The Snap Camera Manual

Provisional Copy

Alex Van Someren

Micro-Robotics

Contents

Introduction

Setting up your camera

Supplied Software

-EV1 -MOVIE

- SECURE

-ARTY

-ANIMAL

Applications of Electronic Vision

Projects

Optics - Adding to your camera

Principles of Operation

Service calls

Appendix - Software listings

Introduction

The Snap camera allows your computer to capture and process pictures. By programming the computer it Is possible to make use of these pictures in any way you choose, for example to automatically recognise objects. The camera has the same effect and the same sort of controls as an ordinary film camera, but the computer can extend its usefulness a great deal because the picture is stored electronically. This means that you can examine and alter parts of the picture in ways which film cameras do not allow sophisticated software is supplied which allows you to make full use of the facilities that the camera offers. If you wish to perform more complex tasks, the best use is made of the camera by writing your own programs. The first chapter of this manual describes how to plug in your camera and start to use it. Since the camera is a sensitive piece of electronics it is important that you read this section carefully before you try anything else. Once you have successfully installed the camera you can then read the rest of the manual in any order you chose. The second chapter covers the software which is supplied, and examines each program In detail. Chapter Three examines some of the current uses of electronic vision in industry. Chapter Four discusses further projects for you to experiment with, whilst Chapter Five examines the various optical extras which you can attach to the camera to expand its capabilities.

In chapters six and seven you will find detailed technical Information on the operation of the camera. This is only essential reading if you are planning to write your own software for the camera, otherwise you can ignore it.

Whilst Snap is complex it is possible to achieve impressive results very quickly. The limit to what you can do is really set by the software which you use, since the computer is capable of doing a great deal of work with careful programming. The suggestions which are Included in Chapter Four are designed to give you an Idea of the sort of application which is feasible using a microcomputer. They should provide the basis for many other projects along similar lines, and the techniques which they use are directly useful in all manner of other ways.

Chapter One
Setting up the camera

When you receive your camera the package should contain the following parts. Ensure that each is present before you continue, and If any items are missing then get back in touch with the dealer or representative where you purchased it.

The package should contain:

1 Camera with lens attached
1 Copy of this manual
1 Disc or Cassette of software

Ensure that your computer is turned off and unplugged from the mains supply. Turn the computer upside down so that the set of connectors on the underside are facing towards you. Plug the camera lead into the socket marked USER PORT so that the lead sticks out away from the computer. Ensure that the plug is pushed fully home and the retaining tabs on either side are clicked home.

Once you have plugged the camera in, turn the computer over again. Do not turn it on yet. We shall first set the scene for testing it.

Place a piece of printed text (for example the front of the BBC Micro User Guide) about 18 inches In front of the camera. Adjust the focus control to 1.5 feet (0.5 metres) and set the camera on a flat surface or preferably a tripod. Position a fairly bright light (a 60 Watt bulb is best) so that it points at the manual from a couple of feet away.

Operation

Now that the camera is plugged in you are ready to use it. Turn the computer on at the mains and insert the disc or cassette of software into the disc drive or tape recorder. Type at the keyboard:

CHAIN"EV1"

and then press RETURN. The disc drive should start to whirr or the Cassette Motor light will come on. When the program is loaded press the STOP button if you are using a tape recorder. The program should start with a rough picture of the manual on the screen. The program has done its best to cope with the lighting available and has made a rough guess at the exposure time needed to produce a good picture. ft has not, however, found the best possible exposure; and you will need to adjust this by hand. Using the UP ARROW and DOWN ARROW keys on the keyboard you can change the exposure time until the picture looks sharp. If you do not seem to be able to get it perfect then try moving the lighting around: too much lighting can saturate the picture. Having done this then re-adjust the focus on the camera and try using the arrow keys once again. If you are still unable to get a clear picture then consult the Faults section below, otherwise go on to Trying out the camera.

Faults

If you some random combination of white and black on the screen then the camera is possibly not working correctly. Ensure that the camera is correctly plugged Into the computer, particularly checking to see that the plug is firmly pressed into the User Port socket and the correct way around. If you are still unable to make the camera operate correctly then contact your supplier for advice.

Trying out the camera

You are now ready to try using the camera, The EV1 software will display whatever the camera sees on the screen, and it will keep changing the picture as fast as it can A good thing to start with is your own face. Point the camera at yourself and adjust the lighting so that you are tit full-on from the front. If the lighting is even you should begin to some sort of image of your face on the screen. You will probably need to adjust the exposure (with UP and DOWN ARROW) and you may also need to change the focus to get a sharp picture.
Another good test is to draw some symbols or writing on a piece of white paper and place it in front of the camera. Note that a black pen is best and that because of the colour response of the camera, red pens will not give very good results. Most importantly though, try to discover the limitations of the camera and lens. You can try to get pictures of any convenient objects (computers, books, pens, other cameras!) and you should find that a little adjustment of exposure and focus will consistently give good results.

