When a Slow Device is Faster then a High Speed PC in the Intelligence Space

Gyula Max

Budapest University of Technology and Economics
H-1521 Budapest, P.O.Box. 91, Hungary, Tel: +36-1-463-2870, fax: +36-1-463-2871, e-mail: max@aut.bme.hu

Abstract: The paper compares an existing PC controlled camera to an FPGA controlled one in an Intelligent Space (iSpace). Devices use high-speed serial communication to flow information among them. Meanwhile the PC uses its resources parallel according to the theoretical model, the FPGA application which speed is 2% of the PC can reach the same result using the pipeline theory. The paper tries to find the boundaries of the recent applications to map the possibilities.

1 The Intelligent Space (iSpace)

Hashimoto Lab. in University of Tokyo has proposed 'Intelligent Space' (iSpace) since 1996 [1]. At the beginning, it consisted of two sets of vision cameras and computers with a homemade 3D tracking software. Later, a large-sized video projector (100 inches) was added to the Intelligent Space as an actuator. Actuators are mainly used to provide information and physical support to the inhabitants. These are done by speakers, screens, pointing devices, switches or robots and slave devices inside the space. Vision cameras and computers sets were arranged around an entire room and it changed into the Intelligent Space. The various devices of sensory intelligence cooperate with each other autonomously, and the whole space has high intelligence. Each intelligent agent in the Intelligent Space has sensory intelligence. The intelligent agent has to operate even if the outside environment changes, so it needs to switch its role autonomously. The agent knows its role and can support man. At present, each agent obtains the sensing information from multiple input attributes, such as cameras, microphones and sensor gloves. In addition, they obtain the augmented information from other agents. The agents can recompose the whole space as a virtual reality by using the augmented sensing information. In addition, the agents need to display intuitively and intelligibly the augmented sensing information and the support information to man. However, since an intelligible display differs for each individual, the method of display needs to be changed according to the person using it. In Intelligent Space, intelligent agents switch their roles autonomously. Intelligent Space

recomposes the whole space from each agent's sensing information, and returns intuitive and intelligible reactions to man. In this way, Intelligent Space is the space where man and agents can act mutually.

A space becomes intelligent, when Distributed Intelligent Network Devices (DINDs) are installed in it Fig. 1-1. DIND is a very fundamental element of Intelligent Space. It consists of three basic elements. The elements are sensors (cameras for computational elements (processors, computers, etc.) and communication devices (e.g. LAN connection). DIND uses these elements to achieve three main functions. First, the sensors monitor the dynamic environment that mainly contains people and robots. Second, the computational elements process the sensed data, extract information, and make decisions. Third, the DINDs communicate with other DINDs or robots through the network to share the sensed information.

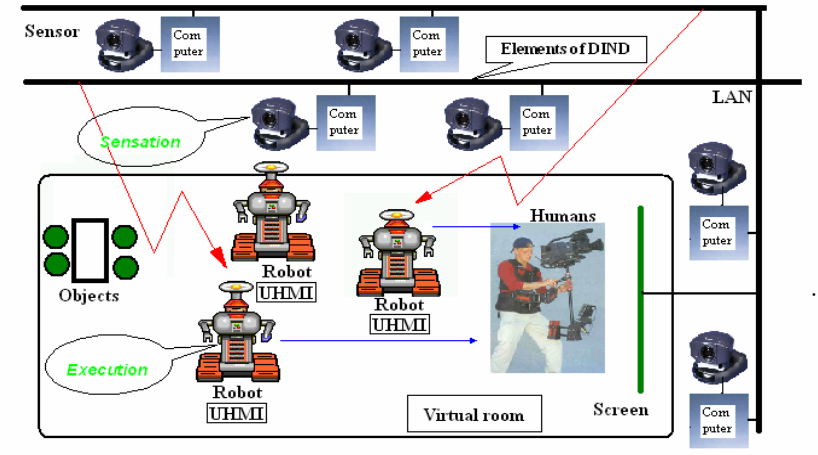

Figure 1-1
Basic Elements of Ubiquitous Sensory Intelligence

The ongoing research activities about Intelligent Space achieved several results and solutions in the field of feature extraction, human following, motion control, etc. These algorithms mainly use classical mathematical and soft-computing methods. Although these algorithms perform well in Intelligent Space, in some aspects it is easily possible to face their limits (e.g. in some cases they are not robust enough, or simply because of their architecture they cannot deal with some aspect of the problem).

