HARDWARE AND SOFTWARE IMPLEMENTATION OF AN IMAGE TRACKING SYSTEM USING VIDEO BLASTER AND FRAME DIFFERENCE WINDOW TECHNIQUE

S. GHONIEMY[*], S.F. BAHGAT[**], G. KASSEM[***]
R. N. MOHAMED[***]

Abstract:

Active target tracking systems always suffer from jamming conditions which disturb the proper behavior of their control systems. One of the most effective methods to avoid jamming environment is the TV image tracking of moving targets. TV image tracking is carried out by manually selecting the target to be tracked and then creating proper control signals to the camera via the interface (control unit) to ensure accurate target tracking.

This paper proposes a practical TV image tracking system based on Video Blaster as a tool for detecting the target image and converting it into a format suitable for computer processing . In addition, the frame difference window technique is introduced. In this technique, the tracking window is automatically positioned at a predicted target position and the center of this window is considered as the center of the intended target. The hardware interface between the camera and the computer is presented.

It was found that, the proposed system overcomes most of the disadvantages of the traditional TV tracking systems, being more stable and reliable.

I. Introduction:

Recent wars have proved the effectiveness of optical sensors (TV cameras & IR sensors) when employing electronic countermeasures against radar sensors.Computer analysis of time varying images implies the processing of a time sequence of images in order to extract some properties which change with time. There are two possible approaches to extract motion information: where the moving objects themselves are of interest (tracker typify such system), and where the motion information is used to provide additional features to scene analysis system. In this paper, a general discription of TV tracking systems based on Video Blaster cards and frame difference window technique is introduced in part II. The hardware, implementation of the proposed system is presented in part III. Part IV introduces the algorithms neccessary for software implementation. The design and implementation of a practical hardware interface between the computer and camera control system is proposed.

---

*   Professor, Faculty of Computer Science & Information Systems, Ain Shams University.
**   Associate Professor, Military Technical College, Cairo, Egypt.
*** Post Graduate Student, Military Technical College, Cairo, Egypt

II. Proposed TV image tracking system :

As shown in Fig.1, the conventional TV image tracking system based on using Video Blaster unit is composed of the following units: video camera, TV monitor, Video Blaster, camera driving system, joy stick for manual control, and PC computer.
The system functions as follows:

The camera provides the life image frame through the TV monitor to the Video Blaster. The Video Blaster converts the life image frame into a format suitable for display on PC monitor. The system has the ability to store the life image in a file on the hard disk. The camera can be controlled manually by using joy stick control unit, i.e. this system can track a moving target manually only.

According to Fig.2, The proposed TV image tracking system consists of the following units: video camera, TV monitor, Video Blaster card, proposed hardware interface, camera driving system, a software package for TV image tracking, a microcomputer, joy stick for manual control, and video tape recorder.

The function of the proposed system is as follows:-

The video camera provides a real - time analog image to the proposed system which operates in one of two modes : searching mode and tracking mode .

In searching mode , the camera provides the life image frame to the video blaster through the TV monitor. Video blaster receives the life image and changes it into a form suitable for displaying on the PC monitor. The objective of the proposed software is to recognize the image format provided by the Video Blaster and convert it into a format suitable for microcomputer operation.
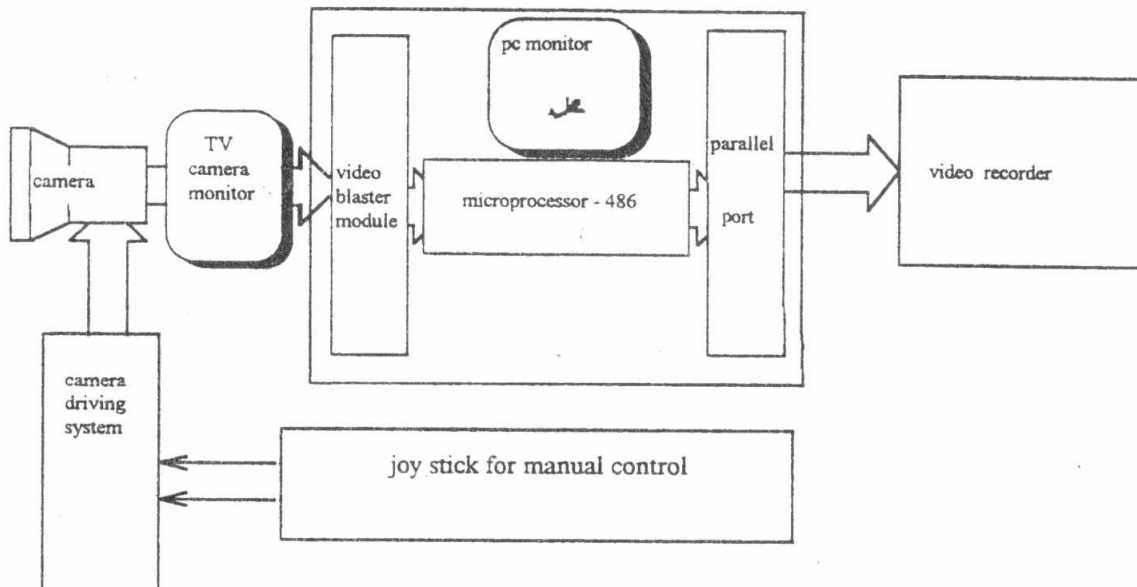
Fig.1 Block diagram of a conventional TV image tracking system
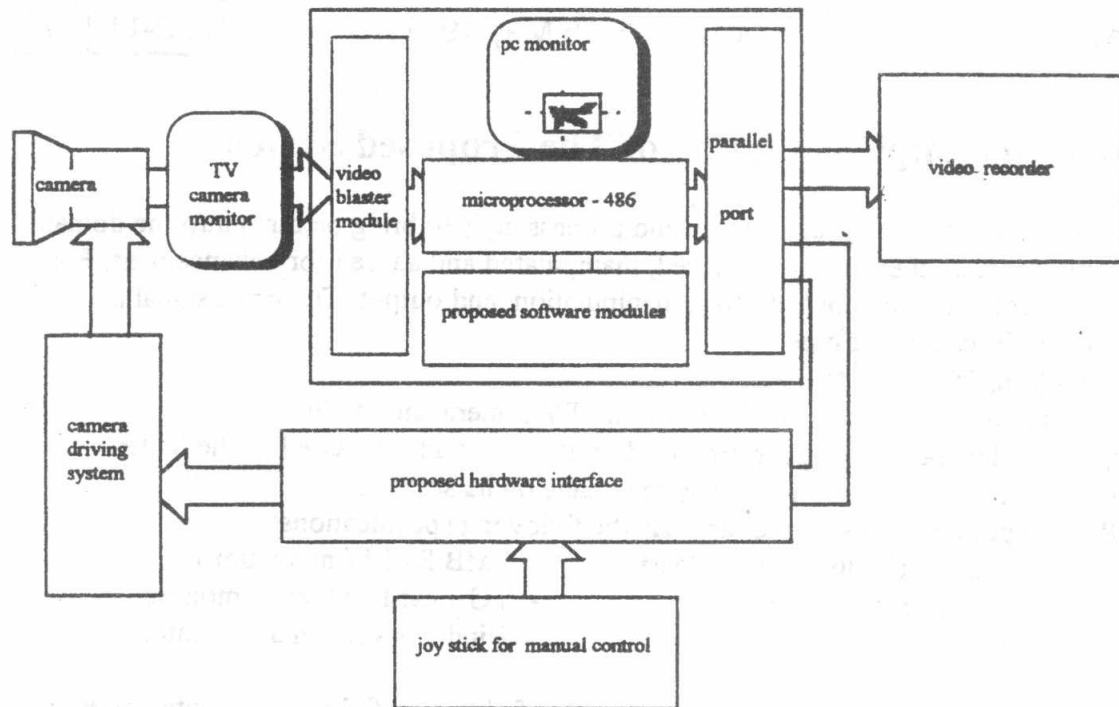
Fig.2 Block diagram of proposed TV image tracking system

The proposed software includes:
1- Programs responsable for controlling the mode of operation of Vide Blaster.
2- Programs for reading the header file developed by Video Blaster.
3- Programs for converting Video Blaster format into a format suitable for computer processing .
4- Programs for generating tracking window.
5- Programs for generating the frame difference of the two fields of view.
6- Programs for calculating the coordinates of the field of view of the image frame and the center of the tracking window.
7- Programs for calculating the difference of the measured coordinates and changing them into an error voltage to control the camera driving system.