Hints and tips

Here are a few hints on how to get the most out of the camera:

- The camera does not respond to colours in the same way as the eye. Notice that reds do not show up nearly as well as other colours. This is because the camera is over-sensitive to the red end of the spectrum, making reds look as bright aswhite in some cases.

- The camera will not work well in direct sunlight. This is because sunlight is usually too bright for it (even in this country!). It is possible to attach filters to the lens to compensate for this: see the Optics section later in this manual for further details

- An easy way to focus the camera on a particular scene Is to hold something, such as the page of a book, just in front of the spot you want to photograph. Then adjust the focus on the camera so that it is sharp on the screen. The important thing is to ensure that sufficient detail Is visible in the camera's picture. (This may need two people).

- All black marker pens are not equal; that is, whilst they tend to look the same to the human eye, they present different degrees of darkness to the camera. The Pentel Marker Pen has proved to give very good results.

- It is usually better to have a well-lit scene and the exposure short than the other way around. This allows the camera to update the picture more quickly. Be careful not to get the lighting too bright, as this will saturate the scene and make photography impossible.

Thin line drawings will not show up well through the camera. Remember that when the camera is focussed the line must be at least one pixel wide on the screen.

The camera has a 'blind spot' in the centre of its area, causing a discontinuity in the picture. As this is a property of the chip there is no way of entirely avoiding it. However, we do not feel that it presents any serious difficulties but it should at all times be borne in mind.

The Applications Software

Five pieces of software are supplied with the Snap system to help you get started. Some users may find that one of these packages will provide the functions they want for a particular application; others will see the need to modify them or even to start from scratch Details of how the software works can be round in the Advanced Programmers Guide, for the present we shall just concentrate on how they are used.

Each program is described in three parts: its purpose, its controls and a description of how it is used.

EV1

Purpose

The EV1 program is the basic program for displaying what the camera sees. It repeatedly displays the scene that the camera is pointing at on the screen. A facility exists to capture a particular frame and keep it as a still, after which the frame can be saved onto tape or disc. It is also possible to retrieve a stored frame and display it on the screen.

Controls

UP-ARROW	-Increase exposure
DOWN-ARROW	-Decrease exposure
SPACE-BAR	-Switch between Still and Moving modes
COPY	-Screen dump

Description

The program is used by typing CHAIN"EV1" and then pressing RETURN. This will cause the program to be loaded, and it will immediately display some kind of picture on the screen.

The computer starts by automatically finding an approximate exposure setting for the scene which it can see through the camera. You will usually then have to set the exposure more accurately with the UP and DOWN ARROW keys, which increase and decrease the exposure time respectively. You should also adjust the focus control on the camera, The software will continuously display the scene, allowing you to adjust the focus and exposure to the desired levels. Pressing the SPACE-BAR will freeze the picture until you hit it again, allowing you to take a still photograph in much the same way as with an ordinary film camera.

Pressing the COPY key will cause the picture on the screen to be dumped to a graphics printer connected to the Parallel Printer Port. The screen dump is configured for an Epson MX-series printer and further details may be found in the chapter seven.

MOVIE

Purpose

The Movie software will allow you to record and replay a sequence of frames in the computer's memory. Thus the program can be used to create short films or animated sequences and display them on demand. It is used in much the same way as a video cassette recorder and camera.

Controls
SPACE-BAR -Record

Description

The program is started by typing CHAIN"MOVIE" at the keyboard and pressing return. Use the UP and DOWN ARROW keys to adjust the exposure to the desired level, and then press the SPACE-BAR. This will start the camera recording the scene it sees, and after a few seconds it will then stop recording and play it back.

When you want to record a new movie, press the space bar again.

GREY

Purpose

The grey-scale program displays a picture over the entire screen with eight distinct levels of brightness. This Is as distinct from the usual two-colour (black and white) pictures that the camera generates. GREY provides pictures with a good deal more realism than the normal technique, and is particularly suited to photographing people and other scenes where shading is Important.

Controls

SPACE BAR -Switch between Freeze and Motion modes

Description

The program 1s started by typing CHAIN"GREY" and pressing RETURN, at which point the computer will immediately start to analyse the scene which can be seen through the camera. The computer tries to find a good contrast setting given the differences in brightness across the picture. It then displays the scene in eight levels of grey on the screen. This process is known as Auto Contrast Setting, and we shall look at Its uses further on. Notice that this allows the camera to photograph both a pound note (low contrast) and a dark doorway (high contrast) without you altering any controls.
The program is run by typing CHAIN"GREY" at the keyboard and pressing RETURN. Once running, the program will take a few moments to set the exposure and then display a changing picture on the screen. The picture takes rather longer to refresh itself in this program than it does In EV1 because of the eight grey levels. Pressing the SPACE BAR will cause the picture to freeze until you press it again.

