Hardware Implementation of the Hough Technique for Irregular Colour and Grey-level Pattern Recognition

Witold ŻORSKI, Andrzej ŻAK, Martin TURNER*

Institute of Automation and Robotics, Military University of Technology S. Kaliskiego 2, 00-908 Warsaw, POLAND

> *Institute of Simulation Sciences, De Montfort University The Gateway, Leicester LE1 9BH, ENGLAND

Abstract. This paper presents a hardware implementation of the Hough technique applied to the tasks of irregular colour and grey-level pattern recognition. The presented method is based on the Hough Transform with a parameter space defined by translation, rotation and scaling operations. An essential element of this method is the generalisation of the Hough Transform for grey level and colour images. The technique simplifies the application of the Hough Transform to irregular pattern recognition tasks. The hardware implementation accelerates the calculations considerably and may be used in computer vision systems, for example, in a robotic system.

Keywords: Hough Transform, computer vision, hardware implementation, irregular patterns.

1. Introduction to the Hough Transform

The Hough Transform was patented in 1962 as a method for detecting complex patterns of points in a binary image [8]. It introduced the possibility of determining a set of parameters circumscribing the searched pattern. The problem of complex pattern detection in an image is converted into one that searches for local maxima in a parameter space. This method has become very popular.

In 1981 Deans noticed [5] that the Hough Transform for straight lines was a specific case of the more general Radon Transform [15] known since

1917, which is defined as (for function I(x, y) in two-dimensional Euclidean space):

$$H(\rho, \alpha) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) \,\delta(\rho - x \cos(\alpha) - y \sin(\alpha)) dx dy \,, \tag{1}$$

where δ is the delta function. This result shows that the function I(x, y) is integrated along the straight line determined by the parametric equation $\rho = x \cos(\alpha) + y \sin(\alpha)$. The Radon Transform is equivalent to the Hough Transform when considering binary images (i.e. when the function I(x, y) takes values 0 or 1). The Radon Transform for shapes other than straight lines can be obtained by replacing the delta function argument by a function, which forces integration of the image along contours appropriate to the shape.

Using the Radon Transform to calculate the Hough Transform is simple (almost intuitive) and is often applied in computer implementations. We call this operation **pixel counting** in the binary image.

An (alternative) interpretation of the Hough Transform is the so-called **backprojection** method. The detection of analytical curves defined in a parametrical way, other than straight lines is quite obvious. Points (x, y) of image lying on the curved line determined by n parameters $a_1,...,a_n$ may be presented in the form:

$$\lambda_{O} = \{ (x, y) \in \mathbb{R}^{2} : g((\hat{a}_{1}, ..., \hat{a}_{n}), (x, y)) = 0 \},$$
(2)

where $g((\hat{a}_1,...,\hat{a}_n),(x,y)) = 0$ describes the given curve.

By exchanging the meaning of parameters and variables in the above equation we obtain the backprojection relation (mapping image points into parameter space), which may be written down in the following way:

$$\lambda_{T} = \{ (a_{1},...,a_{n}) \in \mathbb{R}^{n} : g((\hat{x},\hat{y}),(a_{1},...,a_{n})) = 0 \}.$$
(3)

From equation (3) the Hough Transform $H(a_1,...,a_n)$ for the image I(x,y) is defined as follows [9]:

$$H(a_{1},...,a_{n}) = \sum_{(x_{i},y_{i})\in B} h(\hat{x}_{i},\hat{y}_{i},a_{1},...,a_{n}), \qquad (4)$$

where

$$h(\hat{x}_{i}, \hat{y}_{i}, a_{1}, ..., a_{n}) = \begin{cases} 1 & \text{if } g((\hat{x}_{i}, \hat{y}_{i}), (a_{1}, ..., a_{n})) = 0\\ 0 & \text{otherwise} \end{cases}$$
(5)

In order to calculate the Hough Transform digitally an appropriate representation of the parameter space $H(a_1,...,a_n)$ is required. In a standard

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

implementation, any dimension in the parameter space is subject to quantisation and narrowing to an appropriate range. As a result, an array is obtained where any element is identified by the parameters $(a_1,...,a_n)$. An element in the array is increased by 1 when the analytical curve, determined by co-ordinates $(a_1,...,a_n)$, passes through point (\hat{x}, \hat{y}) of the object in image 1. This process is called **accumulation** and the array used is called an **accumulator** (usually marked with a symbol A).

