University of West Bohemia

Faculty of Electrical Engineering

Department of Applied Electronics

Position, size and rotation normalisation

by

Radek Holota

Supervisor: B A Wilkie

May 2000

Acknowledgements

I would like to express my thanks to my supervisor:

Doc. Eur.Ing. B A Wilkie, Ph.D., C.Eng., MIEE

for his helpful guidance and advice throughout this project.

The special thanks go to:

Mr N Fudge

Mr S Gardner

Mr J Newton

for their technical and material support.

Finally, I would like to thank to the ***ERASMUS*** project for providing financial support during this course.

Abstract

This diploma project undertakes image pre-processing. It outlines several techniques of position, rotation and size normalisation. The principles of these techniques are explained and some problems with their use are pointed out. Some basic methods of normalisation systems are introduced here and their advantages and disadvantages are discussed. The main task of this project was the software design of a normalisation system, which can be used as an image pre-processing stage for an image recognition system. The normalisation system is incorporated within complex system, which also utilises an image recognition system [8]. This system uses N-tuple and Min/Max node methods for image recognition. The complex system can execute both parts separately or together.

Contents

List of Figures

List of Tables

Summary of Essential Abbreviations

DAC	Digital to Analogue Converter
DSP	Digital Signal Processor
GS	Greyscale
ITU	International Telecommunication Union
LUT	Look-Up Table
MRV	Maximum Radius Vector
PLA	Programmable Logic Array
PNM	Position Normalisation Module
P&SNM	Position & Size Normalisation Module
RCS	Radial Circular Scanning
RS	Running Sum
R&SNM	Rotation & Size Normalisation Module
VCO	Voltage Controlled Oscillator
WRS	Weighted Running Sum

1 Introduction

1.1 Image processing

Image processing is concerned with moving and composing images in order to improve the performance of image recognition systems or to help humans to look at them. It often is a form of pre-processing coming before an image recognition system. Image processing can cover many various functions (e.g. position, rotation and size normalisation; image enhancement and noise reduction).

1.1.1 Position normalisation

Position normalisation ensures that the position of the object of interest within the image will be normalised. It locates a centre of the object, which is important for rotation normalisation.

1.1.2 Rotation normalisation

The purpose of this form of normalisation is a rotation of the object to a predefined orientation. It uses the centre of the object, which is known from position normalisation, for rectangular-to-polar transformation. It finds a Maximum Radius Vector (MRV) and rotates to a normalised orientation. It often uses a retino-cortical transformation.

1.1.3 Size normalisation

The aim of size normalisation is to transform the dimension of the object within the image to a predetermined magnitude, without loss of proportionality or shape. The use of polar co-ordinates to map the dimensions of the object and so normalise its size about the centre of the image is not complex.

1.2 Image Recognition

Many of normalisation techniques and other enhancement help to improve the performance of an image recognition device. Image recognition is a very important scientific problem, which is solved in a lot of industrial branches. It can be used e.g. in industry for robot vision, in Astronomy to explore space or in security systems.

2 Theory of normalisation

2.1 Position normalisation

2.1.1 Methods of position normalisation

The aim of position normalisation is to enable the output of an image with the object of that image at the centre or other pre-defined position. There are several methods of implementing this function:

I. 'Real –Time Position Normalisation'

This method is described in [5] and locates the x- and y- centroid co-ordinates of a binary image.

The y-centroid is found during the input field, but the x-centroid is found in the second field by re-sampling the input image. In odd fields, pixels are read into a frame store. The cumulative sum of black pixels is stored at the end of each row in a scratch-pad memory. At the end of the field, the running sum is divided by 2 to obtain the 'half area'. The y-centroid is the number of the row in which the running sum first exceeds the half area. In even fields, the image is re-sampled in the input frame store and the same algorithm, for finding the y-centroid, is applied to find the x-centroid. The half area from the previous odd field is used.

These centroid values are used to calculate an address offset to enable the centred image to be output.

II. 'Character Size and Position Normalisation'

This algorithm is described in [7] and combines size and position normalisation in a single architecture. Another name for this method is the Minimum Enclosed Rectangle Method.

The object is not centred but shifted to the top-left-hand corner of the image. Then the size normalisation is applied. Expansion ratios are used for the calculation of a frequency at which the stored image can be sampled.

$$\text{Vertical expansion ratio} = \frac{\text{height of window}}{\text{height of object}}$$

$$\text{Horizontal expansion ratio} = \frac{\text{width of window}}{\text{width of object}}$$

III. 'Implementation of a Position –Normalisation Module'

This solution is described in [3] and involves the use of the 'half-area' as in (I) above.

In the next part, this method and improvements are discussed.

2.1.2 Calculation of x- and y-centroid

In principle, the Position Normalisation Module utilises 'running sum' and 'weighted running sum' operations. It needs the total black pixel sum. This problem is solved by assuming similarity in area of the current field and the previous field.

- X–centroids

  When x–centroid is calculated, it uses 'weighted running sum' along each TV line. These sums are linearly summed and result is then divided by the total black pixel count. The value of the x–centroid is the integer of the yielded quotient. This is shown in equation (2.1).

$$x = INT \frac{\sum_{r=1}^{R}\sum_{c=1}^{C} c[p(c,r)]}{\sum_{r=1}^{R}\sum_{c=1}^{C} p(c,r)} \qquad (2.1)$$

  where x = x-centroid

  C number of columns

  R number of rows

  p(c,r) the value of the pixel (1 for black, 0 for white) at rectangular co-ordinates c,r.

- X–centroids (different arrangements)

  This arrangement is used in practice in order to reduce propagation delays and the number of components used. It uses a 'weighted running sum'

(WRS) along each TV line as in the previous method, but the result of each WRS is divided by the total black pixel count. A remainder is carried forward to the next line. In the end, partial quotients are summed and the result is the value of the x-centroid. This is shown in equation (2.2).

$$x = \sum_{r=1}^{R} \left\{ \frac{R(r-1) + \sum_{c=1}^{C} c[p(c,r)]}{\sum_{r=1}^{R} \sum_{c=1}^{C} c[p(c,r)]} \right\} \qquad (2.2)$$

where x = x-centroid

C = number of columns

R = number of rows

p(c,r) the value of the pixel (1 for black, 0 for white) at rectangular co-ordinates c,r.

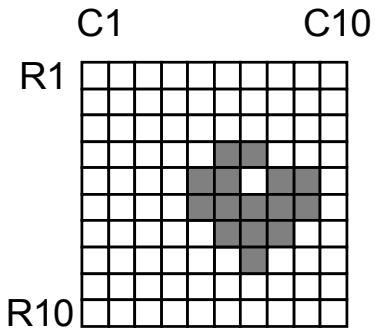
R(r-1) remainder left from previous line

- Y–centroids

  Calculation of the y-centroid uses only the 'running sum' (RS). A value of the y-centroid is a row's number where the RS is equal to, or greater than, half of the total black pixel count.

2.1.3 Finding the centroids of an image (example)

Input image:



Total black pixel count = 15

(It is obtained from the previous field)

X-centroid:

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Σ
R1	0	0	0	0	0	0	0	0	0	0	0
R2	0	0	0	0	0	0	0	0	0	0	0
R3	0	0	0	0	0	0	0	0	0	0	0
R4	0	0	0	0	0	6	13	13	13	13	13
R5	0	0	0	0	5	11	11	19	28	28	28
R6	0	0	0	0	5	11	18	26	35	35	35
R7	0	0	0	0	0	6	13	21	21	21	21
R8	0	0	0	0	0	0	7	7	7	7	7
R9	0	0	0	0	0	0	0	0	0	0	0
R10	0	0	0	0	0	0	0	0	0	0	0
											104

$$x = INT\left[\frac{104}{15}\right] = 7$$

X-centroid (different arrangement):

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Σ	Quot.	Rem.
R1	0	0	0	0	0	0	0	0	0	0	0	0	0
R2	0	0	0	0	0	0	0	0	0	0	0	0	0
R3	0	0	0	0	0	0	0	0	0	0	0	0	0
R4	0	0	0	0	0	6	13	13	13	13	13	0	13
R5	13	13	13	13	18	24	24	32	41	41	41	2	11
R6	11	11	11	11	16	22	29	37	46	46	46	3	1
R7	1	1	1	1	1	7	14	22	22	22	22	1	7
R8	7	7	7	7	7	7	14	22	22	22	22	1	7
R9	7	7	7	7	7	7	7	7	7	7	7	0	7
R10	7	7	7	7	7	7	7	7	7	7	7	0	7
												x=7	

Y-centroid:

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
RS	0	0	0	2	6	11	14	15	15	15

RS(R6)>(15/2)

i.e. y=6

2.2 Size normalisation

There are several techniques for size normalisation. Some of these techniques use the manipulation of rectangular arrays. These techniques have one disadvantage. They can be asymmetrical. But if we use radial scanning then the size normalisation will be symmetrical.