SECURE

Purpose

The Secure program detects changes between a stored picture and the scene which the camera sees. It is very effective as a kind of burglar alarm, and it can also form the basis for a variety or change-detection applications of a simpler nature. An alarm is sounded when the picture has changed by more than a preset amount, and the program displays a running graph of the number of alarms over a period of several hours.

Controls

UP-ARROW	-Increase exposure
DOWN-ARROW	-Decrease exposure
SPACE BAR	-Select Expose/Run mode

Description

The Secure program is run by typing CHAIN"SECURE" at the keyboard and pressing RETURN. It begins by presenting the camera's view on the screen. Set the focus on the camera and adjust the exposure with the UP and DOWN ARROW keys to your satisfaction; having done this, press the SPACE BAR. This begins the analysis of the picture by the computer to find which parts of the picture are already changing. This is useful to prevent the computer from noticing things that always move (cars, swaying trees etc.). If you want all changes to be detected then hit the SPACE bar again straight away. Otherwise give the computer long enough to see all the parts of the scene you want excluded from the test change, then hit the SPACE bar.

At this point the computer is ready to detect differences between the scene it can see now and the original picture it was shown. It displays two graphs in the lower part of the screen. At

the bottom is a graph showing activity in the picture over each five minute period, and this will cycle along repeatedly. When the activity exceeds a certain limit the computer will make a bleep.

The upper graph counts these bleeps and displays them as a histogram, allowing the level of activity over an eight hour period to be recorded.

If you wish to reset the compared picture to a new scene then press RETURN.

ARTY

Purpose

The Arty software allows you to compose pictures on the screen and adjust their colour, position and the ways in which they overlap. It requires a standard linear (ACORN-type) joystick to be used effectively, and this should be connected to the Analogue Port on the rear of the computer in the normal way

Controls

UNKNOWN AT THIS TIME

Description

The software is run by typing CHAIN"ARTY" at the keyboard and pressing RETURN, at which point the scene which the camera sees will be displayed on the screen. You can move it around the screen using the Joystick, and pressing the FIRE button on the Joystick will deposit a copy on the screen.

MORE WHEN INSTRUCTIONS AVAILABLE

ANIMAL

Purpose

This software provides a sophisticated method for recognising objects. It is presented in a rather simple way, a visual analogue or the animal guessing-game that 1s available for many computers, but it should be recognised that the facilities it provides are well-suited to more complex applications. In particular it provides a very good starting-point for implementing some of the industrial applications we discussed in the previous chapter.

Controls

UP-ARROW	-Increase exposure
DOWN-ARROW	-Decrease exposure
SPACE BAR	-Analyse screen
L	-Learn screen

Description

The software is started by typing CHAIN"ANIMAL" at the keyboard and pressing RETURN. It will show the scene that the camera sees at the bottom of the screen, but notice that only the outlines of the visible objects are shown. At this paint you should adjust the focus on the camera and exposure with the UP and DOWN ARROW keys so that all the Items which you wish to examine are in the picture and have clean outlines which do not overlap. Pressing the SPACE bar will cause the computer to attempt to analyse the visible objects, initially it will have no Idea what the objects are, and will label them all Unknown. You may now teach the computer about the objects by pressing V, whereupon it will ask you what each of them is in turn. Enter a name for each type of object, but use the same name for objects that you wish the software to recognise as the

same. Once you have given it this information you may then proceed to present it with a new set of objects to attempt to recognise. Press the SPACE BAR to start the recognition.

Notice that the software will recognise a particular shape irrespective of which way up it is. This means that both the letter "W" and the letter "M" will be recognised as the same. The program will also distinguish between objects which are inside one another, so donut shapes will be seen as two circles, one inside the other.

Where you are drawing objects, try to use a thick black marker pen of some kind, and avoid making the lines too thin. It is best to use a very large sheet of plain white paper as the background.

If you are using real objects then ensure that the lighting is even over them. Shadows will cause the software to see the objects as being larger than they really are, so it is important to check the screen image for this kind of problem before you start the analysis. The best possible results are achieved by using a light-box (the kind used for looking at transparencies) and placing objects on top with the camera looking down.

Chapter 3

Industrial applications of Electronic Vision

The eye and the brain are capable of feats of recognition which will continue to stump electronic vision system designers for many years to come. However, there are often problems associated with having real people doing complex work, particularly when the environment they nave to work in is hazardous, it is frequently very expensive to have manual workers, and as the cost of automatic systems decreases they are applied to more and more tasks. But automation is not without it's own problems.

It is easy to forget what complex work our bodies do to help us get on in the world, and It is important to understand this when looking at uses of electronic vision. To give a good example, consider a production line worker who takes metal parts from a bin and assembles them onto a car engine. To do tills requires that a part in a random position in the bin is removed, turned the correct way around and then fitted Into the engine. It is currently impossible for a computer to accomplish this feat, and yet even young children would have little trouble doing it after minimal training. This is an example of the difference between the way the Human brain works and the way a computer works. It is expected by experts in this field that It will be possible for a computer to do this by around 1988.