Thus, we may assume that the Hough Transform is based on a representation of the image I into the accumulator array A, which is defined as follows:

$$A: P \to N$$
, where $P = P_1 \times P_2 \times \cdots \times P_p$. (6)

The symbol $P_i \subset N$ determines the range of i -parameters of a p-dimensional space P. Determining array A is conducted through the calculation of partial values for points of an object in image I and adding them to the previous ones (see 4) which constitutes a process of accumulation. Initially, all elements of array A are set to zero.

This paper presents a hardware implementation of the Hough technique to the tasks of irregular colour and grey-level pattern recognition. It is based on the Hough Transform with a parameter space defined by translation, rotation and scaling operations. A fundamental element of this method is the generalisation of the Hough Transform for grey-level and colour images. The Generalised Hough Transform has been previously described in detail in [19]. Nevertheless a short introduction to the technique is given in this paper.

2. The Binary Hough Transform for irregular objects

Basic definitions

Let us consider binary **digital images**, i.e. images, which are formed with sets of points, which by convention are either black or white. Such binary images may be represented with the following function:

$$I_{B}: D \to \{0,1\}, \text{ where } D = [1,...,W] \times [1,...,K] \subset N^{2}.$$
 (7)

Hence, we may consider a digital image as a matrix with row and column indices identifying a point in the image.

Given an image I_B , we can define an object $b(I_B)$ as follows:

$$b(I_B) = \{ (x, y) \in D : I_B(x, y) = 1 \}.$$
 (8)

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

The Binary Hough Transform

The Hough Transform may be successfully applied to detect irregular objects [3],[9]. In the Generalised Hough Transform, an object is represented by a pattern, which is a list of boundary points $\{(x_i, y_i): i = 1, ..., n\}$ (without a concrete analytical description), and the parameter space is defined for translation $[x_T, y_T]$, rotation α and scale s of the pattern in the image.



Figure 1. Scaling, rotation and translation of pattern M_P with respect to an arbitrary point (x_r, y_r)

Figure 1 shows rotation with respect to an arbitrary point (x_r, y_r) of a given pattern by an angle α with scaling determined by $s = d_2/d_1$. As a result, a given point (x_i, y_i) of the pattern M_P is transformed into (x'_i, y'_i) where the relationships are as follows:

$$\begin{cases} \mathbf{x}'_{i} = \mathbf{x}_{r} + \mathbf{s}(\mathbf{x}_{i} - \mathbf{x}_{r})\cos(\alpha) - \mathbf{s}(\mathbf{y}_{i} - \mathbf{y}_{r})\sin(\alpha) \\ \mathbf{y}'_{i} = \mathbf{y}_{r} + \mathbf{s}(\mathbf{x}_{i} - \mathbf{x}_{r})\sin(\alpha) + \mathbf{s}(\mathbf{y}_{i} - \mathbf{y}_{r})\cos(\alpha). \end{cases}$$
(9)

When a translation operation by a vector $[x_T, y_T]$ is carried out, we have

$$\begin{cases} x_{i}'' = x_{i}' + x_{T} = x_{r} + s(x_{i} - x_{r})\cos(\alpha) - s(y_{i} - y_{r})\sin(\alpha) + x_{T} \\ y_{i}'' = y_{i}' + y_{T} = y_{r} + s(x_{i} - x_{r})\sin(\alpha) + s(y_{i} - y_{r})\cos(\alpha) + y_{T}. \end{cases}$$
(10)

Each point of the image generates an appropriate hypersurface, as a result of backprojection, in parameter space. A number of hypersurfaces that criss-cross a given point (x_T, y_T, α, s) of the parameter space is equivalent to a

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

number of points common for a given object in the image and the fitting pattern. In this case the Hough Transform $H(x_T, y_T, \alpha, s)$ for an image $I_B(x, y)$ is defined as:

$$H(x_{T}, y_{T}, \alpha, s) = \sum_{i=1}^{n} h(x_{i}, y_{i}, x_{T}, y_{T}, \alpha, s), \qquad (11)$$

where

$$h(x_i, y_i, x_T, y_T, \alpha, s) = \begin{cases} 1 & \text{if } (x_i'', y_i'') \in b(I_B) \\ 0 & \text{otherwise.} \end{cases}$$
(12)

Fortunately, many problems do not include the issue of pattern scaling. However, direct implementation of equation (11) is inadvisable when scaling must be taken into consideration.