The microcomputer starts execution by calculating the center of the image frame and the center of the tracking window including the target of interest. The difference between the two centers is calculated and transmitted through the parallel port to the proposed hardware interface . The output from the proposed interface is sent to the camera driving system to align the center of image frame with the center of the tracking window .

In tracking mode: the proposed software produces the required control signals to control the mode of Video Blaster to update the image file and repeat the process starting from determination of the center of the new position of the tracking window and producing an error signals associated with the new position of the moving target for the camera driving system to follow the intended target.

III. Hardware Implementation of The Proposed System:

The general goal for image acquisition and processing is to bring pictures into the domain of the computer, where they can be displayed, manipulated and altered for enhancement. Four processes are involved here: input, display, manipulation, and output. The input signal is captured with a TV camera of type hs6Zm2-5, 8-48 m, 1: 1.42cs, tc 9948a.

The TV monitor is used as a link between the TV camera and the PC computer, where the real life image is displayed on its screen and is passed to the Video Blaster card via the PC computer to display the image on its screen .

The PC computer used for processing has the following specifications:-
- IBM computer with CPU 386 or 486 at least.   - 4 MB RAM (minimum).
- 40 MB free hard disk space (minimum).   - VGA card and VGA monitor.
- DOS version 3.3 or later.   - Windows version 3.1 or later.

With Video Blaster Kit, the system is capable of obtaining full motion digital video, as well as capturing and manipulating video images. It is overlays computer text and graphics onto video and even scale the size of the video window according to our needs.

The activities of the Video Blaster include:
- File formating supported by Video Blaster.
- Saving and restoring an image. The command VBSEVIEW.EXE allows to capture a life video image and save it as a graphics file. It also allows to convert this file into another format and display it on the VGA monitor. The operation of Video Blaster can be controlled from the DOS commands line.
- Digital image representation.

III. 1 Microcomputer-based motion control system :

The used TV camera [hs6Zme-5] is equipped with a manual control system for controlling the camera motion, zooming, and focusing. The control box is equipped with a joy-stick for controlling the camera manually in both horizontal and vertical planes. However, a microcomputer-based motion control system is built and implemented according to Fig.3.
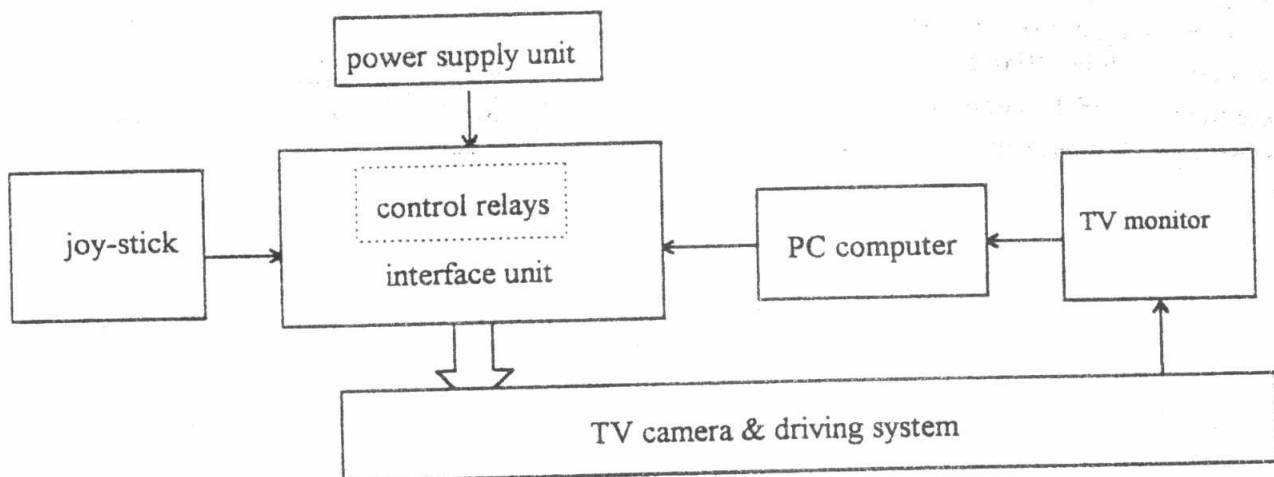
III.2 Interface (control) unit:

The interface is designed to control the camera either automatically through the PC computer or manually using joy-stick. The interface unit consists of the following elements (components) : 8 Transistors 2N2222, 8 Leds (indication lamps) connected in series with 8 resistances (1 kΩ), 8 Diods (to protect the transistors from damage), 8 Relays (12 vdc), 3 Relays (30 vdc). Fig.4 shows the block diagram of the interface unit The interconnection of this unit with the other units is shown in Fig.5.

The interface unit has the following Input/output signals:

1- Up and down control signals (vertical motion).

2- Left and right control signals (horizontal motion).

3- Zoom (in) and Zoom (out) controlling signal.

4- Focus (far) and Focus (near) controlling signal.