One of the reasons that we have such difficulty in telling a computer how to perform such a feat is that we simply don't understand how to do It ourselves. By saying that we mean that it is impossible to sit down and list every detail of what our brain and eyes go through to do it. Phrases like "See if the part is the right way around" are easy enough for a human to understand, but the process is actually fiendishly complex. A great deal of

research has been done, and many techniques have been developed in order to allow such tasks to be automated more efficiently.

Having said all that, you will probably be surprised at what can be achieved with the Snap system and a little programming effort. One of the appealing aspects of electronic vision and robot technology in general is that it is very easy to experiment with them. The family of products of which this camera is a part have been specifically designed to allow people to do just this kind of experimentation. In order to give you an idea of the kinds of things which are done at the moment by computer we will look at some of the most sophisticated industrial applications of electronic vision in the rest of this chapter.

Using robots for welding

A successful area of application for electronic vision is the automatic welding of metal parts, particularly for the automotive industry. To accomplish this a vision sensor of some kind and a robot welding torch are controlled by computer in order to weld a seam or joint The thing that makes this task difficult is that the pieces to be welded are not always the same way around (they are frequently on a conveyor belt) and the seam to be welded is often not straight This means that the robot really needs to look at the parts and find the seam before it can direct the torch. After finding the start of the seam, it is a case of following the seam by sight until the weld Is completed.

This is a classic example of what is known as a feedback loop. Here the picture that the camera sees controls the movement of the welding torch. In order to follow the seam to be welded, the computer must be able to see the section of seam which is about to come under the torch. It must then move the torch to follow the pattern of the seam. Having welded a section the computer must also ensure that the weld was successful, and

so it needs to be able to return to the seam if the parts did not weld correctly.

A technique which is commonly used for this process Is to shine a thin slit of light across the seam of the weld from above the torch and then view the resulting scene through the camera.

This method, known as Structured Lighting, allows the computer to pick out depth information about the parts being viewed. Further details can be found in books (1), (3) and (4).

Automatic assembly by robots

Building a complex device out of mechanical components is an example of the kind of assembly tasks which need electronic vision. For example a number of components (rotors, washers, magnets etc.) might need to be assembled into a casing. The components are usually fed from trays or racks, and the robot simply needs to pick them up in the correct order and Insert them.

In this kind of application the camera is important in two respects. Firstly it is used to Identify and orientate the parts before they are assembled. Secondly it is needed to ensure that the parts are fitted together correctly and to Identify Jamming of the components which then need to be repositioned. The computer assists in these operations by allowing the parts to be identified and manipulated. Using recognition techniques and software which can backtrack through a sequence of movements, damaged or misplaced components can be identified.

Painting by electronic vision

The painting of manufactured items is a messy and unpleasant job which has traditionally been done manually for a variety of practical reasons.

Method	For	Against
Manual	-Efficient use of point	-Unhealthy
Automatic	-High quality	-Wastes paint
Robot	-High quality	
	-Efficient use of paint	

Fig 1. Comparison of spraying methods

With robot spraying the parts are usually fed past a spray-gun and camera by means of an overhead conveyor. The computer examines the parts and directs the spray-gun accordingly. When trying to make an automatic system to accomplish this task, one group of designers discovered and interesting problem with the usual feedback method. As soon as the camera had started to direct the spraying arm over the object, a large cloud of paint obscured the camera's view. This meant Chat the computer went on spraying the cloud of paint. They got around this problem by moving the spray gun upstream on the conveyor so that it could spray the part that the camera had just seen, without interfering with the vision system.

Decorating chocolates by robot

A rather appealing application of electronic vision is the automatic decoration of chocolates. Chocolates are delicate objects and cannot under any circumstances be handled. In the past a large number of trained manual-workers have Individually Identified and iced the chocolates as they appear from the ovens. Clearly this is tedious and expensive. A successful automation of this task has been achieved by having a conveyor belt covered in chocolates of assorted shapes and sizes pass underneath a camera. The computer identifies the chocolates from the picture and squirts the appropriate pattern onto each as they pass below. Any reader who has tried this themselves will understand what a boon it is to have a machine

do it accurately and repetitively.

Conclusion

Even if you are not planning to set up your own automated production line, these examples do show what sort of results are possible using electronic vision systems. In the past these techniques have only been available to Industry because of the high cost of the equipment used. But It Is not difficult to draw a parallel between the squirting of icing on chocolates and the spraying of water on potted plants. This kind of application is well within the means of the owner of a Snap camera and some kind of robot arm. The next chapter discusses some of the applications that are possible using various extra pieces of hardware, but you will need to write the software yourself. The Advanced section of this manual contains information which you will probably find valuable in this task.