The Hough Transform operation applied directly to irregular objects, which do not require pattern scaling, may be illustrated by means of an example. The parameter space P is defined as:

$$P = P_1 \times P_2 \times P_3 = [1,...,W] \times [1,...,K] \times [0,...,L-1], \quad (\Delta \alpha = \frac{2\pi}{L}).$$
(13)

The example task (Figure 2) is to search for a particular fragment of the motif as indicated by M. Special attention should be paid to the content of the accumulator. There are many local maxima and there are a larger number of objects closely matching the pattern in the image.



Figure 2. Hough Transform for complex image

3. Generalisation of the Hough Transform for grey-level images

Many existing algorithms apply the Hough transformation to only binary images. Observe that in the case of analysing grey-level images the

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

process of binarisation can loose important information. The problem lies in the process of accumulation. At the beginning let us try to write the equation (12) in a different variation:

$$h(x_{i}, y_{i}, x_{T}, y_{T}, \alpha, s) = \begin{cases} 1 & \text{if } I_{B}(x_{i}'', y_{i}'') = M_{P}(x_{i}, y_{i}) \\ 0 & \text{otherwise,} \end{cases}$$
(14)

where M_P is the image of the pattern (the task is to identify pattern with an object in image I_P).

Equation (14) suggests the idea of modifying (12) in the following way:

$$h(x_{i}, y_{i}, x_{T}, y_{T}, \alpha, s) = 1 - \left| I_{B}(x_{i}'', y_{i}'') - M_{P}(x_{i}, y_{i}) \right|.$$
(15)

This form tells us what to do in the case of grey-level images. However, we must first define the concept of a grey-level image, an object appearing in such an image and the concept of a grey-level pattern in a computer vision system.

Definitions

An image with 256 grey levels means a set of points, which have a value or "shade" from the set {0,...,255}. Such an image may be presented as:

$$I_G: D \to \{0,...,255\}, \text{ where: } D = [1,...,W] \times [1,...,K] \subset N^2.$$
 (16)

Object $b(I_G)$ in image I_G means any fragment of that image which may be recorded in terms of

$$Q_{G}: D_{Q} \rightarrow \{0, K, 255\}$$
, where: $D_{Q} \subset D = [1, ..., W] \times [1, ..., K] \subset N^{2}$. (17)

Pattern M_P defines an image (square matrix) of size $N_P \times N_P$ which is

$$M_{P}: D_{P} \rightarrow \{0,...,255\}, \text{ where: } D_{P} = [1,...,N_{P}] \times [1,...,N_{P}] \subset N^{2}.$$
 (18)

Generalisation of the Hough Transform

The Hough Transform $H(x_T, y_T, \alpha, s)$ (which takes into account translation, rotation and scaling) for image $I_G(x, y)$ (see 16) in the process of identifying pattern M_P determined by (18) may be defined as

$$H(x_{T}, y_{T}, \alpha, s) = \sum_{(x_{i}, y_{i}) \in M_{P}} h(x_{i}, y_{i}, x_{T}, y_{T}, \alpha, s) , \qquad (19)$$

where

$$h(x_{i}, y_{i}, x_{T}, y_{T}, \alpha, s) = 255 - \left| I_{G}(x_{i}'', y_{i}'') - M_{P}(x_{i}, y_{i}) \right|,$$
(20)

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

and the values x_i'', y_i'' are calculated from

$$\begin{cases} \mathbf{x}_{i}^{"} = \mathbf{x}_{r} + \mathbf{s}(\mathbf{x}_{i} - \mathbf{x}_{r})\cos(\alpha) - \mathbf{s}(\mathbf{y}_{i} - \mathbf{y}_{r})\sin(\alpha) + \mathbf{x}_{T} \\ \mathbf{y}_{i}^{"} = \mathbf{y}_{r} + \mathbf{s}(\mathbf{x}_{i} - \mathbf{x}_{r})\sin(\alpha) + \mathbf{s}(\mathbf{y}_{i} - \mathbf{y}_{r})\cos(\alpha) + \mathbf{y}_{T}. \end{cases}$$
(21)

Application of the histogram function

To improve this elaborated method we wish to find a characteristic of the pattern that is invariant under rotation. The histogram is the obvious characteristic especially for diverse images (of 256 grey levels). The histogram of pattern M_P is determined once only and compared with the histograms of fragments of image I_G , determined at all possible locations of the pattern M_P . A histogram of a grey-level image defines a function that maps for any grey level (from 0 to 255) the number of image pixels that have that level and may be denoted by

$$\Phi:\{0,...,255\} \to \mathsf{N} . \tag{22}$$

<u>ALGORITHM</u> – histogram analysis

<u>Step 1</u>: Determine histogram Φ_{P} of the identified pattern M_{P} .

