and to the display device and all this in a comfortable and to avoid frustration very fast way.

The Fortran language does not provide those features. We have to create our own assembly-language routines to perform these special tasks. A subroutine can be called in one of the two ways: through a CALL instruction or through an interrupt. The BIOS services are called through interrupts using the INT instruction. The addresses of the service routines are implicit in the interrupt number.

Conceptually the BIOS programs lie between our programs (including the operation system) and the hardware. That means using the BIOS services we are independent, on one side from the hardware such as the display device and on the other side we work below the operating system, thus we are also independent from the operating system.

The ROM-BIOS video services are all requested by generating interrupt 16 (hex 20). The most important services are:

```
service 2 : set cursor position
service 3 : read cursor position
service 8 : read character and attribute
service 9 : write character and attribute
```

The keyboard services are invoked with interrupt 22 (hex 16) and are mainly:

```
service 0 : read next keyboard character
service 1 : report whether character ready
```

With these BIOS services we have in fact a very fast and direct access to screen and keyboard, very useful for programming interactive menus and for the display of information during program execution.

The MENU programs for PATM and PATB:

Principally a menu is a tool for editing special data necessary for execution of a special program. The PAT menu is a program that's easy to learn and use. You can start using it immediately, without having to study a manual.

As soon as you start the PAT program, the MAIN menu page is displayed on the screen. From here you can either invoke the PATM or the PATB program but first you have to pass the PATM-MENU respectively the PATB-MENU to define the basic data

necessary for execution of them.

Both the PATM-MENU and the PATB-MENU consists of ten different menu pages. The structure of the menu is primarily based on the use of the F10 function keys available on any keyboard. Each menu page is connected with a specific key word and a specific function key. The ten key words tied with the related function key numbers permanently appear in reverse video in the bottom line of the screen throughout the menu session. The key words will always remember you in which menu page you can find the data you like to change. Thus you can easily switch from the present to any other menu page just by pressing the respective function key.

The key word of the present menu page always appears in the heading line of the menu page and is substituted in the bottom line by the word "help" indicating that you can invoke the help function by pressing the respective function key.

Here is a quick summary of the function keys and their key words for both the PATM-MENU and the PATB-MENU:

PATM-MENU:

F1	-	MAIN
F2	-	FILES
F3	-	INIT
F4	-	GROUP
F5	-	FORMAT
F6	-	PRINT
F7	-	STDEVM
F8	-	STDEVC
F9	-	LIMITS
F10	-	AUX

PATB-MENU:

F1	-	MAIN
F2	-	FILES
F3	-	INIT
F4	-	GROUP
F5	-	FORMAT
F6	-	PRINT
F7	-	STDEVP
F8	-	STDEVC
F9	-	LIMITS
F10	-	GPS

Within each menu page you can enter or edit certain data, explained by a short text, within specified input fields. The cursor can be moved within the input fields and from one to another input field by pressing the cursor keys normally located on the right side of the keyboard.

Here is how the cursor can be moved:

Right arrow	-	right one character or next input field
Left arrow	-	left one character or previous input field
Up arrow	-	previous input field
Down arrow	-	next input field
Home	-	first character of input field
End	-	last character of input field
Pg Up	-	first input field of page
Pg Dn	-	last input field of page

The Right and Left arrow keys act slightly differently, depending upon where the cursor is. Right arrow moves the cursor one character to the right, if the cursor is already located in the rightmost position it moves to the next input field. The Enter key acts the same as the Down arrow key. The input field where the cursor is located changes always to reverse video.

Here are the two commands that delete text:

Del - deletes character under cursor
Backspace - deletes one character to the left

By pressing the Ins-key you toggle the menu between insert and replace (overstrike) mode. The actual mode is indicated in the status line located just below the present menu page.

When the cursor is positioned within the input field of data which allow only fixed answers for instance YES or NO, all keys besides the Ins-key are blocked and the Ins-key can be used to toggle between the answers. For this data the status line shows the toggle function of the Ins-key.

As soon as the cursor leaves an input field, the entered data are checked for all possible errors and corresponding error messages appear on the screen. As long as the data are not corrected, you can't force the cursor to leave the field.

Display during execution:

In opposite to users of mainframe systems, PC users like to control their programs during execution. For this reason a very informative display was added to both block adjustment programs. This display shows information about the input data and reports the actual status of execution by recording the different program steps, the iteration number, the maximum change of coordinates between the iterations and the actual estimated sigma naught. In case of requested automatic elimination of blunders, the present number of rejected observations is displayed.