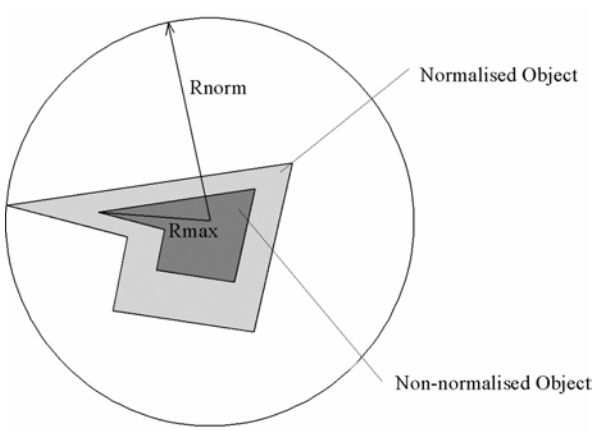


Figure 2.2.1: Size normalisation

There is a normalising radius R_{norm}, which indicates radial distance from the most extreme point of the normalised object to the centre point. The maximum radius vector (MRV) determines the size of the object under inspection. Using equation (2.3) the object is transformed radially, from the image centre, to achieve the required dimensions.

$$R_{new} = R \cdot \left[1 + \frac{\left(R_{norm} - R_{\max} \right)}{R_{\max}} \right] \tag{2.3}$$

In hardware, the MRV is applied to a Digital to Analogue Converter (DAC). A Voltage Controlled Oscillator (VCO) is controlled by the resulting analogue signal. The VCO determines the transferred data rate. The data are transferred to a video output store at normal sampling frequency, if $R_{max}=R_{norm}$. If the size of object is half, $R_{max}=R_{norm}/2$, then the transferred data rate is half. Some normalisation systems provide a possibility to pre-set the centre sampling frequency of the VCO. The pre-set is referred to some reference sizes.

2.3 Rotation normalisation

When the image of an object is presented to an image recognition device, it could be at any angle of rotation relative to the horizontal. Therefore, the system must be trained to recognise that object at any angle for successful recognition. This requires higher performance of a system. If a rotationally invariant image is presented instead, then the system will only require training

on images from a very reduced range of angles of rotation. Hence, rotation normalisation can greatly increase a performance of image recognition systems.

2.3.1 Retino-Cortical Transformation

This method changes size and angular rotation of the retinal image into orthogonal shifts of the cortical image which simplifies subsequent processing. Retino-cortical transformation is shown in Fig. (2.3.1). The scan-angle and pixel position within that scan form a composite address for the cortical image. Example is in Appendix E.

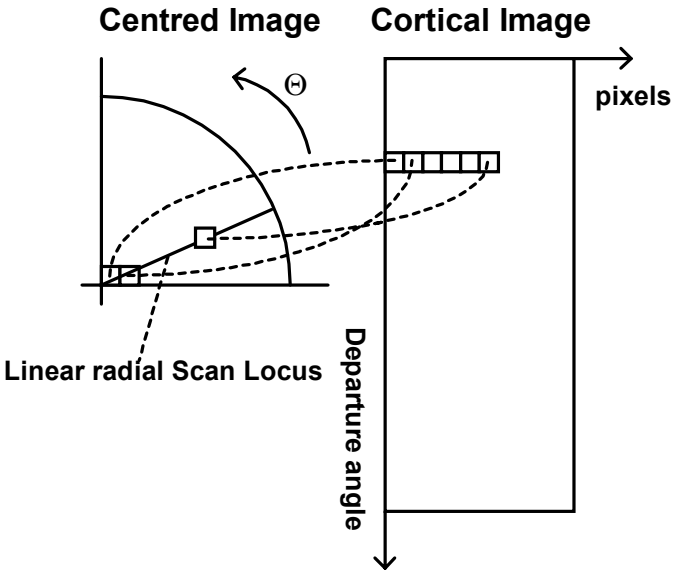


Figure 2.3.1: Retino-cortical transformation

2.3.1.1 The Normalised Cortical Image

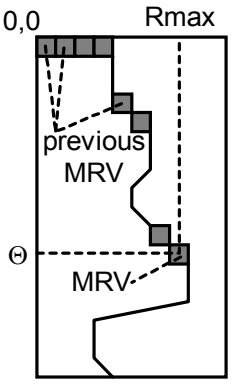


Figure 2.3.2: Detection of Maximum Radius Vector

The rotation normalisation process involves detection of the Maximum Radius Vector (MRV) of the centred image. This value is used as a reference for orthogonal shifts of the cortical image. First of all, the MRV is set to zero. Then the distance of each object-pixel is compared to the current MRV. The MRV is updated when the distance is greater. An example is shown in Fig. (2.3.2).

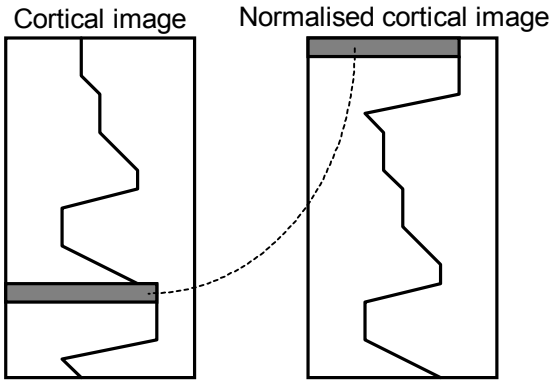


Figure 2.3.3: Shift of Maximum Radius Vector

The principle of the normalisation is very simple. The MRV is shifted to angle θ=0. This shift needn't take place but need only offset the linear address by θ. This is shown in Fig. (2.3.3). Example is in Appendix E.

2.3.1.2 The Square Tessellation Problem

The main problem of cortical transformation is the error, which is caused by the square tessellation of the pixels in an image. The causes and effects of this quantisation error are shown in Fig. (2.3.4).

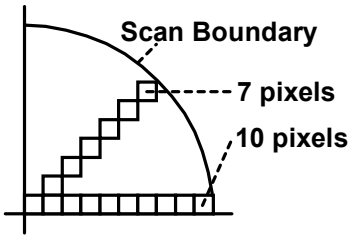


Figure 2.3.4: The square tessellation problem

2.3.2 Radial Circular Scanning

Instead of taking linear radial scans of the centred image, which erroneously hold different numbers of pixels, the Radial Circular Scanning (RCS) method uses a scan-locus which contains the same number of pixels in each scan, regardless of the scan-departure-angle. These are main idea and advantage of the RCS method. The use of this scanning makes better utilisation of the input image store. The resolution of the radial scan isn't reduced when the scan is rotated throughout the four quadrants.

2.3.2.1 Principles of radial circular scanning

The basis of RCS is shown in Fig. (2.3.5a) and (2.3.5b). The scan-locus is a quarter of a circle of radius $R_{max}/2$, where R_{max} is the radius of the scan boundary. The centre of scan-locus for a departure angle θ is positioned

orthogonally i.e. θ+90°. Its radius is $R_{\max}/\sqrt{2}$. Then the number of samples is equal to R_{max}. This is derived in the paper [11].

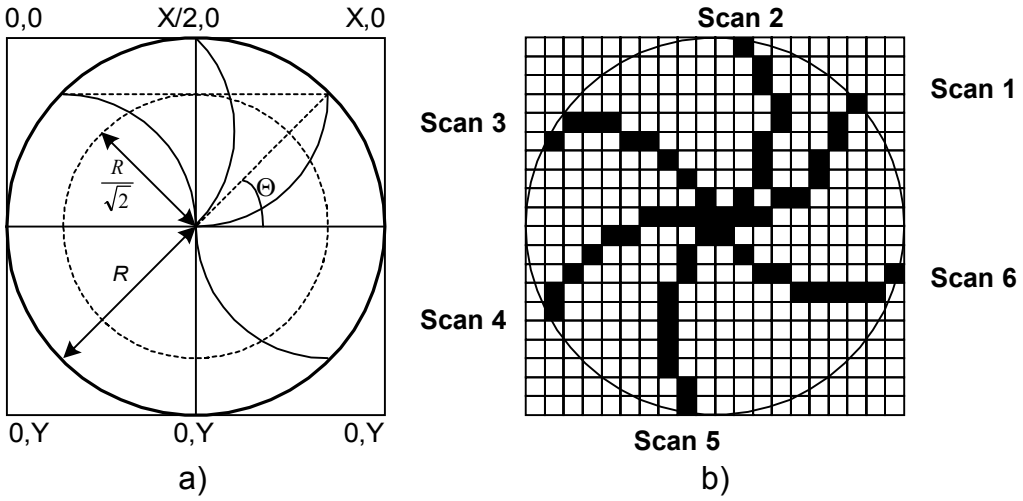


Figure 2.3.5: Principle of radial circular scanning

2.3.2.2 Radial-to-Raster Transform

The x and y co-ordinates of any radial scan are calculated with using equations (2.4) and (2.5). The equations are made up of three parts. The first part is an offset for the centre of the radial scan from the 0,0 position of a raster scan. We suppose that the 0,0 reference co-ordinates are at the top left-hand side of the image Fig. (2.3.5a). The second part is an offset for the centre of radius of the radial circular scan and the third one is a pixel position in the radial circular scan.