- <u>Step 2</u>: Determine histograms $\Phi_{Q(i,j)}$ for all fragments $Q_G^{I_G(i,j)}$ of size $N_P \times N_P$ of image I_G where $i = 1,..., W - N_P + 1$ and $j = 1,..., K - N_P + 1$.
- <u>Step 3</u>: Compare the received histogram $\Phi_{Q(i,j)}$ with the histogram Φ_P using the following value

$$d_{(i,j)} = \frac{4}{\pi \cdot N_{P}^{2}} \sum_{k=0}^{255} \left| \Phi_{Q(i,j)}(k) - \Phi_{P}(k) \right|,$$
(23)

where the factor $\pi/4$ results from the relation of a circle area inscribed into a square.

<u>Step 4</u>: If $d_{(i,j)}$ is higher than a threshold value $d_{treshold}$ then it is excluded when calculating the accumulator array A.

This simple method reduces the complexity in terms of the calculation performed for the whole process (often by more than 50%).

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

4. The Hough Transform and the scaling problem

Taking into consideration pattern scaling adds an extra dimension to the parameter space. However, because the scale range is commonly known and it is often not too large, only a few values of scale factor s are enough to achieve the process of identification.

If we assume that the following n values of the scale s factor must be taken into consideration

$$\xi_1,...,\xi_n\,,\qquad\qquad n\in N\,,\qquad\qquad (24)$$

then the parameter space may be determined in the following way:

$$P = P_1 \times P_2 \times P_3 \times P_4 = [1,...,W] \times [1,...,K] \times [0,...,L-1] \times [\xi_1,...,\xi_n], \quad (\Delta \alpha = \frac{2\pi}{L}).$$
(25)

In order to accelerate calculations (applying the histogram study) the set of patterns

$$\{M_{P}^{1},...,M_{P}^{n}\},$$
 (26)

must be generated first by scaling a given pattern M_P within a range determined by values $\xi_1,...,\xi_n$. The pattern localisation process for any pattern formed from the set (26) can then be applied. Such an approach can drastically reduce the number of calculations required.

However, this method has one disadvantage that results from having to calculate histograms for an initial image n times (as the size of each pattern is different). A solution is to create a new set of patterns $\{\overline{M}_{P}^{1},...,\overline{M}_{P}^{n}\}\$ of the same size but without losing information connected with the scale of patterns $M_{P}^{1},...,M_{P}^{n}$. Note that the size of pattern M_{P}^{1} is $N_{P}^{1} \times N_{P}^{1}$. Appropriate patterns $\overline{M}_{P}^{1},...,\overline{M}_{P}^{n}$ from patterns $M_{P}^{1},...,M_{P}^{n}$ can be obtained by separating their central part of size $N_{P}^{1} \times N_{P}^{1}$. As a result, we have patterns that "remember" the scale they were created at, but are of the same size. A graphical illustration of this process is shown in Figure 3.

Since the received patterns are of the same size, it is sufficient to calculate their histograms once and compare them with the calculated histogram of the initial image. As the size of the patterns decreases, the time to create the histograms is reduced. Decreasing the patterns size also results in shortening the CPU time for the accumulator calculation. Unfortunately, patterns $\overline{M}_{P}^{1},...,\overline{M}_{P}^{n}$ carry less information than patterns $M_{P}^{1},...,M_{P}^{n}$.

Biuletyn Instytutu Automatyki i Robotyki, 17/2002



Figure 3. Graphical illustration of the process of patterns $\overline{M}_{P}^{1}, \dots, \overline{M}_{P}^{n}$ creation

5. Generalisation of the Hough Transform for Colour Images [18]

The first approach

The question arises - what to do in the case of coloured images? Because any pixel of a coloured image has three components (R, G, B) there is a problem with adopting the Hough Transform to identify objects in coloured images. The simplest method is to convert the initial image I (TrueColor, HighColor or binary) to a grey-level image I_G . Such a solution may be implemented in the way shown in Figure 4.