Chapter 4
Suggested projects

The following are some uses of the Snap camera which we have been unable to explore, but which we feel might be found useful. Do remember that the real value of robotic devices such as Snap is in their use as tools for experimenting. Much still needs to be discovered in the fields of Robotics and Artificial Intelligence and who knows, maybe you will discover something Important. Above all else you can bear in mind that almost nobody knows any more than you do!

A servo-controlled camera tripod

The Beasty controller and the supporting range of servo motors which it drives can be used in many ways to expand what the Snap camera can do. One example Is the construction of a motor-driven mounting which will allow software control of the direction In which the camera points. This can be achieved in a number of ways, for example by placing the camera at the end of a Beasty arm. Alternatively by mounting the camera on two servos, one for Pan and the other for Tilt, a simple co-ordinate positioning system can be devised. This done, the camera could be programmed to follow a moving object so as to keep it central in the field of view. One expansion you might like to consider is to have the focus control on the camera controlled by a separate servo. This could be used to provide control in an automatic focussing system where the image was examined and the focus changed until the scene was correctly focussed.

Mechanical colour separation

Since the Snap camera cannot in itself provide colour pictures, it would be useful to have a method for producing them. One possibility is to photograph a scene several times through a set of coloured filters and then combine the pictures In colour on the computer's screen under software control. This would be a rather tedious task but for the option of using a servo to select each of the colour filters in turn. This could be achieved by fitting the filters into a disc and turning the disc in front of the camera lens so that each of the filters can be used. Having recorded the image through each of the filters, it simply remains to generate a colour picture on the computer in a suitable graphics mode.

Robot Table-Tennis

For several years a competition for designing maze solving robots called Micromouse has been popular in the amateur robotics world. A great number of people have gained enjoyment and valuable experience from this competition, and more recently a further trial has been announced.

A robot table-tennis competition is to be established to allow robotics fans to try their skills. The idea is that a camera such as snap produces a picture from which the computer finds the ball and directs a bat to return it to the opposing robot. This is certainly a difficult task, but then people said that about the Micromouse contests too and they now have to have heats to filter out the hundreds of entrants each year.

Locating robot mobiles

One of the main problems that a mobile robot has is keeping track of exactly where it is. Many methods exist, such as sonar, radar, infra-red light sensors, collision switches etc., but they all have their problems. If a camera were used the mobile, or its host computer, might find this task rather simpler. For example by suspending a curved mirror on the ceiling above the mobile's terrain, the camera can provide a plan view of the area and object recognition software could then find the robot
Alternatively the robot could use its own camera to examine its surroundings and make an estimate of where it was from what it could see.

Chapter 5 Optics

The camera can use extra optics to see more in less detail or vice-versa. Various devices can be employed to accomplish this including using mirrors, lenses and filters. Here we shall discuss some of these additions and how they are useful:

Lenses

The lens mounting for the camera is a standard Pentax '110' size and a wide variety of lenses is available to fit it. The standard lens fitted to Snap is either a 24mm Standard lens or the 18mm Wide-angle lens. As their names imply the former is designed for general-purpose use whilst the latter has a wider field of view (with a consequent reduction in detail). A Zoom lens is available which provides a 20mm-40mm range and provides a variable field of view. The Close-up attachment for both 18mm and 24mm lenses allows the camera to focus on very close objects with 3 consequent Increase in detail.

Filters

A limited range of filters exist for the 24/18mm series.

-Neutral Density filters ND4 and ND8 cut down the amount of light entering the aperture and thus allow photography under very bright lighting, particularly outdoors.
They can be combined to provide a very large reduction in light levels.

-The Ultra-Violet filter has no effect on the Snap camera but It provides a very cheap method of protecting expensive lenses as it can be screwed onto the front of most of them. This prevents expensive damage from knocks, drops etc. and is HIGHLY RECOMMENDED.

-An Infra-Red filter can be obtained (with difficulty) through camera shops. This cuts out a very large proportion of the visible light and allows 'heat photographs' to be taken.

-Colour separation filters can be constructed from stage lighting gels (from theatrical suppliers) which may then be fitted over the front of the lens using home-made mounts. They will increase the foreground/background contrast of colour scenes. Though few filters exist to fit the l8/24mm lenses, the normal 49mm size of filter will work with the Zoom lens so this is the quickest solution.

Mirrors

Where the field of view needs to be dramatically increased mirrors are of significant value. Curved mirrors such as those found on the walls in large shops can be wall- or ceiling-mounted to extend camera coverage. They do tend to distort the scene rather heavily so be careful when using them in shape-analysis applications.

Chapter Six

Principles of Operation

The most important part of the Snap camera consists of a silicon chip known as the Image Sensor which is housed in an integrated circuit with a clear window on the top. The scene which the camera is pointing at is focused by the lens onto the surface of the chip. The computer is programmed to reconstruct the picture from the electronic signals and display it on the screen.