$$x = INT\left[\frac{X}{2} + \frac{X}{2\sqrt{2}}\left\{\cos\left(\frac{\pi}{2} + \frac{2\pi\theta}{Q}\right)\right\} + \frac{X}{2\sqrt{2}}\left\{\cos\left(\frac{3\pi}{2} + \frac{2\pi\theta}{Q} + \frac{\pi}{2}\frac{R}{P}\right)\right\}\right] \qquad (2.4)$$

and

$$y = INT\left[\frac{Y}{2} - \frac{Y}{2\sqrt{2}}\left\{\cos\left(\frac{\pi}{2} + \frac{2\pi\theta}{Q}\right)\right\} - \frac{Y}{2\sqrt{2}}\left\{\cos\left(\frac{3\pi}{2} + \frac{2\pi\theta}{Q} + \frac{\pi}{2}\frac{R}{P}\right)\right\}\right] \qquad (2.5)$$

where

X=Y Width and height of area

θ = departure angle

Q = number of radial circular scans mapped from the image

R Arc length of the radial circular scan

P Number of pixels in one radial circular scan

2.3.2.3 Over-sampling and under-sampling

There is a problem in the radial scanning technique. The data are over-sampled in the foveal region and under-sampled in the peripheral region. The equation (2.6) implies that a boundary between over-sampling and under-sampling area is at $R_{\max}/\sqrt{2}$. This circumstance evokes only the 0.707 reduction in peripheral resolution. It is unimportant in practice.

$$\text{Sampling resolution} = \frac{\mathrm{R}_{\max}}{\mathrm{INT}\left[\mathrm{R}\sqrt{2}\right]} \tag{2.6}$$

where

R_{max} Maximum Square Element Radius

R Square Element Radius

3 Methods of realisation

This chapter introduces basic methods for the realisation of position, rotation and size normalisation systems.

Basic types of realisation are:

- ***Software realisation***

This method is very good for testing and developing new normalisation algorithms, educational purposes and the presentation of current algorithms.

- ***Hardware realisation***

The advantage of this method is its high speed. The real-time normalisation at TV frame rates can be achieved. These systems are usually developed for real applications and a change of algorithm is very difficult or impossible.

3.1 Software realisation

This type of realisation is very important for its flexibility. Different parts of the program, the algorithms and the methods of calculation can be simply changed. This main advantage is used in testing and developing processes. Furthermore, it is very practical for the presentation of algorithms and educational purposes because the requirements for special hardware are not high. So this method is cheaper than a full hardware solution.

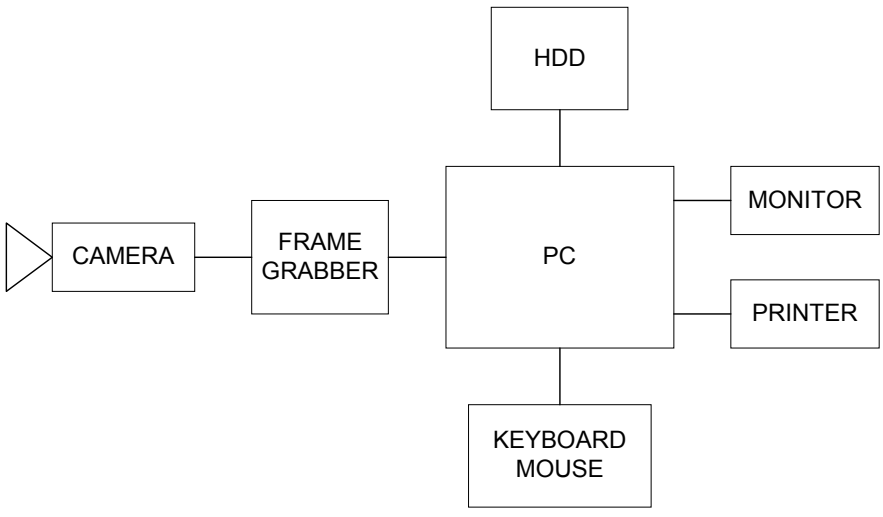


Figure 3.1.1: Block diagram of software realisation

- CAMERA

  A video camera is used as an input device. It must work in a TV system and data format that a frame grabber can process.

- FRAME GRABBER

  It is a device that transforms data from the video camera and stores data into a buffer in specified format. The frame grabber is usually a PCI extension card, which is plugged into the computer.

- PC

  It is a main part in this type of realisation. The PC takes data from the frame grabber and applies normalisation algorithms. Further, it can display, store and present results. PC enables the control application and to set parameters for a user-friendly desktop.

- MONITOR

  It displays the results of normalisation.

- KEYBOARD, MOUSE

  Users can control the application and set parameters with these devices.

- HDD

  It can be used for storing results (optional).

- PRINTER

  It enables the printing of results (optional).

The disadvantage of this method is lower speed. The speed can be increased when we will use a PC with higher performance. A multiprocessor machine is recommended.

3.2 Hardware realisation

The hardware method is important in the design of normalisation systems for some ‘real-time’ applications. These systems are fully dedicated for some normalisation methodology and are not flexible. These systems, which are designed for a ‘real-time’ application, are usually optimised for high

performance and speed. This is the main advantage of a full hardware solution.

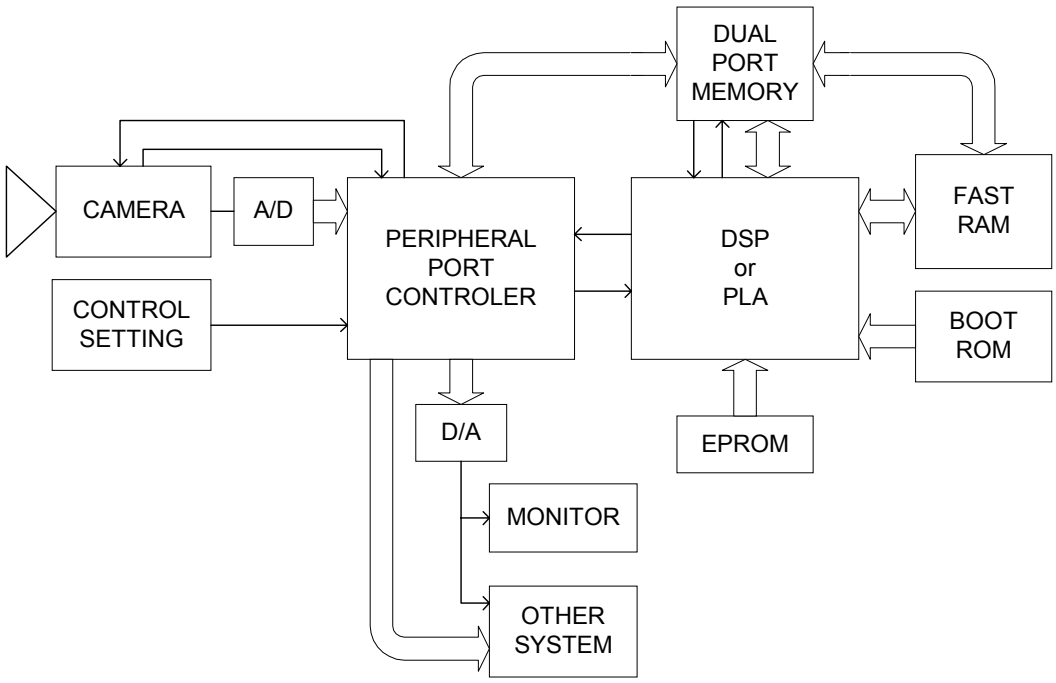


Figure 3.2.1: Block diagram of hardware realisation

- CAMERA

The CAMERA is used as the input device in this block diagram. Nevertheless, other systems can be used (line scan cameras, laser scan etc.).

- A/D

The fast A/D converter is necessary for conversion of the input analogue signal to digital data. The transfer rate is very dependent on image resolution and number of colours. For example the PAL system has the resolution of 768x576 pixels, colour in 24bit and 25 frames per second. This means a transfer rate of 32MBps.