Figure 4. The simplest way of conduct with colour images

Within most current computer technologies we consider TrueColor images (8 bits per RGB component – i.e., a pallet of 16,777,216 possible colours for each image pixel). Conversion of such an image to grey levels may be done through the projection of points in an RGB cube onto its diagonal (combining whiteness and blackness). There are no crucial calculation problems with converting any image (as regards colouring) to the form of 256 grey levels. There is nothing new in this approach to the problem, nevertheless in many applications it will be an acceptable solution because of its simplicity.

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

The second approach

The second approach to the problem of a colour pattern location in a colour image is based on calculating the distance between $M_P(x_i, y_i)$ and $I(x'_i, y'_i)$ with the following formula (see 20):

$$\left| I(x'_{i}, y'_{i}) - M_{P}(x_{i}, y_{i}) \right| = \sqrt{(r_{I} - r_{M})^{2} + (g_{I} - g_{M})^{2} + (b_{I} - b_{M})^{2}}, \quad (27)$$

i.e. Euclidean distance between two points (r_1, g_1, b_1) and (r_M, g_M, b_M) in the RGB cube. Unfortunately, this is often not acceptable due to its high computational complexity.

The third, final approach

The third approach assumes that an input image and a pattern are given with 256-colour depth. Then it is easy to calculate the distance between $M_P(x_i, y_i)$ and $I(x'_i, y'_i)$ by pre-calculating a square matrix M_d , that includes all possible distances between any two of 256 base colours. The size of the matrix is 65,536 integer cells, which is acceptable in terms of memory requirement. Such a matrix solves the problem of calculating a distance as well as accelerates the calculations considerably. It allows us to obtain a distance in the following way (compare 20 and 27):

$$|I(\mathbf{x}'_{i}, \mathbf{y}'_{i}) - M_{P}(\mathbf{x}_{i}, \mathbf{y}_{i})| = M_{d}[I(\mathbf{x}'_{i}, \mathbf{y}'_{i}), M_{P}(\mathbf{x}_{i}, \mathbf{y}_{i})].$$
(28)

Nevertheless another problem appears, i.e. the problem of creating matrix $\rm M_d$. There are many colour models for example: RGB cube, CMY or single-hexcone HSV. Thus it is necessary to choose the colour model first and then establish the colour quantisation method, and then finally calculate all possible distances to be stored in matrix $\rm M_d$.



Figure 5. Object location in a colour image obtained from HST

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

6. Hardware implementation of the Hough technique [23]

This section describes a hardware implementation of the Generalised Hough Transform (introduced in section 3) based on a powerful development board made by Altera[®].

Introduction to the Hardware [22]

The Altera system-on-a-programmable-chip (SOPC) board is a development and prototyping platform that provides system designers with an economical solution for hardware verification. Figure 6 shows a picture of the Altera SOPC board with the section of APEX device being zoomed in.



Figure 6. Altera SOPC board (the APEX device is zoomed in)

The board supports a variety of microprocessor-based designs, incorporating memory, debugging, and interface resources. The development board has been primarily designed for implementing microprocessor functions and other standard intellectual property (IP) functions in the on-board APEX[™] EP20K400E 652-pin device. The APEX device features 423,000 ASIC-equivalent gates with 16,640 logic cells and 212,992 bits of RAM.

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

The board includes physical interfaces for widely used standard interconnects as follows:

- 10/100 Ethernet with full and half duplexing
- peripheral component interconnect (PCI) mezzanine connector
- high and low-speed universal serial bus (USB) host supporting
- IEEE Std. 1394a FireWire interface at 100, 200 and 400 Mbps
- IEEE Std. 1284 parallel interface and two RS-232 ports (DCE and DTE)
- custom interface, i.e. 50 user I/O pins connected directly to the APEX device.

In order to support processor functions implemented in the APEX device, the board includes a memory system consisting of the following parts:

- volatile 64MB of SDRAM memory
- non-volatile 4MB of Flash memory and 256KB of EPROM memory
- pipelined 1MB of cache memory with burst SRAM.

The board also supports IEEE Std. 1149.1 Joint Test Action Group (JTAG) for system testing, as well as Extended JTAG (EJTAG) for development and debugging of MIPS-like microprocessor functions. For additional analysis, the JTAG port can be used with the SignalTap embedded logic analyser available with the Quartus[™] development software.