The interface unit can operate in two modes, automatical or manual. These modes can be controlled by an external switch.
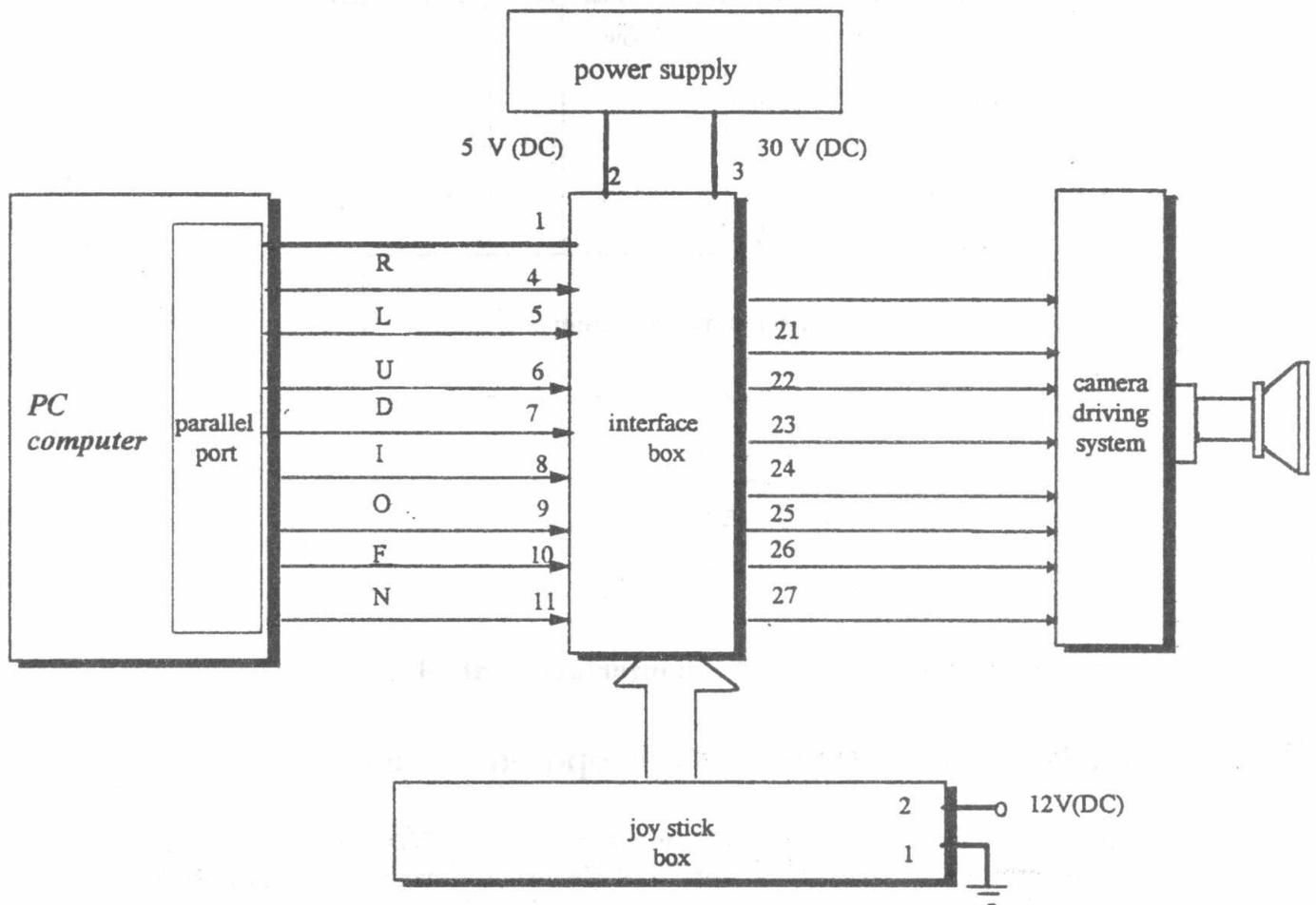
Fig.4 The block diagram of the interface control unit.
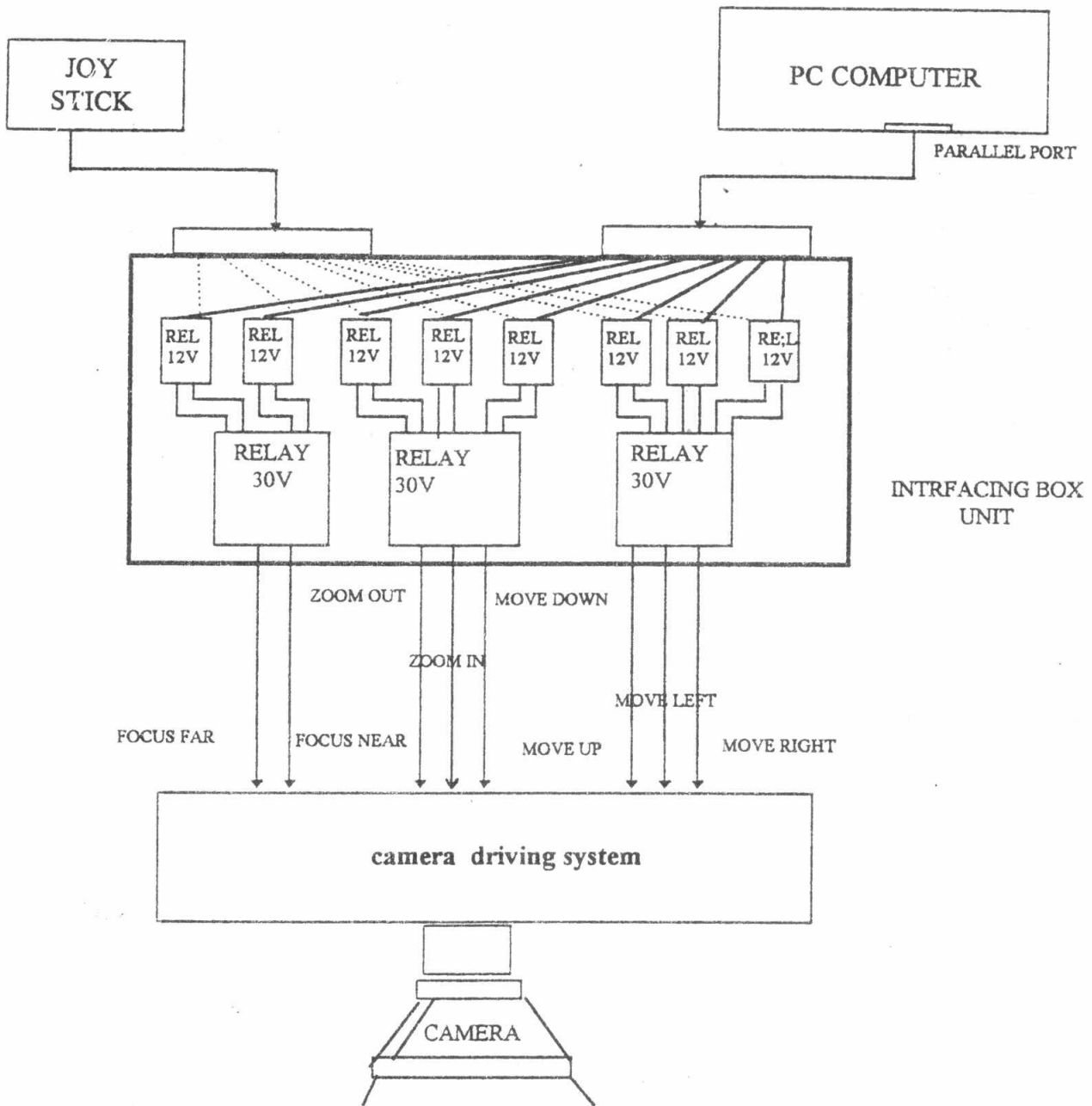
Fig.5 The circuit diagram of the interface control unit.

IV. Software Implementaion Of The Proposed System :

The software is implemented using C- language and the .PCX format. The .PCX format is useful for programs that read or write PCX file images. Image files used by .PCX format begin with a 128 byte header which consists of header1 from bytes 0 to 15, palette information from bytes 16 to 63, header2 from bytes 64 to 73, and filler from bytes 74 to 128 (set all zeros).

IV.1 Reading the header of the image file:

Using a source file called PCXREAD.C file, the header file of the image can be read.. The flowchart for reading the header of the image file is shown in Fig.6.
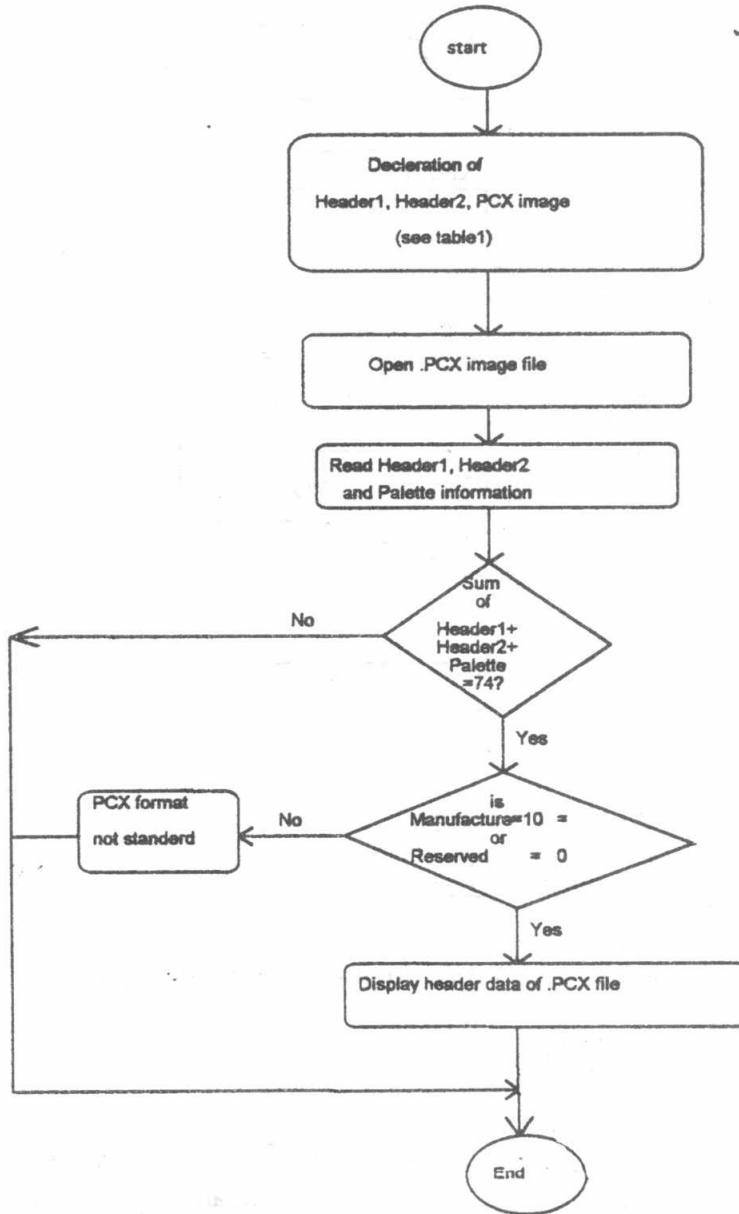
Fig. 6 The flowchart for reading the header of the image file