Fig. 1-1 shows sensors communicating to robots and other DINDs Fundamental structure of DIND for new-generational, cognitive psychology inspired algorithms in computation and implement them in a DIND environment. In order to show why the cognitive psychology can give the needed boost consider the following example: The task is to find Mr. Smith is in the iSpace as well as to push a button

if Mr. Smith is there, but not if somebody else is there. Today this task is impossible to perform for a computer, yet a human can do it reliably in half a second or less. This result becomes more shocking if we know that the "processing time" of a typical neuron is about five ms. This seems to be fast, but a normal PC can do two-three hundred million operations in a second, and it means that the computer is one million times faster. The answer for how our "slow" brain can solve this up-to-now unsolvable task is its special architecture and particular information representation and processing. It is our belief, however, that in order to make major breakthroughs, many parts of architectures of the modern computer systems and the way of information representation need to be changed. Thus, new DIND concept is based on this phenomenon. The new architecture is built up by numerous, simple computational elements that can perform only primitive functions like addition, subtraction, but they do it quickly. These computational elements are connected to each other like the neurons in the brain. This architecture can be much more efficient in certain tasks than the complex, classical algorithms as in spite of the fact that thousands of simple operations are done, due to its special architecture, they can be performed in a fully parallel manner that tremendously reduce the calculation time.

In Intelligent Space, the models based on cognitive psychology and biology can work together and become an integrated cognitive system. This paper introduces one and half typical solutions on image recognition that are developed on the analogy of the human vision processing. Each solution shows how can be solved the image recognition imitated the biological structures and why and how we avoid our real task performing the same operations on the image. The rest of the paper is organized as follows. Section 2 introduces functions of the human visual pathway, how the brain processes an image. Section 3 describes how we reduced the size of the calculations. In the last section, we try to give a method how to solve our problems keeping our original task.

2 The Human Eyes

Devices of the iSpace need information on humans living together in this space. Devices must recognize them. The recognition is very important part of our system. Either the humans or the devices in the iSpace must know where the other is. A camera would have to found the human, to save his parameters and to try to follow him in the space.

In the human eyes and brain, millions of photoreceptors and neurons follow the target to recognize it [2]. We would like if our model was also based on a neural model that followed the biological aspects of visual information processing.

Visual processing begins in the retina Fig. 3-1. The photoreceptors that include 120 million rods and 6–7 million cones are located in the back of the retina, and are responsible for phototransduction. The rods are sensitive to light intensity, while cones are also sensitive to three different colors. These photoreceptors modulate the activity of the bipolar cells, which in turn connect with more than one million ganglion cells in each eye. The axons of the ganglion cells leave the eye at the optic disc and form the optic nerve, which carries information from the retina to the brain.

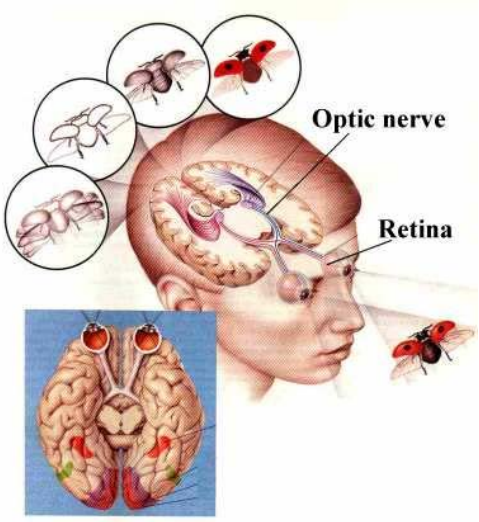

Figure 3-1
Human vision

The bipolar cells and the ganglion cells are organized in such a way that each cell responds to light falling on a small circular patch of the retina, which defines the cell's receptive field. Both bipolar cells and ganglion cells have two basic types of receptive fields: on-center/off-surround and off-center/on-surround. The center and its surround are always antagonistic and tend to cancel each other's activity. On the other hand, the on/off or off/on arrangement of the receptive field makes ganglion cells more responsive to differences in the level of illumination between the center and surround of its receptive field. The primary visual cortex populates approximately 2 billion neurons in a two-dimensional sheet about 2-3mm thick and topographically maps the visual field, with neighboring neurons responding to neighboring parts of the visual field.

3 Matching Pixels

If the mechanism described in Section 2 were followed, millions of memory would be needed. Somehow, the memory must be reduced. Our solution based on neural model and we want to keep this model. The goal of the proposed system is to adjust the camera on the center of the image planes [3,4]. The images provided by the camera will be referred to as master and slave images. The system adjusts the optical center of the slave camera so that the same point P is projected on both of the optical centers, while the master camera is not moving at all. A window of 9x9 pixels is considered as a pixel surrounding, which corresponds to a set of receptive fields getting inputs from an area of 81 pixels. On the master image, the

position of the 9x9 window is fixed to the optical center. The final task is to find a window on the slave image that best matches the window of the master image according to all the image features taken into consideration. Not all the pixels on the slave image are taken into consideration. In stereo vision if the external and internal parameters of the cameras (such as their position, orientation, and focal length) are known, then for each point on one of the images a line can be defined on the other image which will contain the pair of the point seen on the first image. This line is referred to as the epipolar line, providing a constraint to the pixels to be taken into consideration during the matching process.