The APEX 20K device is programmed via the JTAG interface. It may be programmed directly using the Quartus or the MAX+PLUS[®] software using either the MasterBlaster[™] or ByteBlasterMV[™] dedicated cable.

Technical Assumptions for the Project

The main aim is that the Generalised Hough Transform (introduced in section 3) is implemented in the on-board APEX device. This means that the project refers mainly to the case of irregular pattern recognition for grey-level images taking also into consideration the scaling problem (see section 4).

The on-board APEX device interacts with a stationary PC using one of the available standard interfaces. In order to calculate the Hough Transform the SOPC board has to receive an input image and a pattern from the PC. In implementation it proved to be more convenient to generate the appropriate set of patterns from the origin pattern on the PC and then next send it to the board. Thus, the SOPC board receives the input image and the set of patterns generated by rotating and scaling the given pattern. The results (i.e. the calculated values of the accumulator array cells) are then sent to the PC. Some more detailed assumptions are as follows:

- the input image and the patterns are 256 grey levels
- the size of the patterns is set at 25x25 pixels

Biuletyn Instytutu Automatyki i Robotyki, 17/2002



- value L (see 25) is set at 72, therefore $\Delta \alpha = 5^{\circ}$
- the maximum size of the input image is set at 512x512 pixels.

Based on the above, the calculating process of the Hough Transform proceeds in the following way. At the beginning it is necessary to generate the appropriate set of patterns by rotating and scaling (if required) the origin pattern. The obtained set consists of L · n patterns (see 24 and 25). Next, in order to reduce calculation complexity and to decrease transmission from the board to the PC, the histogram analysis (see section 3) must be precomputed. This is the last crucial time-consuming operation for the PC. After that, the input image and the set of the patterns are sent to the SOPC board (i.e. to SDRAM on the board). Based on the results of histogram analysis the PC sends co-ordinates (x_T, y_T, α, s) one by one of the required cell of the accumulator array and in reply it receives value H(x_T, y_T, α, s) calculated according to equation 19. Finally, the PC chooses the global maximum in the accumulator array and displays the result.

The most important problem in the presented scenario is the implementation of the function $H(x_T, y_T, \alpha, s)$ in the APEX device. In practice it means that "the *sealing* method" must be implemented. Thus, the difference between the fragment (indicated by x_T, y_T) of the input image and the pattern (indicated by α, s) must be calculated by the hardware.

Implementation of function $H(x_T, y_T, \alpha, s)$

In this subsection the elaborated solution to the problem of function $H(x_T, y_T, \alpha, s)$ implementation in the APEX device is presented. Figure 7 shows the block diagram of the structure implemented in the APEX device.

As previously described the SOPC board receives the input image and the patterns from the PC at the beginning. The implemented structure has only one aim; to calculate the difference between the given fragment of the input image and the given pattern. The image fragment and the pattern are indicated by the arguments of function $H(x_T, y_T, \alpha, s)$. This means that the PC sends to the SOPC board co-ordinates (x_T, y_T, α, s) of the accumulator array and waits for the result, i.e. value of function $H(x_T, y_T, \alpha, s)$. This is a single cycle of the whole process of the accumulator array calculation which starts by sending signal RESET. Signal RESET clears the main elements of the implemented structure: COUNTER, BUFFER, SUBTRACTER and COMPARATOR. This is the initial step for the process and after that the device is ready to calculate accumulator array cells one by one. In order to describe precisely the basic cycle for the implemented structure the following algorithm will be useful.

Biuletyn Instytutu Automatyki i Robotyki, 17/2002



Figure 7. Block diagram of the structure implemented for function $H(x_T, y_T, \alpha, s)$

<u>ALGORITHM</u>

- <u>Step 0</u>: The APEX device receives (subsequent) co-ordinates (x_T, y_T, α, s) and indicates adequate fragment of the input image (by x_T, y_T) and adequate pattern (by α, s).
- <u>Step 1</u>: At the j-th step the BUFFER receives the j-th pixel (i.e. 8 bits) of the input image fragment and the SUBTRACTER receives the j-th pixel (i.e. 8 bits) of the pattern.
- <u>Step 2</u>: The COUNTER sends the SUBTRACTER signal that begins the calculation process of the difference between the pixels. The obtained difference is added to the previous one stored in the SUBTRACTER.
- <u>Step 3</u>: If there is any pixel of the pattern left, then j := j+1 and go to step 1. If there is no more pixels of the pattern, the COMPARATOR receives value $H(x_T, y_T, \alpha, s)$ from the SUBTRACTER and compares it with the previous one.
- <u>Step 4</u>: The COMPARATOR sends out value $H(x_T, y_T, \alpha, s)$ to the PC with information whether it is the temporary minimum or not. END

The COUNTER acts as the control element of the implemented device. It controls all elements by sending control signals and has a database containing information about the indexes of the pattern pixels, as shown in Figure 8.