IV.2 Displaying the image on the screen:

The flowchart for displaying the image on the screan is shown in Fig.7.

```
                          ┌──────────┐
                          │  START   │
                          └────┬─────┘
                               │
                   ┌───────────▼────────────┐
                   │  open image file (.PCX) │
                   │   and open data file    │
                   └───────────┬─────────────┘
                               │
                   ┌───────────▼────────────┐
          ┌───────►│    Jx = Xmax-Xmin+1     │
          │        └───────────┬─────────────┘
          │                    │
   NO     │              ╱─────▼─────╲
          └─────────────◄   Jx >640   ►
                         ╲─────┬─────╱
                          YES  │
                   ┌───────────▼────────────┐
          ┌───────►│    Iy = Ymax-Ymin+1     │
          │        └───────────┬─────────────┘
          │                    │            NO
          │              ╱─────▼─────╲
          └─────────────◄   Iy>480    ►──────┘
                         ╲─────┬─────╱
                          YES  │
                   ┌───────────▼──────────────┐
                   │  Jx = 640 and  Iy = 480   │
                   │     and display image     │
                   └───────────┬───────────────┘
                               │
                          ┌────▼─────┐
                          │   end    │
                          └──────────┘
```
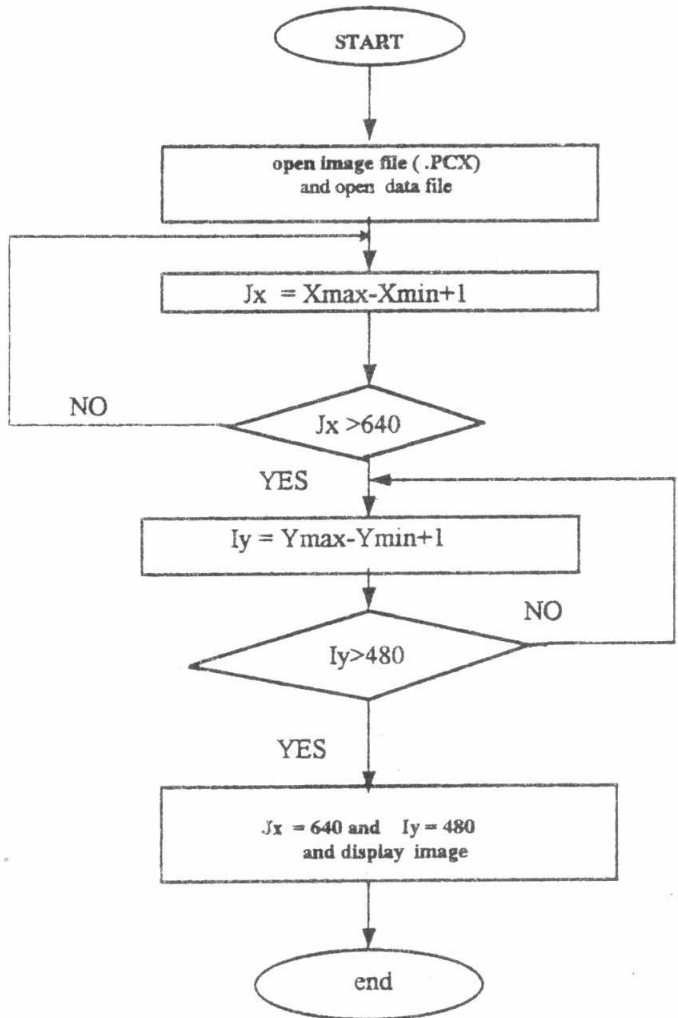
Fig.7 The flowchart for displaying the image on the screan

IV.3 Manual control of the camera :

The aiming angle of the camera can be adjusted through the arrows of the keyboard (right, left, up, and down). In addition, camera zooming (in, and out), and focusing (near, and far) can be carried. The corresponding flowchart is shown in Fig.8.

IV.4 Automatic control of the camera towards a simulated target:

Using a source file called MCAMTGT1.C file, the camera can move automatically towards the simulated target according to the difference between the center of the target and the center of the camera (which is the center of the screen) . Fig:9 shows the flowchart of automatic directioning of the camera towards a simulated target.
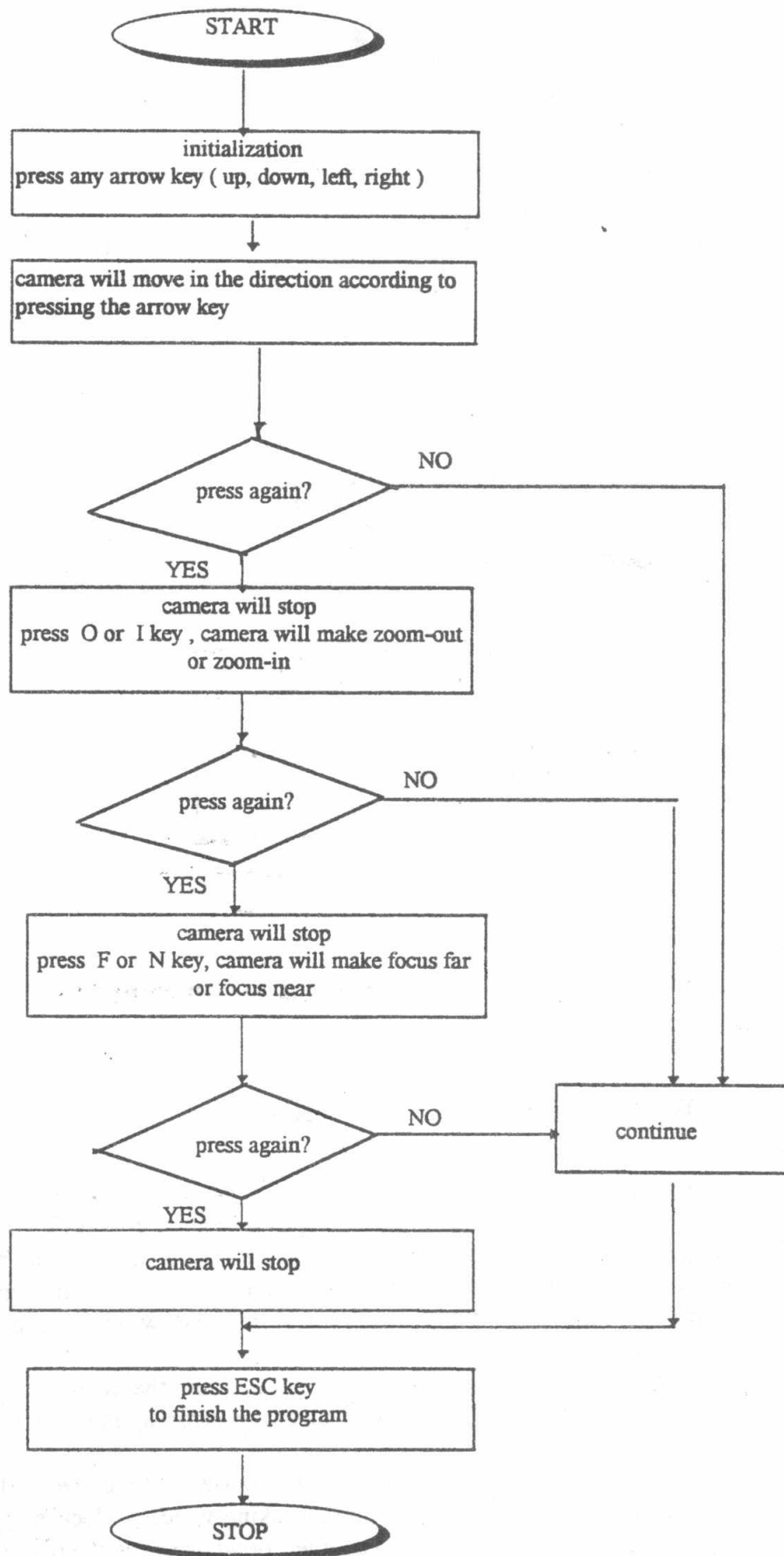
Fig.8Flowchart for controlling the camera position manually.

```
                          ┌─────────────┐
                          │    START    │
                          └──────┬──────┘
                                 │
                   ┌─────────────┴─────────────┐
                   │ initialization and open    │
                   │ the data file              │
                   └─────────────┬─────────────┘
                                 │
                            ◇ data file exist? ◇ ──── NO ────┐
                                 │                            │
                               YES                            │
                   ┌─────────────┴─────────────┐              │
                   │      display the life image,│             │
                   │ creat the scan window and expand it to │   │
                   │ include the intended target │             │
                   └─────────────┬─────────────┘              │
                                 │                            │
                   ┌─────────────┴─────────────┐              │
                   │ calculate the centers of the window and │ │
                   │ the camera. and calculate the error     │ │
                   └─────────────┬─────────────┘              │
                                 │                            │
                            ◇ error exist? ◇ ──── NO ────┐    │
                                 │                        │    │
                               YES                        │    │
                   ┌─────────────┴─────────────┐          │    │
                   │      align the camera      │          │    │
                   └─────────────┬─────────────┘          │    │
                                 │◄──────────────────────┘    │
                                 │◄───────────────────────────┘
                          ┌──────┴──────┐
                          │    STOP     │
                          └─────────────┘
```
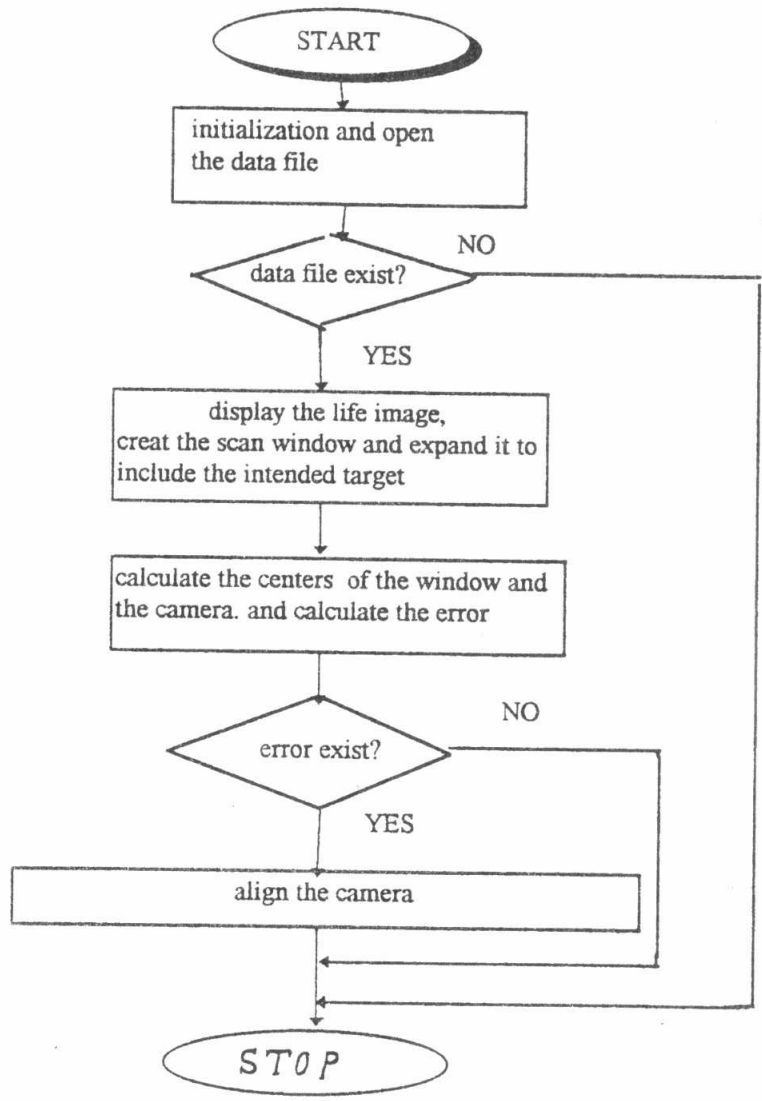
Fig. 9 Flowchart of automatic directioning of te camera towards a target

IV.5 Scanning of the tracking window:

In real time image tracking systems, images can be captured by using the Video Blaster card, and stored in a data file that can be read and displayed on the computer screen.

An initial scanning window is used . To search for the intended target, the pixels inside this scanning window can be tested to check if there is any change in the background-pixels. The scanning operation continues till capturing a target and window will be expanded to contain the whole image of the target inside it.

The difference between the center of the screen (camera) and the center of the target (window) can be calculated. This difference is considered as error signals that act to control the tracking motion of the camera.

The camera moves continuosly towards the target in horizontal then vertical planes until the error between the center of field of view and the tracking window decays to zero.

Figures 10,10(a),10(b), and 10(c) show the detailed flowcharts for the real time tracking process using scan-window technique and Video Blaster card.

START

INTIALIZE GRAPHIC
MODE

OPEN THE TARGET FILE (.DAT)
DISPLAY THE TARGET FILE(.PCX)

1

CREAT THE SCANNING WINDOW
(X1,Y1,X2,Y2)

2

is
color of pixel insidethe
scanning window
/=0

NO   YES

DELETE SCAN WINDOW
(X1,Y1,X2,Y2)   3

                              EXPAND THE SCAN WINDOW TO CONTAIN
NO   ALL THE TARGET .
        X2>=639   SWITCH-STATEMENT :
                               CASE(0) :MOVE RIGHT
              YES   CASE (1) :MOPVE DOWN
                               CASE(2) : MOVE UP
X1=0   X2=X2-X1   CASE (3) :MOVE LEFT

Y1=Y1+1   Y2=Y1+=10
                                    4

                              CASE (4) :CALCULATION OF THE ERROR
                          THE CENTER OF THE SCREEN IS :(GETMAXX()/2 , GETMAXY()/2)
YES   THE CENTER OF THE WINDOW IS :(X1+X2)/2 , (Y1+Y2)/2)
        Y2<=479   THE DIFFERENCE BETWEEN THE TWO CENTERS IS :
                          (GETMAXX()/2- (X1+X2)/2 , GETMAXY()/2 - (Y1+Y2)/2)
              NO   START THE TRACKING PROCESS