This constraint is also present in the biological system. Most of the animals cannot move their eyes up and down independently. For this reason, it is also supposed that in the proposed model, there is no vertical rotation or translation between the master and the slave cameras; furthermore, the window of sharp vision on the master image is always in the optical center. This yields that the epipolar line on the slave camera's image plane is always horizontal and passes through the optical center of the slave camera. As a result, all possible windows of 9x9 pixels along the epipolar line of the slave image will be compared to the single window in the center of the master image. This implies that the image parts used as inputs to the model are the 9x9 square on the optical center of the master image and an Xmax times nine stripe on the slave image, where Xmax is the width of the slave image in pixels. In order to help comparison between master and slave images two auxiliary image matrixes are needed. An example is shown on Fig. 3-1 where the same picture can be seen. The intensity (top), edge magnitude (middle) and edge orientation (bottom) matrixes of the same image are shown in Fig. 3-1.

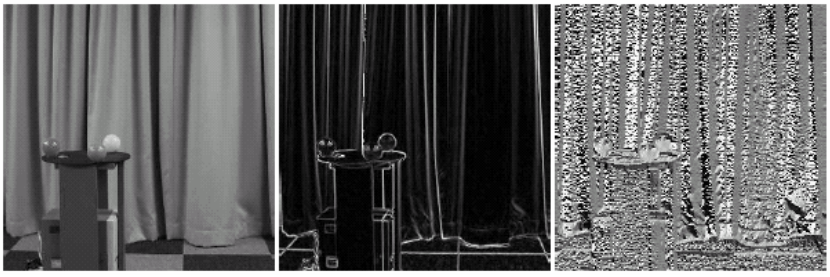

Figure 3-1
The intensity (left), edge magnitude (middle) and edge orientation (right) maps of an image

The input pixels of the model came from the master and the slave images. In our neural model inputs come from a camera. Since the binocular cells are always orientation selective, the input of the model is not only the input image from the cameras, but also the orientation map of the image, which includes the edge orientations and magnitudes in each pixel. To avoid using two cameras the first image is the master image and our task is to follow the human represented by the master image in the iSpace.

4 An Existing Model

An existing model is described in [5]. The model can be seen in Fig. 4-1.

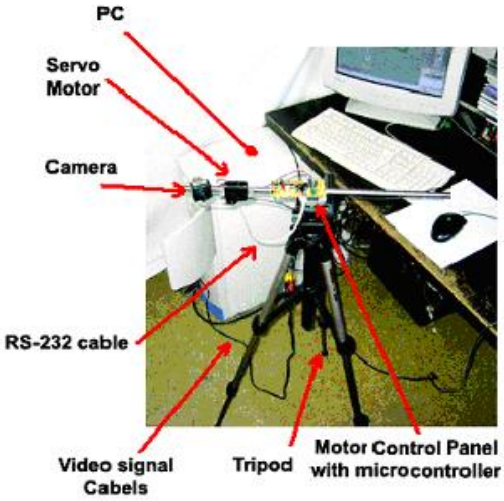

Figure 4-1
Implementation of the existing model

The model can be described as a feedback process. The camera makes images. After having the master image the camera tries to find the most comparable image in its neighborhood using an Xmax = 320 times 9 pixel slave image. The result, the new camera position is transferred through the serial port of the PC that is connected to the controller of the camera (Fig. 4-2).

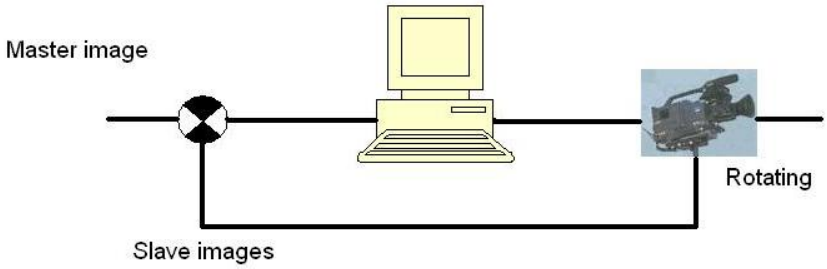

Figure 4-2
The feedback process

Since our process is planed to realize as a neural net, some layers are defined to separate its tasks.