- CONTROL, SETTING

These devices enable the control of the normalisation process and set some parameters of the system.

- PERIPHERAL PORT CONTROLER

The PERIPHERAL PORT CONTROLER controls all I/O devices in the system.

- DSP or PLA

This is a main part of the hardware realisation. The DSP or PLA controls all processes in the system and performs normalisation. It transforms an unnormalised image in input buffer to a normalised image in the output buffer utilising arithmetical operations.

- DUAL PORT MEMORY

The DUAL PORT MEMORY can be used for hi-speed data transfer without idle times and waiting states.

- FAST RAM

The FAST RAM is necessary for storing image data. It is used for input buffer, output buffer and temporary buffers during calculation of normalisation.

- BOOT ROM

The BOOT ROM memory is required when the system is started or re-booted. This part is not necessary when the system utilises a power down mode instead of turn off.

- EPROM

The EPROM memory is used for storing a LUT, which performs a mapping function.

- D/A

This application needs a fast D/A converter for obtaining an output analogue signal for monitor or other system.

- MONITOR

The monitor displays input/output data and enables the observation of the course of normalisation.

- OTHER SYSTEM

This refers to a system which utilises normalisation data. For example, an image recognition system can use it as pre-processing part. Inputs of this system can be either analogue or digital data.

The main disadvantage of a full hardware realisation is that it is very difficult to change the normalisation method.

A compromise between software and full hardware realisation is the usage of a hardware board in the computer. It can be a universal or custom board. There are two possible software modes. Firstly, the PC is the master and utilises the board for some computing operations. Secondly, the system on the hardware board is the master and utilises the PC for displaying, storing and data processing.

4 Software descriptions

4.1 Reasons for software realisation

Initially, a choice had to be made for the method of realisation. Full hardware realisation is very time-consuming and so I decided between full software realisation and a realisation with a DSP universal board as a support device of the PC. The reasons for choosing full software realisation were several. Firstly, DSP universal boards, which were available, supported a connection with the PC by serial or parallel port only. That means a very limited speed. Secondly, I had the disposal of a high performance PC. Thirdly the system was developed for future educational purposes and the presentation of normalisation methods.

4.2 Choice of software language

Choice of software language is an important area of design. We were limited in this choice because software libraries for MATROX frame grabber, which was used, supported only Visual C++ and Visual Basic. I decided for Visual C++. I used Visual C++6 with ActiveMIL-Lite 6.0.

4.3 Description of data flow in system

In this part of the chapter describes the data flow architecture of my software system. It explains the principles, which I used in my design. These data flow diagrams are not the flow diagram of software program but they are drawn in a very general way for their easy use in a hardware design.

The system has three main parts. Namely, the position normalisation module, the rotation & size normalisation module and the position & size normalisation module. An arrangement of these modules is shown in Fig. (4.3.1).

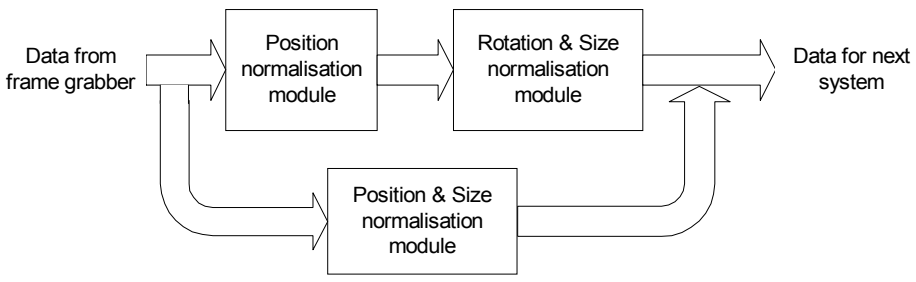


Figure 4.3.1: Data flow in system

Data in format RGB 32bit are transmitted from the frame grabber to the position normalisation module (PNM) or the position & size normalisation module (P&SNM), where they are stored in the input image buffer. An output of the PNM is an image of 256x256, which has an object in the centre. This image is stored in centralised retinal image buffer. It is input for the rotational & size normalisation module (R&SNM). The outputs of the R&SNM are several. Namely the unnormalised cortical image, the normalised cortical image and the normalised retinal image. It depends upon the settings as to which normalisation is enabled. The output of the P&SNM is normalised retinal image. All modules are detailed in the next parts of this chapter.

4.3.1 Position normalisation module

The position normalisation module is the first part of the system, if position normalisation is enabled. This module obtains an image with a size of 256x256 from an input image with a size of 768x576. An object, which is observed, is in the centre of this image. Correct calculation of the centroids is necessary for the correct function of rotational and size normalisation. The data flow architecture of the PNM is shown in Fig. (4.3.2).

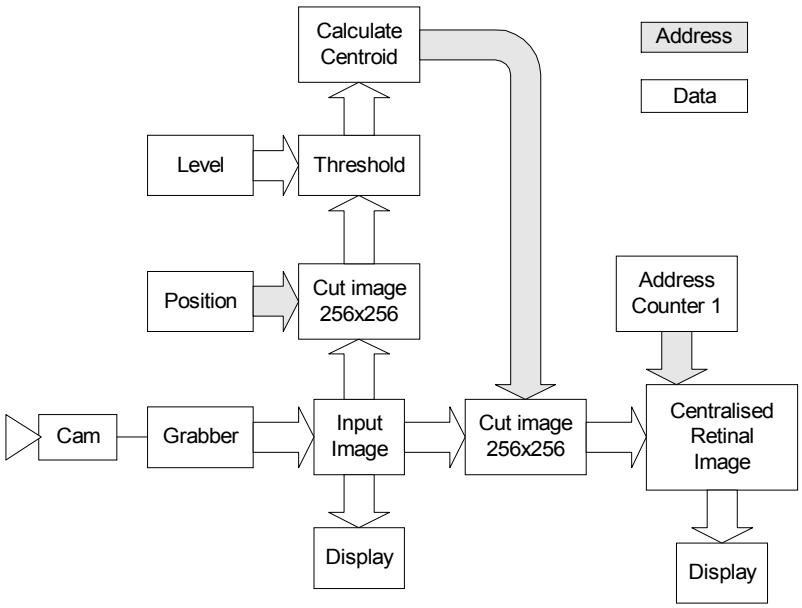


Figure 4.3.2: Data flow in Position Normalisation Module

Data in format RGB 32bit are stored in the input image buffer. The size of the input image buffer is 768x576x4bytes. This image is always displayed. From

the input image is extracted a 256x256 image according to the user setting. Then the x and y-centroid are calculated. For centroid calculation is necessary to have a black and white image. Therefore, data, when are read, are compared with the threshold and transformed from colour to black and white. The threshold is set before the start of normalisation and details about thresholding are in the next part of the chapter. When correct x and y-centroids are available, a new image from the input image buffer is extracted and stored in the centralised retinal image buffer. Linear address counter 1 is used for data storing in the buffer. It counts from 0 to 255. The centralised retinal image can be displayed.

4.3.2 Rotation & Size normalisation module

The next part of the system is the Rotation & Size normalisation module. An input of this module is the centralised retinal image which is stored in the buffer. This demonstrates that rotational and size normalisation can not work without position normalisation. Nevertheless another algorithm is used for only size normalisation. This part of the system, which performs position and size normalisation in a single architecture, is described in the next part of the chapter. The data flow architecture of the R&SNM is shown in Fig. (4.3.3).