                                    5

STOP

Fig. 10 Real time tracking process by using scanning window technique

```
                        ( 1 )
                          |
              ┌───────────────────────┐
              │     FOR : X = X1       │
              └───────────────────────┘
                          |
        ┌─────────────────┘
        |         ┌───────────────────────┐
        |         │    GETPIXEL(X,Y1)      │
  ┌──────────┐    │    GETPIXEL(X,Y2)      │
  │  X = X+1 │    └───────────────────────┘
  └──────────┘               |
        |         ┌───────────────────────┐
        |         │   SETPIXEL(X,Y1,15)    │
        |         │   SETPIXEL(X,Y2,15)    │
        |         └───────────────────────┘
        |                    |
        |                ╱ X <= X2 ╲ ── YES
        └──────────────  ╲        ╱
                            ╲    ╱
                             NO
                              |
              ┌───────────────────────┐
              │     FOR : Y = Y1       │
              └───────────────────────┘
                          |
        ┌─────────────────┘
        |         ┌───────────────────────┐
        |         │    GETPIXEL(X1,Y)      │
  ┌──────────┐    │    GETPIXEL(X2,Y)      │
  │ Y = Y + 1│    └───────────────────────┘
  └──────────┘               |
        |         ┌───────────────────────┐
        |         │   SETPIXEL(X1,Y,15)    │
        |         │   SETPIXEL(X2,Y,15)    │
        |         └───────────────────────┘
        |                    |
        |            YES ── ╱ Y <= Y2 ╲
        └──────────────    ╲        ╱
                             ╲    ╱  NO
                              |
                            ( 2 )
```
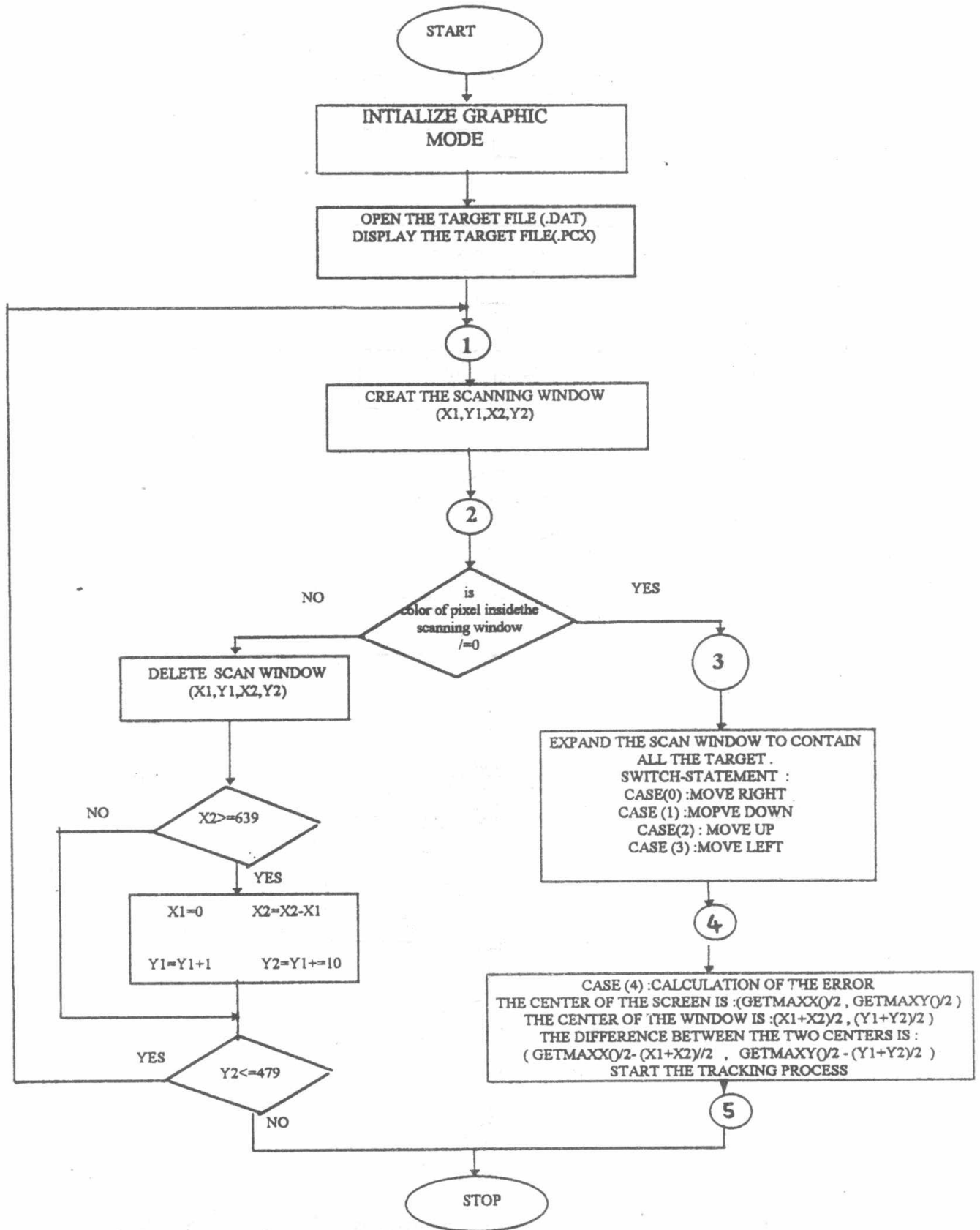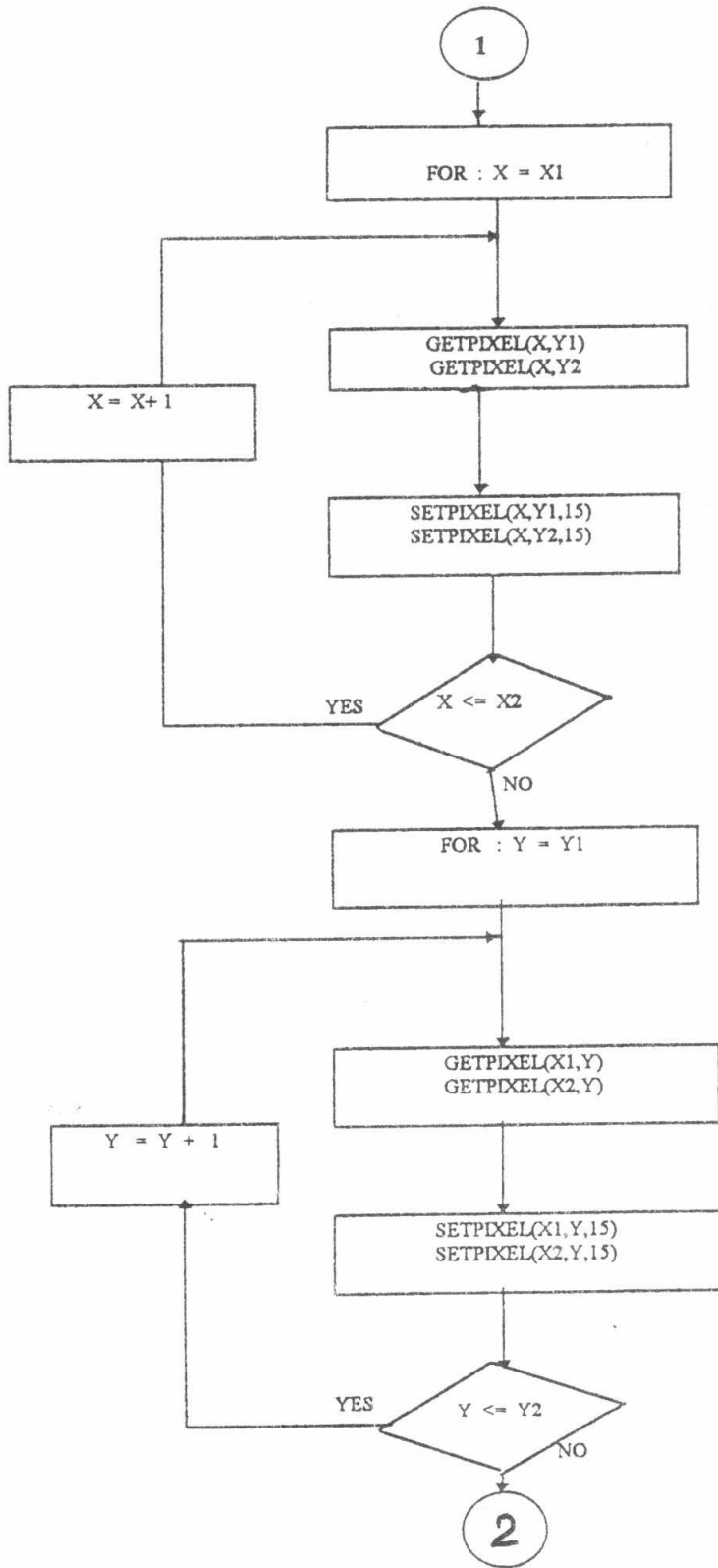
Fig .10(a) The flowchart of creating the scanning window on the screen
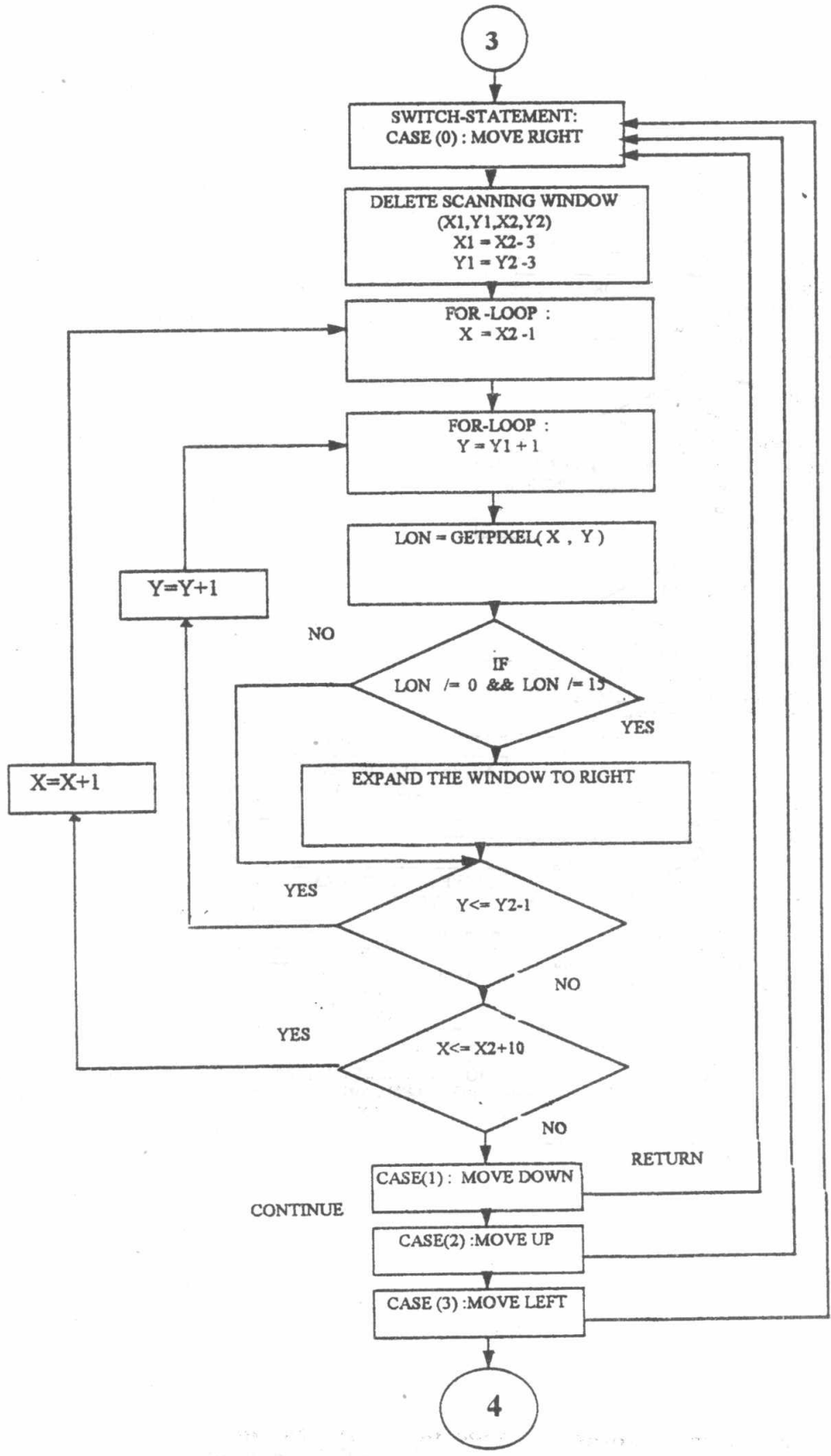
Fig. 10(b) The flowchart of the target capture

```
                              ( 4 )

          NO                    |
    ┌──────────────<  FLAGE=0  >
    │                    |     YES
    │              ┌─────────────────────────────────────┐
    │              │ SET THE IMAGE TO THE BACK COLOR AND THE │