Biuletyn Instytutu Automatyki i Robotyki, 17/2002



Figure 8. Indexed pixels of the pattern

The indexes of the pattern pixels allows the COUNTER to omit these elements (marked grey) of the pattern that are not important when the pattern is rotated. According to the assumption for the pattern size (25x25=625) there are 136 pixels in the pattern that should be omitted.

The technical version of the block diagram (shown in Figure 7) of the structure implemented in the APEX device is shown in Figure 9 with sizes and names of buses and control signals included.



Figure 9. Technical scheme of the implemented structure

The main function of the COUNTER is the generation of the control signals for the BUFFER and SUBTRACTOR. These signals synchronise every

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

single cycle with the use of signal CLK and are generated only for the adequate (marked white) indexes of the pattern pixels (see Figure 8). Additionally, at the end of the calculations for the given pattern, the COUNTER generates a control signal for the COMPARATOR.

The BUFFER task is to store the currently considered pixel of the input image in order to make it accessible for the SUBTRACTOR. The BUFFER is reset asynchronously by signal RST_B.

The SUBTRACTOR is the main element of the considered structure (from the point of calculation process) and is designed to calculate value Q = |A - B| + C. The SUBTRACTOR is reset asynchronously with signal RST_S generated by the COUNTER.

The COMPARATOR sends out (output Q) the calculated value $H(x_T, y_T, \alpha, s)$. Additionally the COMPARATOR tests whether the value $H(x_T, y_T, \alpha, s)$ is less than the previous one. If it is less, then line ALB is set on "1". The COUNTER resets the COMPARATOR synchronously with signal RST_K.

Example results

The final implementation turned out to be quite efficient with a successful series of experiments carried out. The most important fact is that the device accelerated the process of Hough Transform calculation several times (about 10). Example results are shown in Figure 10 and in Figure 11. Each figure includes an input image, an input pattern, the set of 72 created patterns and the fragment of the calculated accumulator array.



Figure 10. Example result with many similar elements in the input image

Biuletyn Instytutu Automatyki i Robotyki, 17/2002



Figure 11. Example result with only three elements in the input image that are similar to the pattern

7. Conclusion

In this paper the hardware implementation of the Hough technique for irregular patterns has been presented. It is very easy to notice that the presented solution is far from perfect. For example the whole process is accelerated by the hardware only in the case of function $H(x_T, y_T, \alpha, s)$ calculation, nevertheless this is the core of the Hough Transform. The increase of calculation speed results directly from "the *sealing* method" and is proportional to the assumed resolution of the pattern. It is possible to exploit this hardware solution very efficiently using more sophisticated software and to build a computer vision system working several times faster than normal.

It is worth mentioning that if the pattern is known in advance then the set of patterns may be pre-computed (e.g. in the guidance system of cruise missiles). Also the process of histogram analysis and the process of calculating function $H(x_T, y_T, \alpha, s)$ may be performed simultaneously. This results directly from the histogram analysis algorithm (see section 3). Since values $d_{(i,j)}$ are independent (see 23), thus function $H(x_T, y_T, \alpha, s)$ may be calculated independently for any fragment $Q_G^{I(i,j)}$ of input image I_G , immediately after obtaining a decision from the histogram analysis for the considered fragment $Q_G^{I(i,j)}$.

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

Although the elaborated hardware implementation of the Hough technique is a first version, it is very promising. There were many problems during realisation of the project, not mentioned in this paper (e.g. the problem of interconnects control logic implementation). Further work in this area is expected to bring more efficient implementations of the Hough technique. This may enable one to build a versatile real-time Hough Transform computer vision.