Thus the user is able to judge already during execution about the quality and precision of the observations and is able to predict the furtheron required number of iteration steps and execution time.

Features of the programs PATM-PC and PATB-PC:

PATM-PC is a multi-purpose program for either spatial or purely planimetric block adjustment with independent models and is well suited for all topographic and engineering mapping projects.

As input PATM-PC allows observations from model and object space and their a priori standard deviations such as model coordinates, perspective centers and horizontal and vertical control point coordinates. In addition auxilliary height observations such as statorscope data, APR data and shore line information can be used in the simultaneous adjustment.

The program automatically alternates between horizontal and vertical iteration steps in the most efficient way depending on initial orientation and the behaviour with respect of existing gross data errors and provides a simultaneous absolute orientation of all independent models.

PATB-PC is a general purpose program for photogrammetric point determination and simultaneous image orientation and is well-proven for highest accuracy specifications. It can be used for all types of aerial triangulation but is particularly

suited for point determination and frequently applied for cadastral surveying, control of ground movements and other high precision photogrammetry.

Self-calibration is used to correct simultaneously for systematic image deformations. Different sets of self-calibration parameters can be applied to ensure an optimum correction and the discovered image deformations are supplied for easier interpretation of their reasons.

Both programs include a procedure for automatic localization and elimination of gross data errors using the method of robust estimators. It relieves of tedious and difficult error analysis and successive adjustments.

All required approximate values of the unknowns are calculated directly from the data to avoid preceding manual preparation, but if approximate orientation parameters are available they can be used to save initial iteration steps.

There are no restrictions concerning overlap and number of points. Optimization algorithms for minimization of storage requirements and computing time provide fastest execution of the adjustments. Detailed information about block statistics and accuracy results simplify the quality control.

Both programs retain not only all features of the mainframe versions but also provide a comfortable menu guided input and an informative display during execution.

Block size and execution time:

The block size is practically unrestricted. PATM-PC can handle blocks with up to 1000 models, PATB-PC allows the adjustment of blocks with up to 500 images. The only restrictions are related to the hardware configuration such as processor speed and available disk capacity.

The following tables show the execution time and the required disk capacity for various block sizes and execution on the PC/AT with 10MHZ coprocessor for block adjustments without automatic error detection:

PATM-PC :

block size	execution time	disc space
200 models	1h10m	3 MB
300 models	1h50m	5 MB
500 models	3h30m	10 MB
1000 models	11h30m	25 MB

PATB-PC :

block size	execution time	disc space
200 images	5h50m	6 MB
300 images	7h30m	10 MB
500 images	17h30m	20 MB

The tables proof the capability of personal computers. However for bigger blocks, especially when applying the feature of automatic elimination of blunders, it is advisable to use the faster systems with 80386/80387 processors and at least 16 MHZ frequency which will speed up execution by a factor 2

Future developments of the hardware:

Three main components determine the execution speed of computers:

1. The speed of the processors mainly determined by the frequency and their architecture.
2. The possible access time of the memory chips.
3. The access time of hard disks.

Up to now the 386 microprocessors work with frequencies up to 20 MHZ, but some microprocessors with the capability of 25 MHZ are already offered, which indeed means up to 4 million instructions per second (4MIPS).

At the time the math coprocessor 80387 is used with the faster systems. A new arithmetic coprocessor, the 1167 processor produced by Weitek, is already offered and some main boards are already prepared for its use. It will improve the efficiency of arithmetic operations by a factor 5 as soon as Fortran compilers are available which make use of it.

Normally dynamic RAMs are used as memory chips. By reading information from the chip it will be a little bit discharged and between 100 and 500 Nanoseconds are necessary to recharge it again for further use. Static RAMs don't need this refresh but are much more expensive, but can be used as so-called cache memory to speed up the access with reasonable increase of costs.

Multiple BUS architecture as used already by COMPAQ within its Flex architecture or the microchannel used by IBM for the PS/2 systems will speed up the transfer of data between the different components of the system.

An other possibility is the use of more advanced hard disk controllers as for instance the already available Enhanced Small Device Interface (ESDI) controller for faster access and transfer of data to hard disks providing an average access time of 16 milliseconds only.

Conclusions:

The low cost personal computers provide a cheap solution for solving block adjustment. Already at the time being they can handle most practical applications. The expected further development will increase their efficiency again and there will be no need anymore to use more expensive systems for all nowadays available application programs.