    │              │     BACKGROUND TO WHITE                 │
    │              │          FLAG=1                         │
    │              └─────────────────────────────────────┘
    │                    |
    │         ┌──────<  FLAG = 1  >
    │         │          |       YES
    │         │   ┌─────────────────────────────────────┐
    │         │   │ CENTER OF THE, WINDOW, CENTER OF THE SCREEN │
    │         │   │ THE DIFFERENCES BETWEEN THE TWO CENTERS     │
    │         │   │          FLAGE = 2                          │
    │         │   └─────────────────────────────────────┘
    │         │         |
    │    NO   │   ┌──<  FLAGE = 2  >
    │         │   │     |       YES
    │         │   │   ┌─────────────────────────────────────┐
    │         │   │   │ CALCULATIONS OF THE CENTER OF THE :  │
    │         │   │   │ WINDOW (Xw, Yw)   SCREEN ( Xs, Ys )  │
    │         │   │   │ THE DIFFERENCES (Xsw , Ysw )         │
    │         │   │   └─────────────────────────────────────┘
    │         │   │         |          YES    ┌──────────────────────┐
    │         │   │      < Xsw>0 >─────────────│ MOVE LEFT OF CAMERA  │
    │         │   │         | NO               │ VLEFT(Xsw)           │
    │         │   │         |          YES     └──────────────────────┘
    │         │   │      < Xsw<O >─────────────│ MOVE RIGHT OF CAMERA │
    │         │   │         | NO               │ VRIGHT(-Xsw)         │
    │         │   │         |          YES     └──────────────────────┘
    │         │   │      < Ysw>O >─────────────│ MOVE UP OF CAMERA    │
    │         │   │         | NO               │ VUP(Ysw)             │
    │         │   │         |          YES     └──────────────────────┘
    │         │   │      < Ysw<O >─────────────│ MOVE DOWN OF CAMERA  │
    │         │   │         | NO               │ VDOWN(-Ysw)          │
    │         │   │   ┌──────────────┐         └──────────────────────┘
    │         │   │   │   FLAG = 4   │
    │         │   │   └──────────────┘
    │                    |
                       ( 5 )
```
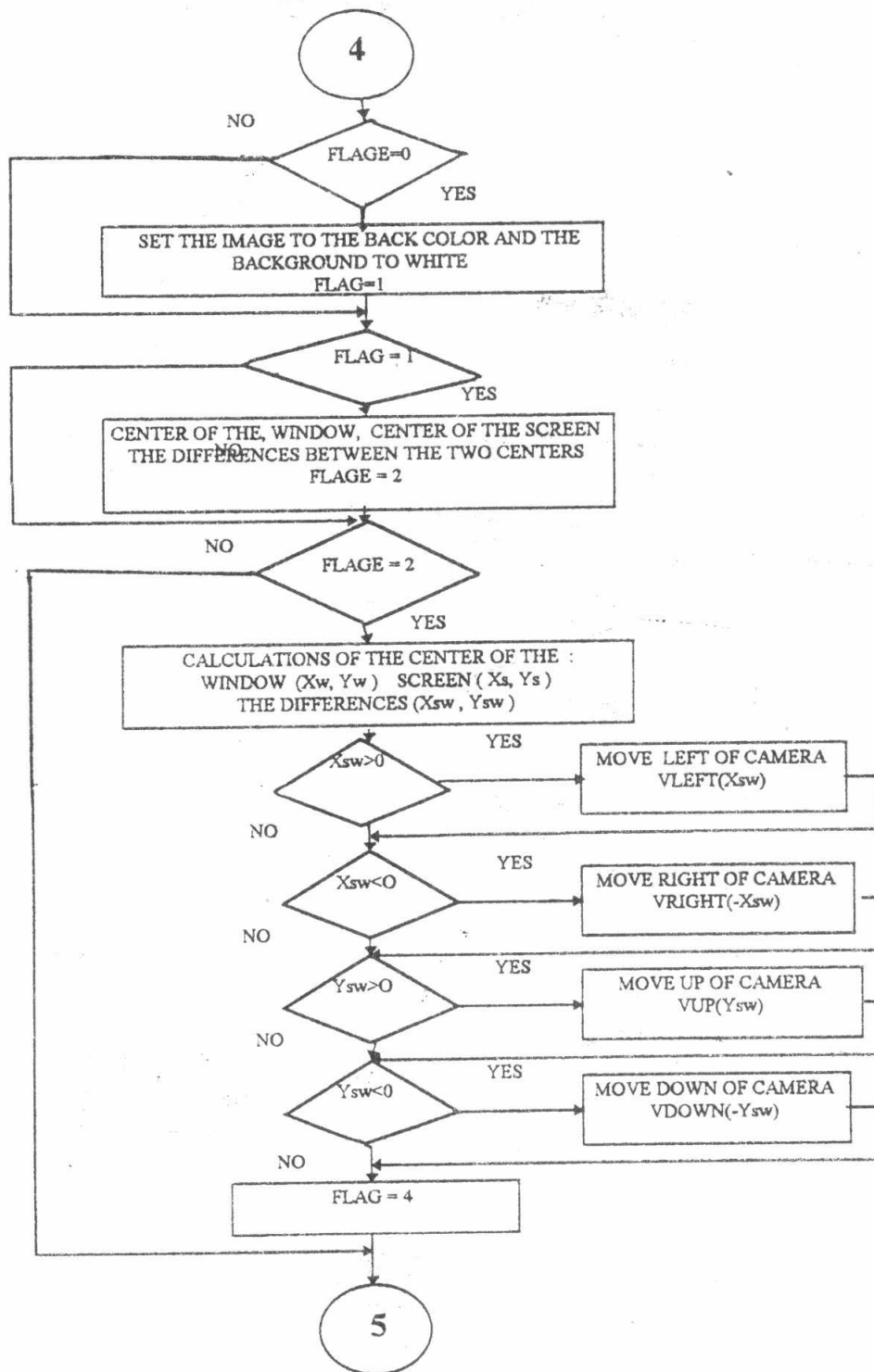
Fig .10(c) The flowchart of tracking image processing of the target