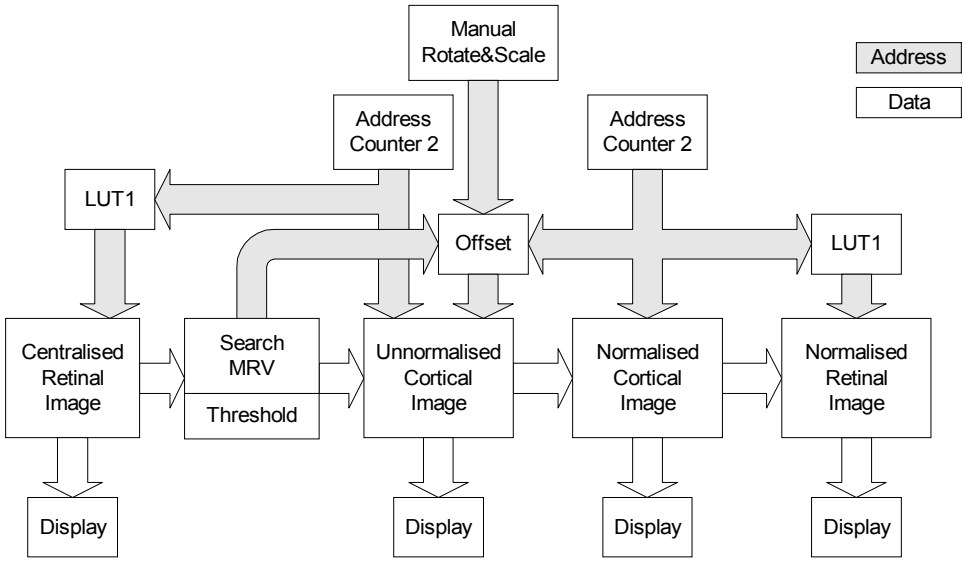


Figure 4.3.3: Data flow in Rotation & Size Normalisation Module

The image from the centralised retinal image buffer is transformed from the retinal to cortical co-ordinates and stored in the unnormalised cortical image buffer. Its size is 128x512x4bytes. The principle of retino-cortical transformation is very simple here. An address counter 2 is used for storing to the unnormalised cortical image buffer. It counts from 0 to 127 for the number of columns and from 0 to 511 for the number of rows. An address for reading from the centralised retinal image buffer is obtained from LUT1. LUT1 is generated at the start of the program. Details about generation of the LUT are in the next part of the chapter. During the transfer from the retinal to cortical image buffer, the Maximum Radius Vector (MRV) is searched. Thresholding is used in this process and the unnormalised cortical image can be displayed. A main part of this module is a transfer from the unnormalised to the normalised cortical image buffer. During this process the rotational and size normalisation is performed. The address for reading from the unnormalised image buffer is composite of the MRV address, the address from the manual rotate & scale block and the address from the address counter 2. The manual rotate & scale block allows the user to change the angle of rotation and size of the object. The address for storing the normalised image is the address from the address counter 2. A principle of the normalisation is to shift MRV to some pre-defined position (rotation normalisation) and a change a value of MRV to some pre-defined value (size normalisation). The normalised cortical image now can be displayed. The last part of this system is a conversion from the cortical to retinal image. That means an inverse retino-cortical transformation. The system uses the same principle as in the retino-cortical transformation. The address from address counter 2 is used for reading from the cortical image buffer. The address for storing the normalised retinal image buffer is obtained from LUT1 and the normalised retinal image then can be displayed. This part of the system is not usually necessary when the system is used as a pre-processing part the of recognition system because recognition systems can use the normalised cortical image. The conversion to the retinal image is performed only for user-friendly observation during normalisation.

4.3.3 Position & Size Normalisation Module

The P&SNM is the part of the system which is used when only size normalisation is enabled. That means, we can choose from two algorithms for position and size normalisation. This module uses a principle of 'Character Size and Position Normalisation', which is described in chapter "Theory of normalisation". This method is modified slightly. Only one size ratio (scale factor) is used. The size ratio is obtained from the greater side of a minimum enclosed rectangle so that the shape of an object is not deformed. The data flow architecture of the P&SNM is shown in Fig. (4.3.4).

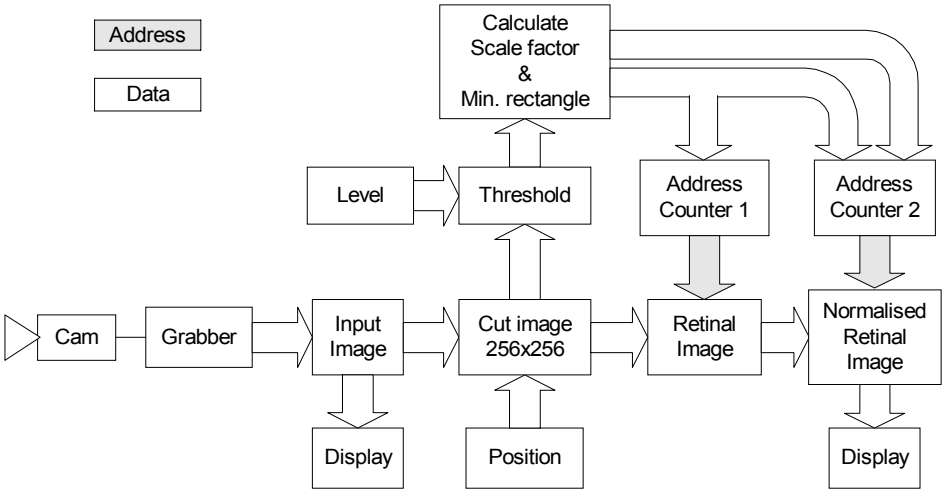


Figure 4.3.4: Data flow in Position & Size Normalisation Module

From the input image is taken a 256x256 image, according to the user setting, and then stored in the retinal image. A threshold is applied to this image for the calculation of a minimum rectangle and a scale factor. The minimum rectangle is the smallest rectangle, which includes all black pixels. Then the scale factor is calculated from the size of the minimum rectangle. The position and size normalisation are performed by appropriate setting of the address for reading from the retinal image and writing to the normalised retinal image. The reading address is generated by the address counter 1, which is dependent on the size of the minimum rectangle. The writing address is generated by the address counter 2, which is dependent on the size of the

minimum rectangle and the scale factor. The normalised retinal image then can be displayed.

4.4 Description of main software parts

This part of the chapter is dedicated to the introduction of the normalisation algorithms, which are used in the software, and to the important software parts.

4.4.1 Position normalisation

This program enables one to choose from two algorithms for position normalisation.

The first of them is used when the position normalisation check, in normalisation settings dialog, is enabled. In this algorithm, the 256x256 image is taken and stored to buffer. Then each pixel is compared with the threshold and the running sums for each row and column are calculated. The total running sum is calculated as the sum of all running sums of rows. If the total running sum is not equal to zero then the x- and y-centroids are calculated. The x-centroid is the number of the column where the running sum is greater or equal to half of the total running sum. The same principle is used for the y-centroid. When the x- and y-centroids are known then the new image, which has the object in the centre, is extracted and stored. This method solves the problem of the wrap around effect, which was in previous systems. A disadvantage is a lower rate, because the new image must be extracted and stored.

The second algorithm is used when only the size normalisation check is enabled. This algorithm covers position and size normalisation. Part of this program, which performs this algorithm, is described in the part about size normalisation.

4.4.2 Rotation normalisation

The principle of rotation normalisation is only to shift the image in the cortical co-ordinates. This is realised by a change of writing address. This change is dependent on the position of the MRV. The MRV and its position are known when the retino-cortical transformation is finished. The position of the MRV

can be modified with the slider for manual rotation. The image is only copied from the unnormalised buffer to the normalised buffer.

4.4.3 Size normalisation

Two methods for the size normalisation are available in this system.

The first is mainly intended for use with rotation normalisation. This method utilises the image in cortical co-ordinates. This image is expanded such that the size of the MRV is equal to the maximum radius. The expansion is only in one direction. First of all, the scale factor is calculated from the size of the MRV and maximum radius. Each pixel from the unnormalised buffer is read and its new position is calculated. All pixels, between the previous and present new position, are filled with the previous pixel. So the expansion is reached. When the pixels are stored to the normalised buffer, the rotation normalisation algorithm is included.

The second method can be used when rotation normalisation is not required. This method combines the position and size normalisation. Firstly, all pixels are read and compared with the threshold. The minimum and the maximum values of the x- and y- co-ordinates are stored. These values determine the minimum rectangle, which includes all black pixels. The scale factor is calculated from the size of the greater side of the rectangle and the size of a window (256 in this system). The rectangle is shifted to the left upper corner and then the algorithm from the first method is used. The algorithm is only modified for 2-D expansion. There is a possibility that reading from outside of the image (256x256) is required and, if so, then the white pixel is read.

4.4.4 Retino-cortical and inverse retino-cortical transformation

The retino-cortical transformation is performed by means of the LUT, which is generated before the start of normalisation. For each pixel in the cortical buffer is found the reading address from the LUT, and the pixel from the retinal buffer is copied to the cortical buffer. Then the pixel is compared with the threshold for MRV searching. When this part of the program is completed, the cortical image and the MRV are known. The principle of inverse retino-cortical transformation is very similar. For each pixel in the cortical buffer is

found the writing address from the LUT and the pixel from the cortical buffer is copied to the retinal buffer.

4.4.5 Threshold

A very important part of the normalisation system is the threshold. There are two places where the threshold is used in the system. Firstly, it is used during position normalisation. Thresholding is necessary in the algorithm for position normalisation because an amount of black pixels is calculated. Secondly, it is used during searching for the maximum radius vector. A correct identification of an object and a background is the main supposition of correct normalisation.