The image sensor consists of a large array (256 by 128 cells) of light-sensitive cells which can store an electrical charge. To take a picture, each of the cells is set to OK and the entire chip is exposed through the lens. As light from the scene hits the cells they slowly discharge. The more light that hits each cell, the faster it discharges. When a cell has discharged below a certain level it turns itself OFF, so that after the picture has been exposed we are left with an array of cells each of which is either ON or OFF. These states are interpreted as DARK and LIGHT respectively.

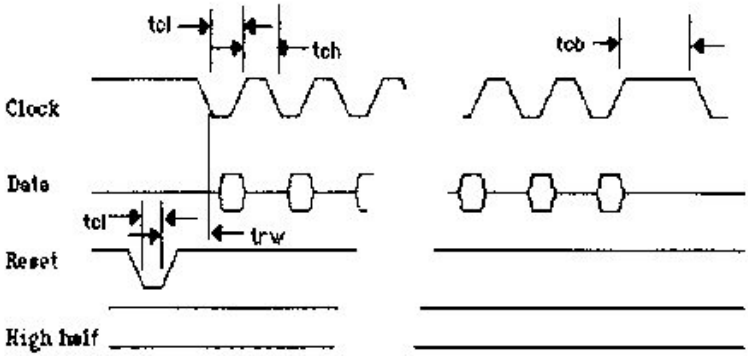
In order to control how sensitive the image sensor is to light (so as to cope with a range of lighting levels), the time for which it is exposed can be varied. This time is known as the Exposure Time, and the UP and DOWN ARROW keys are consistently used in the supplied software to allow you to set it.

Once the array has been exposed, its contents are turned into a stream of binary Information which is fed into the computer. The whole picture is transferred into the computer's RAM, and software then reconstructs it on the screen. It takes 17uSeconds to transfer each byte (of eight pixels) and thus the entire screen is copied in 0.064 seconds.

Interfacing

The camera has been designed to connect directly to the BBC Micro-computer but connection to any 6502/6522 based computer is relatively straightforward. Most other computers can be accommodated by means of a suitable Interface; see below.

The clock line must not be allowed to remain low for more than 10uSeconds, refer to any DRAM data sheet for further details. If the clock line is driven from software than hardware precautions should be taken.



Characteristic	Symbol	Min	Max	Units
Clock low	tcl	1	10	µsec
Clock high	tch	1	∞	µsec
Clock byte time	tcb	2		µsec
Reset low	trl	1		µsec
Reset settle time	trw	3		µsec

Inputting an image

To input an image

1) Pulse Reset low, and lower High half.
2) Clock the clock line low 16000 times reading one data bit on each rising edge.
3) Pulse the reset line
4) Raise High Half
5) Clock the clock line a further 16000 times reading one bit on each rising edge.

The sensor has now been reset. Software should now delay by the exposure time, which can be from nothing to a number of seconds, and repeat the above operation storing the data on the screen.

For best possible performance it is suggested that either a 6522 shift register or a serial-to-parallel circuit is used to decrease the burden on the software.

It is not necessary for more than one column of cells to be read. This means that software can be written which only interrogates part of the array, hence increasing the frame rate.

Alternate blocks of 64 cells in the array have their black and white states inverted bitwise, this can be adjusted by performing NewData = OldData EOR &FF for the relevant blocks. We suggest you examine the listings at the end of the book for more information.

Further Reading

As with most computing disciplines, Artificial Intelligence and Robotics are supported by a very large number of books. These vary in quality and complexity, as is to be expected, and so we include below references to some texts which we have found useful.

(1) Robot Vision
Alan Pugh
Springer Verlag - ISBN 0-903608-32-4

(2) Digital Image Processing
Raphael Gonzalez
AddIson-Wesley - ISBN 0-201-02597-3

(3) Artificial vision for robots
Prof. I. Alexander
Kogan Page - ISBN 0-85038-757-4

(4) Brains, Behaviour & Robotics
J.S. Albus
BYTE/McGraw-Hill - ISBN 0-07-000975-9

Chapter Seven

Service routines

This chapter is devoted to explanations of the machine-code routines which are components of the software supplied with Snap. Each is detailed in terms of its Name, Purpose, Memory Usage and Component routines. Full listings of all of the software discussed below are provided in the Appendix to this manual, and they should be consulted where the information below is insufficient.

Wide

Purpose

Wide contains the machine code drivers for MOVIE and EV1. Routines are available to initialise the camera, put an image on a Mode 4 screen, dump to a printer and to convert image between compressed and screen forms.

Memory usage

First address used given in JMPTAB!22
Start of code space given in JMPTAB!26
Start of jump table &57E0
Zero-page memory used &80 to &89

Black and white imaging routine

Entry point JMPTAB*0 &52C8
Purpose Takes a reading from the camera and places it on a mode 4 screen

Clear camera routine

Entry Point JMPTAB+6 &553C
Purpose Clear the camera clearing all the exposed cells it also returns the number of cells set in JMPTAB!18

Comments This routine is used to clear the camera before soaking it, and also to do automatic exposure.

setting using the cell count.