V . Real-Time target tracking implementation:

Using a source file called K7_1.C file, the camera capture two image files through Video Blaster and calculate the frame difference between them, generate scanning window to locate position of the intended target, calculate the coardinates of field of view and the scanning window, calculate the difference between these centers and the error signal used to align the camera. Fig.11 shows the flowchart for real-time target tracking process.
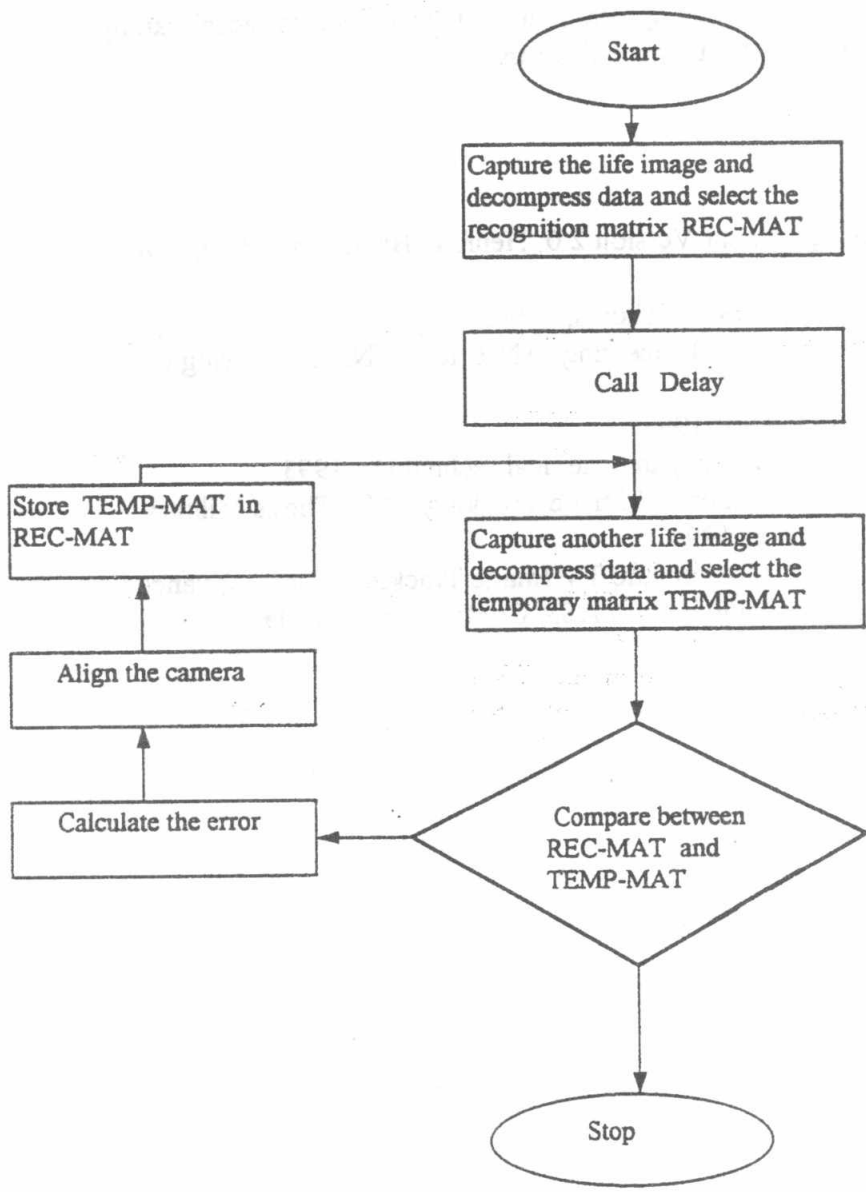
```
                        ┌──────────┐
                        │   Start  │
                        └────┬─────┘
                             │
              ┌──────────────▼──────────────┐
              │ Capture the life image and  │
              │ decompress data and select  │
              │ the recognition matrix      │
              │ REC-MAT                     │
              └──────────────┬──────────────┘
                             │
              ┌──────────────▼──────────────┐
              │        Call   Delay         │
              └──────────────┬──────────────┘
                             │
  ┌───────────────┐          │
  │ Store TEMP-MAT │◄─────┐   │
  │ in REC-MAT     │      │   ▼
  └───────┬───────┘    ┌──────────────────────┐
          ▲            │ Capture another life │
  ┌───────┴───────┐    │ image and decompress │
  │ Align the     │    │ data and select the  │
  │ camera        │    │ temporary matrix     │
  └───────┬───────┘    │ TEMP-MAT             │
          ▲            └──────────┬───────────┘
  ┌───────┴───────┐               │
  │ Calculate the │◄──────◇ Compare between
  │ error         │         REC-MAT and
  └───────────────┘         TEMP-MAT
                               │
                        ┌──────▼─────┐
                        │   Stop     │
                        └────────────┘
```

Fig .11   The flowchart for real time target tracking process

Conclusions and future work :

The conventional TV tracking system uses a statistical clustering algorithm for target /background separation. This technique requires initial manual location, and size adjustement of the window around the target.

The proposed system has the ability to identify and track close distint targets .The evaluation of the target center ,orientation angle, are introduced to improve the tracking process. The interface link between the TV camera driving system and PC computer, and the accompanying enable te motion control, either manually using the joystick or automatically through PC computer.

As a future work, the use of parallel processing techniques can be applied for accelerating the processing speed and getting nearer to real time applications.

References :

1-Ad Oculos, Digital Image Processing Student Version 2.0, Henning Bassmann, Philipp W. Besslich, 1995.

2- Applied Image Processing , G.W. Awcock& R. Thoms, 1996.

3- Advance in Mathematical Models For Image Processing, ANIL K. JAIN, Processing of IEEE, Vol.69, May 1981.

4- Video Blaster-SE User Guide Creative Labs 1993.

5- Modern Image Processing, Warping, Morphing and classical techniques, 1993.

6- Osama M.EL-Mowafy,"Image Processing For TV Image Tracking"M.Sc Thesis Military Technical College, Cairo,Egypt,December 1994.

7- S.F.Bahgat,O.EL.Mowafy, S.Ghoniemy "Automatic TV Image Tracking Using Advanced Positive Frame Difference Window Technique "sixth ASA Conference , 2-4 May 1995, Cairo, Egypt, p.p.245-258.