The system works with images in a RGB colour format of 24bits.This means each component of colour (R, G, B) is stored in 8 bits. There are two possible solutions of threshold. Firstly, the colour image can be converted to greyscale. The calculation of greyscale level must correspond with the theory of colours. It says that each part of colour has a weight in greyscale. It shows equation (4.1) in accordance with ITU standards. Other possible solutions are in [14].

$$GS = 0.2220R + 0.7067G + 0.0713B \tag{4.1}$$

Secondly, the green part of colour can be used for threshold. This part has the biggest weight and is sometimes compared with luminiscence [15]. This normalisation system utilises the second solution because it is not so time-consuming as the conversion to greyscale.

A manual and automatic setting of the threshold level is supported in the software. The automatic setting calculates the average value of colour from all pixels (only on green).

4.4.6 LUT generation

The LUT generation is the main part, which determines a method of mapping from the retinal to cortical co-ordinates. A change of mapping is very simple. The generation algorithm is only changed. In this system it is possible to use four types of scanning, namely, the radial scanning with linear scale, the radial circular scanning with linear, logarithmic or exponential scale. The principle of the scanning is explained in the chapter: “Theory of normalisation. “

The LUT is an array where the addresses to the retinal image buffer are stored. The array size is 128 by 512. Each field of the array is filled by x and y addresses. Reduced equations (2.4) and (2.5) calculate these addresses. Some pixels from the retinal image are not mapped. The explanation of under-sampling is in chapter 2. The result of this effect is shown in Appendix D.

4.5 Functions of the program

A list of the main functions of the normalisation part of the program is outlined in this article.

- Position normalisation (two algorithms)
- Rotation normalisation
- Size normalisation (two algorithms)
- Choice of position of the process window (256x256)
- Threshold setting – manual/auto
- Threshold preview
- Manual rotate and size
- Result images saving
- Display co-ordinates of centre
- Display angle of rotation
- Display rate

5 Results

The software system, which this project describes, is integrated with a complex software system. The complex system supports image normalisation and image recognition. These functions can be executed separately or together. The normalisation part successfully uses methods based on retino-cortical transformation and presents several types of mapping. 'Character Size and Position Normalisation' method with small change is included successfully too. A disadvantage of the system is low processing speed. The performance, which was achieved, is shown in Table (5.1). Nevertheless, the system is sufficient for presentation and demonstration of the methods employed.

5.1 Performance

The performance of the software system is very dependent on the performance of the used computer. All rates, which are mentioned, are achieved with a Pentium III 500MHz with 128MB RAM. The rates are only average, because the rates are dependent on scenes.

Check			Normalisation function	Average rate *** (frame/sec)
P	R	S		
X	X	X	Position, rotation, size	4.4 (3.1)
X	X		Position, rotation	4.9 (3.5)
X		X	Position, size *	4.4 (3.1)
X			Position	14.2 (11.3)
		X	Position, size **	10.1 (9.3)

* The method with retino-cortical transformation.
** 'Minimum enclosed rectangle' method.
*** Rate when all display windows is off (on)

Table 5.1: The performance of the normalisation system.

5.2 Examples of results

The descriptions of results and brief discussion about them are mentioned here. All the resultant images are printed in Appendix C.

- ***Position normalisation***

  Figures (C.1) and (C.2) show the achieved results when only position normalisation is enabled. Only the Position must be checked in the

normalisation settings for position normalisation. This example shows an arrow in two different positions. The normalisation dialog, where co-ordinates of the centre and performance are displayed, and the position-normalised image are printed.

- ***Position and rotation normalisation***

  Figures (C.3) and (C.4) show achieved results when position and rotation normalisation is enabled. In this case the radial circular mapping with linear scaling is used. Position, rotation and radial circular linear must be checked in the normalisation settings. This example shows an aeroplane in two positions with different angle of rotation. The position normalised image, the normalised cortical image, normalised retinal image and normalisation dialog are printed out. Co-ordinates of centre, angles of rotation and performance are displayed in the dialog. The angle of rotation is counter-clockwise.

- ***Position and size normalisation***

  Two examples for position and size normalisation are presented. In the first case, the method with retino-cortical transform and radial circular mapping with linear scaling is used. The results are shown in Figures (C.5) and (C.6). Position, size and radial circular linear must be checked in the normalisation settings. The same images as in the previous example are displayed. Co-ordinates of centre, size ratio and performance are displayed in the dialog. In the second example, the minimum enclosed rectangle method is used. The results are shown in Figures (C.7) and (C.8). Only the size must be checked in the normalisation settings for this normalisation. The same images as in the previous example are displayed.

- ***Position, rotation and size normalisation***

  Four examples for position, rotation and size normalisation are presented here. These examples use radial mapping with linear scaling, radial circular scanning with linear and logarithmic scaling. Exponential mapping is not presented because this method is not suitable for use in the future. All normalisation must be checked in the normalisation settings. The

position normalised image, the normalised cortical image, normalised retinal image and normalisation dialog are printed out. Co-ordinates of centre, angle of rotation, size ratio and performance are displayed in the dialog. Results with radial mapping with linear scaling are shown in Figures (C.9) and (C.10). Figures (C.11), (C.12) and (C.13), (C.14) show the results with radial circular mapping with linear and logarithmic scaling. We can observe that the shape is deformed when circular mapping is used and when the size ratio is higher. The last example shows that the system works with colour objects. It is shown in Figures (C.15) and (C.16).

- ***Manual rotate and size***

  Figure (C.17) shows that manual rotation and size variation is possible.

- ***Image recognition with normalisation***

  There are two examples of the integrated normalisation system with the image recognition system. The first example is shown in Figures (C.18), (C.19) and (C.20). Position, rotation and size normalisation are used here. The second example is shown in Figures (C.21), (C.22) and (C.23). The normalised images, input to the image recognition system in RGB parts and the Train and recognition dialog are printed out.

6 Possible solutions

In this chapter are some other methods, which can be used for image normalisation.

One of the important methods is a usage of Fourier Transform techniques in the normalisation process. In principle, the Fourier Transform of an object is position-invariant. Unfortunately, it is not rotation-invariant. A further technique must be used for rotation invariance. The Mellin Transform can be utilised. It is very similar to the retino-cortical transform with logarithmic scaling. The main advantage of Fourier Transform techniques is that, if implemented optically, 'real-time' operation is easily obtained.

All methods, which are used in this diploma project, assume only one object in the image. When more objects are in the image, the results are wrong and other algorithms must be used. Edge detection is usually used in multi-object tracking systems. It enables the detection of more objects and divides the image window into several sub-windows. Then the normalisation system works with these sub-windows separately. Edge detection can be utilised for position normalisation and searching for the maximum and minimum radius vector.

A very important part of the normalisation system is thresholding. Most systems use an automatic threshold because results are very dependent on it. Usage of histogram stratching is possible. Some systems find the threshold as the minimum between two main maximums of the histogram.

At present, a usage of smart cameras in image processing is increasing. Many companies develop their systems with 'smart' cameras. Several approaches are known. The publication [1] outlines the approaches of some companies in 1999. Some of them are outlined here.

Firstly, a retina-like camera can be used. The photosites are arranged specially in this camera. The distribution of resolution is very similar to that of human vision. The camera has higher resolution in the central region and lower in the peripheral region. An example of their arrangements is shown in appendix F.

Secondly, smart cameras, where DSPs are incorporated. These cameras can have one or more DSP. The companies use different strategies. Some of them use the cameras with one DSP per column of photosites. The main advantage of this solution is very high system performance. The others use more cameras, which are not so smart, and improve their connecting capabilities. Their systems communicate by fieldbus (e.g. CAN/DeviceNet). The cameras can be formed into slightly overlapping chains to cover a wide web at high resolution. That means a single extremely long sensor can be assembled from these cameras. This is one main advantage of this strategy.

7 Further work

Some suggestions for further work are outlined here. There are two main ways for further development.

The first of them is the improvement and optimisation of the software solution and the incorporation of further normalisation methods. Of importance, the threshold should be improved because the results are very dependent on the correct threshold. Usage of the Active MIL library - full version can improve the software realisation because Active components for image processing and pattern matching are included. This can solve the problem with thresholding and enhance the performance. It is possible to try direct access to the image memory for achieving a higher processing speed. The rotation normalisation should be modified because several maximum radius vectors in an image can perform wrong normalisation. For example, the black pixel count in any maximum radius vectors can be tested. Implementation of a multi-object-tracking module to the system is important in further work because all methods, which are presently used, support only one object in processed window.

The second suggestion proposes the development of a hardware system utilising the methods which were tested in this project. The main task of this way should be the realisation of a device which will work real-time. Modern components can be used and tested in this application. It is possible to use a universal board for DSP or PLA. This board should support a PCI bus for communication with a PC. A connection with serial or parallel port is not suitable.