References

- Anagnou A., Blackledge J. M.: Research Report Pattern Recognition using the Hough Transform. Science and Engineering Research Centre, De Montfort University, Leicester 1993.
- [2] Ballard D. H., Brown C. M.: Computer Vision. Prentice-Hall, Englewood Cliffs, New York 1982.
- [3] Ballard D. H.: Generalizing the Hough Transform to Detect Arbitrary Shapes. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. Los Altos, CA. 1987, pp. 714-725.
- [4] Davies E. R.: *Machine Vision: Theory, Algorithms, Practicalities.* Academic press Ltd, 24/28 Oval Road, London NW1 7DX, United Kingdom 1990.
- [5] Deans S. R.: *Hough transform from the Radon transform*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 3, no. 2, 1981, 185-188.
- [6] Duda R. O., Hart P. E.: *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Comm. ACM., vol. 15, 1972, 11-15.
- [7] Fu K. S., Gonzalez R. C., Lee C. S. G.: ROBOTICS: Control, Sensing, Vision, and Intelligence. McGraw-Hill, New York 1987.
- [8] Hough P. V. C.: Method and means for recognizing complex patterns. U.S. Patent 3,069,654, Dec. 18, 1962.
- [9] Illingworth J., Kittler J.: *A survey of the Hough Transform*. Computer Vision, Graphics and Image Processing 44, 1988, pp. 87-116.
- [10] Jain A. K.: *Fundamentals of Digital Image Processing*. Prentice-Hall, New Jersey 1989.
- [11] Kiryati N., Eldar Y., Bruckstein A. M.: A probabilistic Hough transform. Pattern Recognition, vol. 24, no. 4, 1991, 303-316.
- [12] Leavers V. F.: *Shape Detection in Computer Vision Using the Hough Transform.* Springer, London 1992.
- [13] Li H., Lavin M. A., LeMaster R. J.: Fast Hough transform: a hierarchical approach. Computer Vision, Graphics, and Image Processing, vol. 36, 1986, 139-161.

Biuletyn Instytutu Automatyki i Robotyki, 17/2002

- [14] McLaughlin R. A., Alder M. D.: Technical Report The Hough Transform versus the UpWrite. Tech. Rep. 97/2, The University of Western Australia, Centre for Intelligent Information Processing Systems, Dept. of E.E. Eng., U.W.A., Stirling Hwy, Nedlands W.A. 6907, Australia 1997, Available from http://ciips.ee.uwa.edu.au/Papers/Technical_Reports/.
- [15] Radon J.: Uber die Bestimmung von Funktionen durch ihre Integralwerte langs gewisser Mannigfaltigkeiten. Berichte Sachsische Akademie der Wissenschaften Leipzig, Math Phys KI., 69, pp 262-267, 1917.
- [16] Żorski W., Foxon B., Blackledge J., Turner M.: Application of the Circle Hough Transform with a Clustering Technique to Segmentation of Digital Images. IMAGE PROCESSING II: Mathematical Methods, Algorithms & Applications, Horwood Publishing 2000, pp. 339-348.
- [17] Żorski W., Foxon B., Blackledge J., Turner M.: *Fingerprint and Iris Identification Method Based on the Hough Transform*. Bulletin of the Institute of Automation and Robotics MUT, 15, 2001, pp. 43-54.
- [18] Żorski W., Foxon B., Blackledge J., Turner M.: Irregular Colour Pattern Recognition Using the Hough Transform. Bulletin of the Institute of Automation and Robotics MUT, 15, 2001, pp. 27-42.
- [19] Żorski W., Foxon B., Blackledge J., Turner M.: Irregular Pattern Recognition Using the Hough Transform. Machine Graphics & Vision, 9, 2000, pp. 609-632.
- [20] Żorski W., Murawski K.: Pattern Recognition in the Case of Small Differences. Proceedings of the 7th IEEE International Conference MMAR 2001, Vol.2, pp.1055-1060.
- [21] Żorski W., Żak A., Turner M.: Hardware Implementation of the Hough Technique for Irregular Pattern Recognition. Proceedings of the 8th IEEE International Conference MMAR 2002, Vol.1, pp.561-566.

<u>OTHERS</u>

- [22] Altera Corporation: System-On-a-Programmable-Chip Development Board User Guide. July 2000, ver. 1.1
- [23] Żak A.: Design of a hardware implementation of the Hough Transform (written in Polish, Projekt sprzetowej realizacji transformaty Hougha). MSc thesis, Military University of Technology, Warsaw 2001.

Reviewed by Professor Włodzimierz Kwiatkowski Received November 2002

Biuletyn Instytutu Automatyki i Robotyki, 17/2002