Load image routine

Entry point	JMPTAB+9 &535C
Purpose	Decodes a compressed image stored at the address given in &80,&81 and places the result on the screen.
On entry	Starting address of compressed given in &80,&81.
On exit	First location after compressed image given in &80, &61

Store image routine

Entry point	JMPTAB+12 &53E5
Purpose	Compress the current image
On entry	Return address for the compressed Image given in &80,&81.
On exit	First location after the compressed image given

in &80,&81

Comments

An uncompressed picture takes up 4K of screen space, and a totally random image could take up 4.1K in compressed form, in practice compressed images take up from 0.7K to 1.4K The method used is run-length encoding, where a string of the same values from the camera are converted into a count and value, hence any solid foreground or background will take up virtually no space.

Printer Dump routine

Entry point	JMPTAB+15	&5214
Purpose	dumps the current image onto a printer connected to the parallel port.	

Due to the large number of different printers with Incompatible specifications we have not attempted to support them all. The jump table entry can however be changed to jump to your own routine. To Install your own printer routine add the machine code below the driver and update 'First location used', 'Start of code space' and the jump table entry.

GREYO

Purpose A set of machine code drivers used by the application GREY

Memory usage

First address used given in JMPTAB!22
Start of code space given in JMPTAB!26
JMPTAB address &2FE0
Zero-page memory used 5.80 to 8.87

Grey scale routine

Entry Point JMPTAB+0

Purpose Displays the image exposed by half of the camera on a MODE 0 screen. The screen is divided up into 160 by 128 cells each containing 8 pixels, for each call of the routines only one of these pixels in each cell is changed. 8 calls of this routine are required to generate a complete image.

On entry Y contains the bit number of the pixel to be changed. (in the range 0 - 7).

On exit JMPTAB!18 contains the number of pixels set

Clear camera routine

Entry Point JMPTAB+6

Purpose Clear the camera and count the number of cells that were set.

On exit Number of pixels set given in JMPTAB!18

CHANGE

Purpose

Used to analyse changes in an image, this is the routine at the heart of the SECURE application.

Description

Change contains the four routines which compose the Secure software. Together they allow a scene to be captured, a mask generated for it and a continuous comparison of the current view performed on it. It is suggested that you consult the listing of this program in the Appendix for further information.

Memory usage

First address used given in	JMPTAB!22
Start of code space given In	JMPTAB!26
Start of jump table	&57E0
Zero-page memory used	&80 to &89

Learning routine

Entry point	JMPTAB+0
Purpose	Updates the screen to the current view. Sets bits in the mask where the current view differs from the template view.

Initialise routine

Entry point	JMPTAB+3
Purpose	Updates the screen and template with the current view. Clears the Image mask.

Clear routine

Entry point	JMPTAB*6
Purpose	Clears the camera ready to take an exposure.
	Running Routine Entry point JMPTAB+9
Purpose	Counts the number of sensitised cells that are different from the template and are not masked.
On exit	The number of differences are returned in JMPTAB+18

OUTL

Purpose

OUTL is a service useful when solving object recognition programs.

Description

OUTL contains service routines which are called by the Basic procedures PROCIMAGE, PROCOUTLINE and PROCCONTAINS. These routines perform low-level data manipulation which would be too slow from Basic. Basic is responsible for maintaining a variety of linked lists which these routines assume to be correct, thus it is recommended that these routines should only be called by the provided procedures. See the section 'OUTL basic support routines' for details of these procedures.

Memory usage

First address used given In	JMPTAB!22
Start of code space given in	JMPTAB!26
Start of jump table	&57E0
Zero-page memory used	&80 to &8F

Viewing routine

Entry point	JMPTAB*0	&4E31
Purpose	Place an outline of the current scene on the screen	

Object removal Initialisation routine

Entry point JMPTAB*6 &5521
Purpose It removes the outline of the first object on the screen each time it Is called.

Subsequent object removal routine

Entry point JMPTAB-9 &5527
Purpose Removes the next object on the screen. Returns a zero length object description if no object present.

Enclosure test routine

Entry point JMPTAB*12 &5694 Purpose Used to establish which objects are enclosed by other ones.

OUTL basic support routines

Purpose

These routines do all the memory management (maintenance of linked lists etc.) and tedious arithmetic (normalisation of area calculations etc.) required by the outl routines. If the outl routines are called without support from these routines, the user program will have to provide this support in order for the results to be meaningful and consistent. It is recommended that the Basic driver be used until the user has a thorough understanding of the implications of this. These routines are in 43 lines of Basic, and may be studied by the user.