8 Requirements and Recommendations

Hardware

- PC with PCI bus and a Pentium processor
- 128MB RAM
- 850 MB HDD
- SVGA colour monitor, resolution 1280x1024
- MATROX Meteor/RGB – frame grabber (see Appendix G)
- MATROX Mystique G200 – graphics card
- Video camera with PAL system and RGB output

Software

- Windows NT4.0
- Visual C++ 6.0
- ActiveMIL – lite library

Note:

Full version of ActiveMIL library is recommended for further work because many functions for image processing and pattern matching are included in this library. These functions are developed and optimised for using with the MATROX card and so higher rates can be achieved.

9 Conclusions

The software realisation of the image processing system was developed and successfully linked together with the image recognition system [8]. The results indicate that software realisation is suitable for the presentation and testing of the normalisation techniques, but it is not suitable for real-time applications. The case of specialised hardware is necessary for applications where a high processing speed is required.

Some techniques of position, rotation and size normalisation were tested. Two types of mapping were used. The best results were reached with radial linear mapping because a shape of an object is not deformed when size normalisation is used. The change of shape is the main problem of radial circular mapping. The radial circular mapping is better when size normalisation is disabled because the radial linear mapping sometimes has errors in rotation normalisation. Three types of scaling were used in radial circular mapping. They vary in area where the pixels are under-sampled. The logarithmic scaling is more similar to human vision. It has higher resolution in the central area and lower in the foveal region. The exponential scaling was tested but it is not suitable for further work. It can be used only in the case where size normalisation is applied before the mapping.

D. Under-sampling

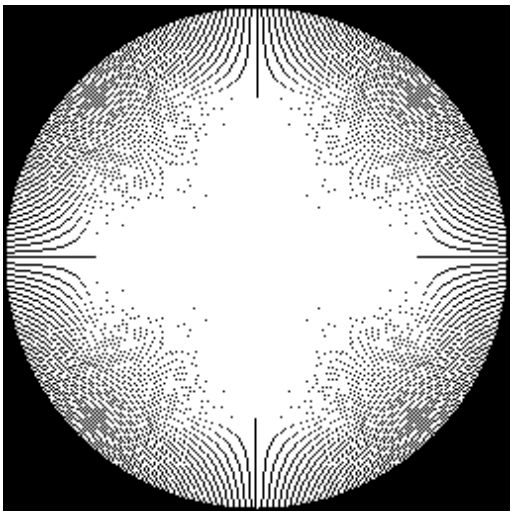


Figure D.1: Scan map for Radial Linear Mapping

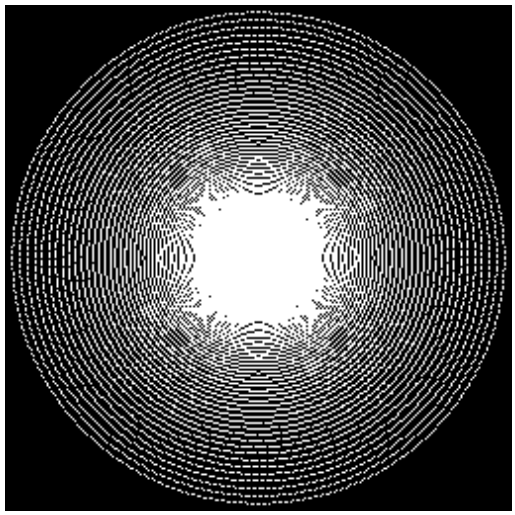


Figure D.3: Scan map for Radial Circular Mapping - logarithmic

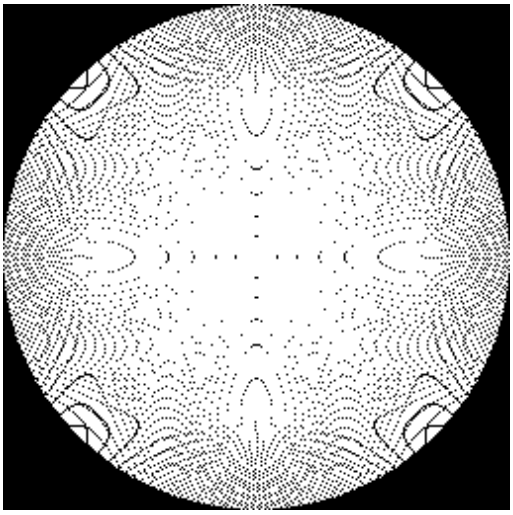


Figure D.2: Scan map for Radial Circular Mapping - linear

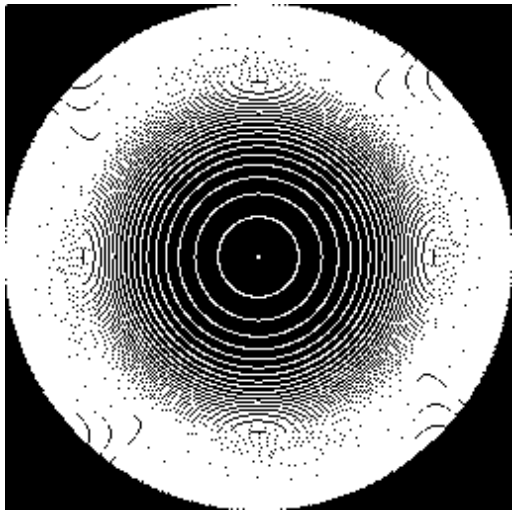


Figure D.4: Scan map for Radial Circular Mapping - exponential

E. Retino-cortical transform

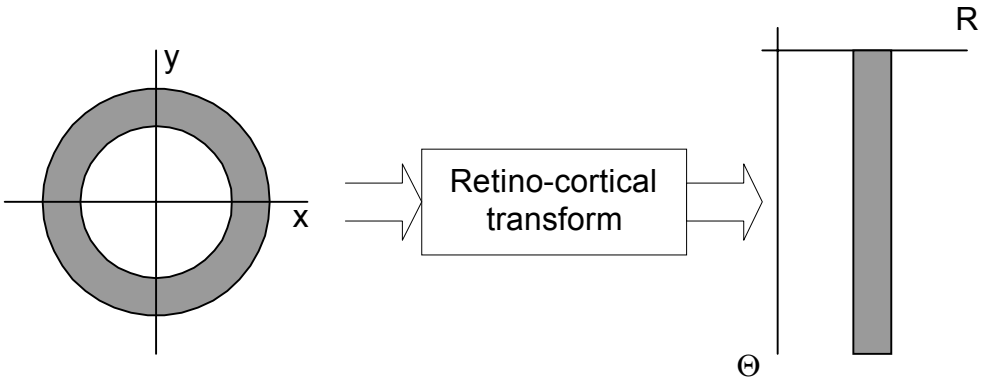


Figure E.1: Example of retino-cortical transform.

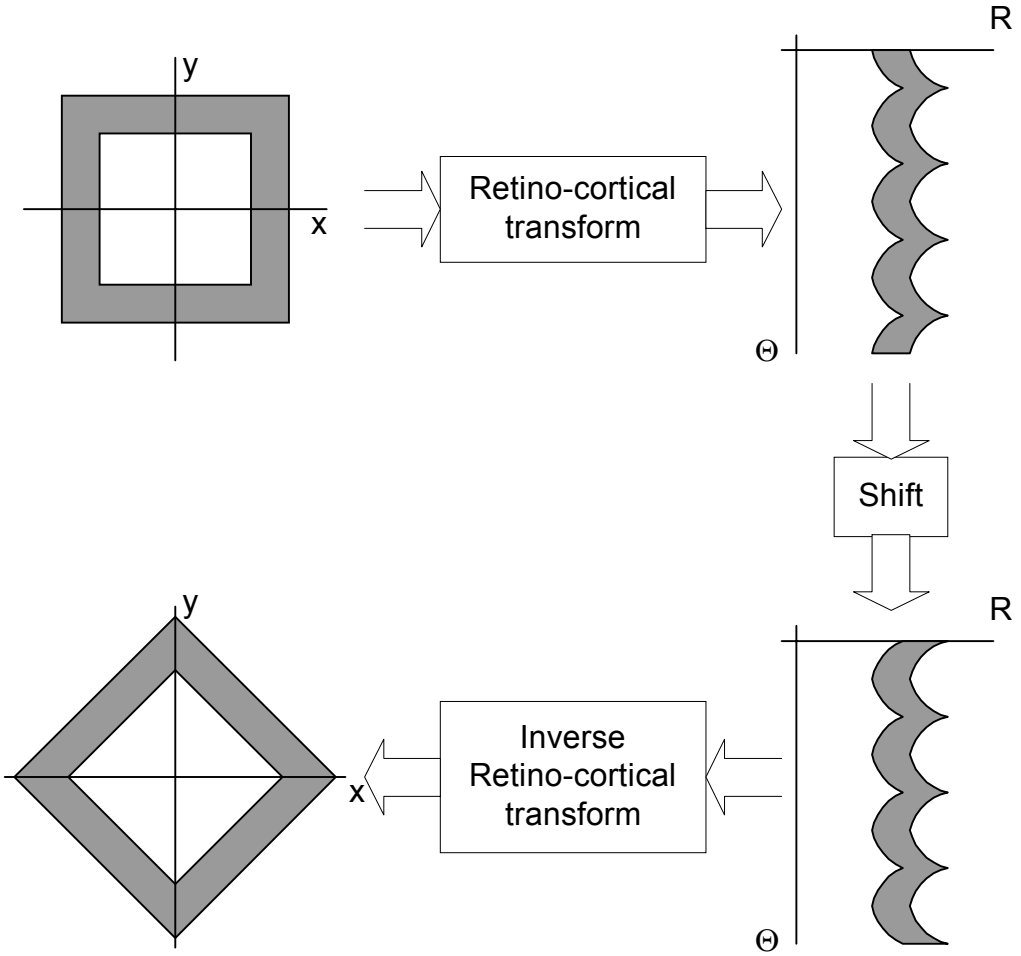


Figure E.2: Usage of retino-cortical transform in the rotation normalisation.