PROCIMAGE

Purpose To get an outline on the screen.
Parameters None
Returns Image on the screen
Current value of exposure time (set by the user) LEARN: flag for learn mode of Animal program KEY: key press which terminated Imaging

PROCOUTLINE

Purpose To analyse outlines of all objects on screen.
Parameters START: start address of data storage area.
Returns Complete data block consisting of a set of linked object descriptions. See format of object description block for details of object descriptions.

PROCCONTAINS

Purpose To analyse relationships between objects.
Parameters START: start address of data storage area data supplied by PROCOUTLINE in this area.
Returns Data storage area updated to reflect the relationships between items in the scene.

Format of object description block.

Byte	Name	Description
0	TYPE	4 Bytes reserved for use by Basic
4	NEXT	pointer to next object at this level
8	CONTENTS	pointer to list of contained objects
12	AREA	area of thls object (in pixels)
16	PERIMETER	perimeter of this object (pixels on border)
20	XCEN	x-axis offset of centroid (from XLOC}
24	YCEN	y-axls offset of centrold (from YLOO
28	XLOC	x-axis location of object on screen
29	yloc	y-axls location of object on screen
30	XMAX	maximum x coordinate of object
31	XMIN	minimum x coordinate
32	YMAX	maximum y coordinate
33	YMIN	minimum y coordinate
34	definition	list of vectors in the object

Explanation of items In the data structure

TYPE is a four byte Integer which is reserved for the use of Basic applications, it is not used by the Animals application, out an example or its use would be to distinguish between 'O' and 'x' in a noughts and crosses playing program which used the camera for input.

NEXT is a four byte pointer to the next object at this level of the data structure. Note that after a call to PROCOUTLINE, all objects are returned at the same level, but proccqntains re-organlses the data so that all objects linked by their NEXT fields are contained by the same (larger) object {counting the screen as a single large object}. The last object in the list has a NEXT field of 0.

CONTENTS Is a four byte pointer to the list of objects contained by this object. (See discussion above)

AREA is the four byte area of the object in pixels.

PERIMITER is the four byte count of the number of pixels on the boundary of the object.

XCEN.YCEN contain the signed offset of the centre of mass of the object from the 'location' of the object (see next entry). All distances are given in screen units; to convert these to PLOT units (compatible with calls to the VDU from Basic) you should multiply these numbers by 4.

XLOC,YLOC contain the x and y co-ordinates of the first point that was found in the object. Since the driver routine searches the screen for objects using a raster scan which goes from top left to bottom right (in exactly the same way as the spot on a television set or monitor) this point will be the left-most point of the top line of the object. A more useful idea of the position of the object can be had by adding the offsets of the centre of gravity of the object to these co-ordinates,

XMAX,XMIN,YMAX,YMIN contain the maximum and minimum x and y co-ordinates of the object. These correspond with the greatest extent of the object (the object is contained within and touches the edges of a square defined by these values). These values are also in screen co-ordinates and need multiplying by 4 If they are to be used in plotting.

DEFINITION Is a representation of the outline of the object, coding each line element (which points In one of eight directions referred to by points of the compass — N Is straight up the screen) into a number, such that 1-SE, 2-S, 3-SW, 4-W, 5-NW, 6-N, 7-NE, 8-E. These numbers are coded as ASCII characters,

terminated by an ASCII NULL (character code 0). These are the codes that are used by the routines internally, but they may be printed out by the normal Basic PRINT function (as long as the list is not longer than Basics maximum string length of 255 characters). To do this you would type something like:

```
PRINT $START*36
```

Appendix 1:

Outl Basic support routine listings.

```
PROCIMAGE
1000 DEF PROCIMAGE
1010 CALL DRIVER
1020 REPEAT
1030 FOR I=0 TO expo:NEXT
1040 IF INKEY(-58) <> 0 expo=expo*1.1
1050 IF INKEY(-42) <> 0 expo=expo*0.9
1060 CALL DRIVER
1070 REPEAT: KEY=INKEY(0): UNTIL KEY<880
1080 UNTIL KEY <> -1
1090 LEARN = (KEY=ASC"L" OR KEY=ASC"I")
1100 ENDPROC

PROCOUTLINE
2000 DEF PROCOUTLINE
2010 I%=START: K$=0: CNT=0
2020 CALL DRIUER+6
2030 REPEAT
2040 IF L%=0 THEN UNTIL TRUE: ENDPROC
2050 IF L%<10 THEN 2130
2060  IF C%=0 THEN 2130  :REM Surface area  > 0?
2070 CNT=CNT+1: !I%=CNT: I%14=0: I%18=0
2080 I%!16=A%*1.14+B%: REM Perimeter
2090 I%!20=X%/C%: REM X centrold normalisation
2100 I%!24=Y%/C%
2110 IF K%<>0 THEN K%!4=I%; REM Same level link.
2120 K%=I%: I%=I%+L%+35
2130 CALL DRIVER+9
2140 UNTIL L%=0
2150 ENDPROC
```