F. Retina-like camera

Figures (F.1) and (F.2) show an arrangement of photosites in the first commercially available retina-like colour camera. Italian consortium Unitek was showing it at the IST Conference and Exhibition in Vienna in December 1998. This camera has only 8000 photosites but the central area produces a resolution comparable with a conventional camera having 360 000 photosites.

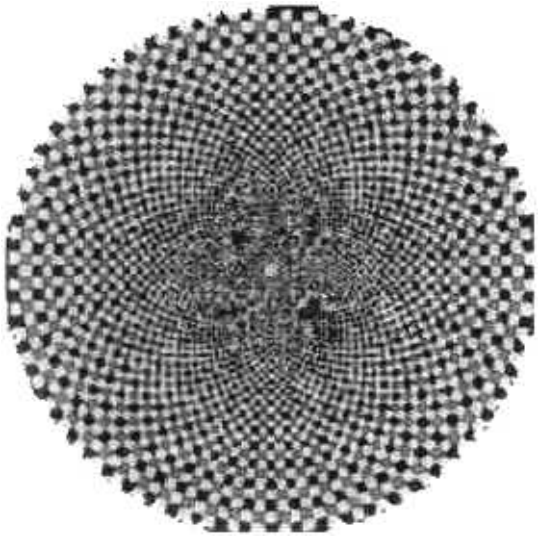


Figure F.1: The colour filter layout used in the retina-like camera

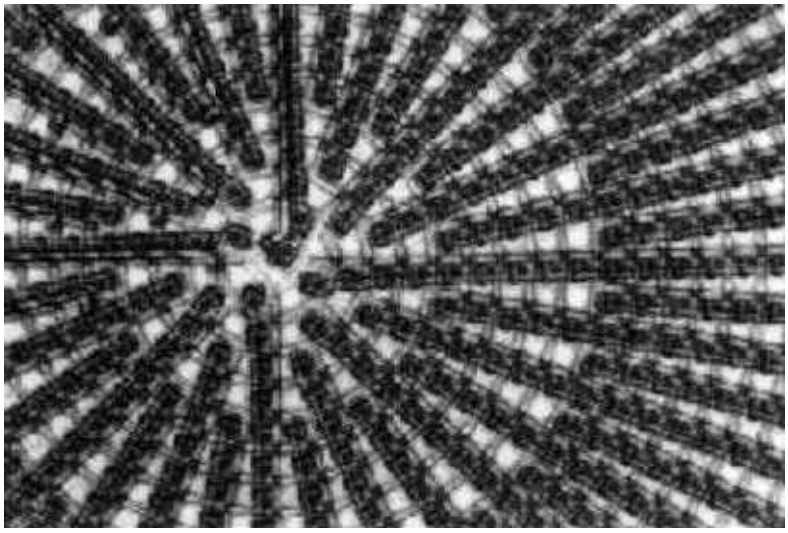


Figure F.2: The central part of the sensor of the retina-like camera

G. MATROX METEOR – flow diagrams

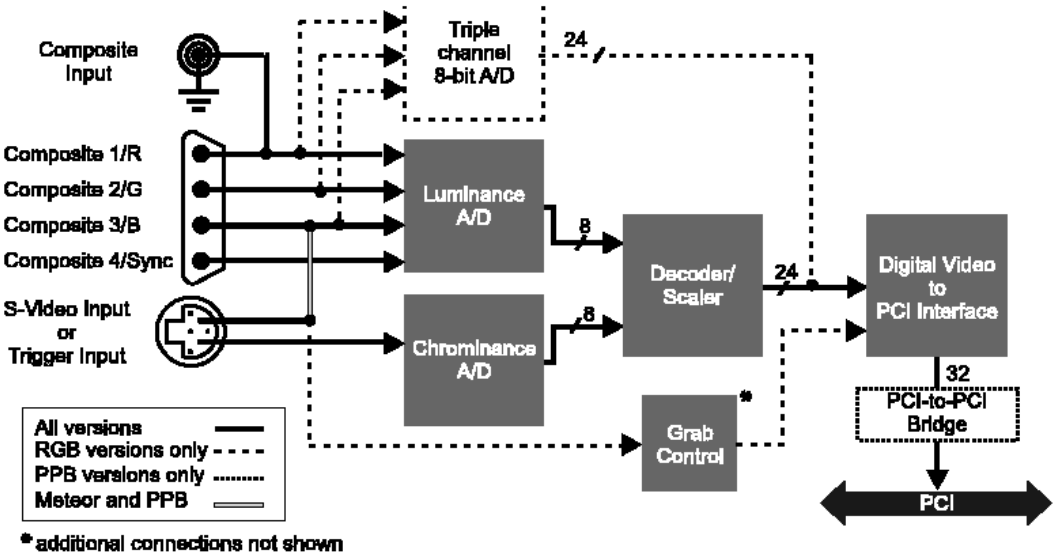


Figure G.1: The flow diagrams of MATROX METEOR boards

References

[1] Braggins, D.: “Inteligent cameras report”, IIP Newsletter, June 1999, p.3.

[2] Horton, I.: “Beginning Visual C++ 6”, Wrox Press Ltd, 1999, ISBN 1-861000-88-X

[3] Luckin, M.: “Implementation of a Position Normalising Module and Investigation of Radial Scanning”, Final year report, Dept. of E.E.& E., Brunel University, 1992.

[4] May, D.S.: “A prototype ‘radial scanning’ image processing system”, Final year report, Dept. of E.E.& E., Brunel University, 1990.

[5] Munro, A.: “Real –time position normalisation”, Final year report, Dept. of E.E.& E., Brunel University, 1991.

[6] Reichert, G.: “Implementation of a rotation normalisation image preprocessor”, M.Sc. dissertation, Dept. of E.E.& E., Brunel University, 1990.

[7] Schofield, A.: “Character Size and Position Normalisation to Aid Recognition”, 4th Year Project, Dept. of E.E.& E., Brunel University, 1990.

[8] Trejbal, J: “Software realisation of colour pattern recognition system using n-tuple and min/max node neural networks”, Diploma project, Dept. of Applied Sciences, University of West Bohemia, 2000.

[9] Walker, C.W.M.: “Real –time object characterisation system”, M.Sc. dissertation, Dept. of E.E.& E., Brunel University, 1990.

[10] Wilkie, B.A. and Stonham T.J.: “Position, rotation and size normalisation of TV images at half frame rate”, Internal report, Dept. of E.E.& E., Brunel University, 1991.

[11] Wilkie, B.A.: “Position, rotation and size normalisation and the classification of single objects within three frame periods”, Internal report, Dept. of E.E.& E., Brunel University, 1997.

[12] Wilkie, B.A.: “The Position, Size and Rotation Normalisation of a Single Object within an Image at TV Frame Rate”, Proceedings of The International Conference on Applied Electronics ‘97’, University of West Bohemia, Pilsen, Czech Republic, May 1997, pp 208-224, ISBN 80-7082-337-2.

[13] Wilkie, B.A.: “Object Recognition after Normalisation for Position, Size and Rotation”, Proceedings of The International Conference on Applied Electronics ‘97’, University of West Bohemia, Pilsen, Czech Republic, May 1997, pp 200-207, ISBN 80-7082-337-2.

[14] Colour quantizations. http://home.earthlink.net/~janos1/vidpage/color_faq.html#spaceconv

[15] Glicksman, H.: White is green. http://www.csulb.edu/~percept/kyotocolor.html, http://www.csulb.edu/~percept/imaging99.html