Layer I (Input of PC)

1 The camera makes an image.

2 This image is sent to the PC.

Using 8 bits black and white pixels, one slave stripe image contains 2880 bytes. During one second, 25 images are made. As we can see later, only two-four images are needed in every second. This transfer is made by the communication between the camera and the FireWire of the PC.

Layer II (Edge detection in PC)

3 After having arrived all information of the original image stripe to the PC, the intensity (I), the edge magnitude (D) and the edge orientation (O) matrixes can be calculated.

Layer III (Matching pixels)

4 Using figures of I, D and O matrixes the master image is compared to each 9x9 image of the slave stripe. The comparison level (t) is an edge magnitude (D) value. This system input figure represents a value that shows from which value we want to distinguish the edges.

The process of the calculation is the following:

To calculate the different matrix (R) of master and slave matrixes:

The edge orientation value of the same pixel of the master and a 9x9 matrix of the slave image are compared to the comparison level.

- If one of them is less than t and the other is higher than t, then for each pixel

$$HA \quad (D_{master(i,j)} < t)\ AND (D_{slave(i,j)} > t)\ XOR (D_{master(i,j)} > t)\ AND (D_{slave(i,j)} < t)$$

$$AKKOR \quad \Rightarrow \quad R_{i,j} = \left| D_{master(i,j)} - D_{slave(i,j)} \right|$$

If both pixels are less than t, which means no sharp edges, the R is the absolute value of the intensity figures.

$$HA \quad (D_{master(i,j)} < t)\ AND (D_{slave(i,j)} < t) \quad AKKOR \Rightarrow \quad R_{i,j} = \left| I_{master(i,j)} - I_{slave(i,j)} \right|$$

- If both pixels are greater than t, which means too sharp edges, the R is the absolute value of the orientation figures.

$$HA \quad (D_{master(i,j)} > t)\ AND (D_{slave(i,j)} > t) \quad AKKOR \Rightarrow \quad R_{i,j} = \left| O_{master(i,j)} - O_{slave(i,j)} \right|$$

The result of comparition of these two 9x9 matrixes is the average of Ri,j.

$$R = \frac{\sum Rij}{81}$$

Layer IV (Decision making)

5 When all existing 9x9 slave matrixes are compared to the master matrix, 312 elements of array is calculated. Let us find which result is the minimal value of these 312 elements. This value shows the least different between the master and the slave image in this array.

Layer V (Positioning)

6 We must find the position of the least value and calculate the new position value of the camera.

In the following example Fig. 4-3, the ninth slave matrix of 9x9 is the same than the master image is. Thus, the ninth element of the 312 elements of array is zero. In this stripe, the minimum is at the position of nine. The camera must be turn to this position.

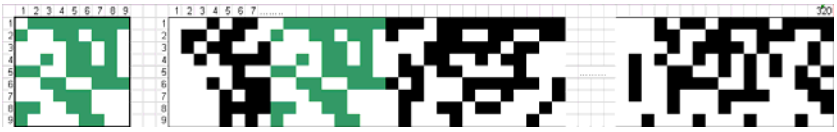

Figure 4-3
Example of a master and a slave image

7 Using the slow serial cable (9600 Baud) connection between the PC and the control panel of the camera the new position must be transferred.

8 Because of the mechanical part of the motor of the camera maximum four new values can be transferred to the camera in a second.

Using this structure the total speed of the process is 16 new positions / sec. The process saved its input images and I, D, O matrixes as well as outputs for testing the serial model.

5 Serial Model Using FPGA

The theoretical and the existing model were described in Section 3 and 4. We want to change the PC into FPGA because of its size, speed and price. The FPGA contains the same system as it was installed before on the PC (Fig. 5-1). The FPGA which was chosen is an xc2s200 (2s200pq208-6) device from the Spartan2 family. It has two hundred thousand programmable gates. Its working frequency is 50 MHz, that is 50 times slower then our PC is.

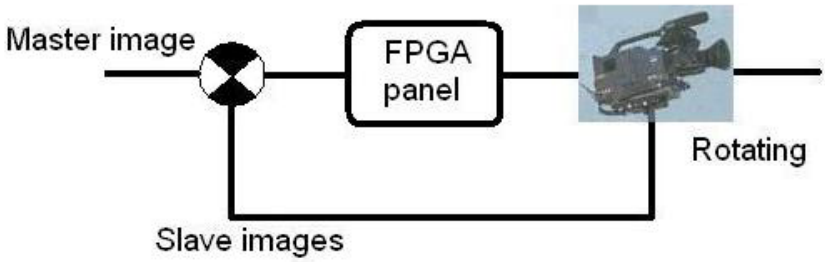

Figure 5-1
The implemented system with FPGA

When the implementation had been started, the problems were coming. One of them was the lack of the memory. It was realized that our FPGA had no enough memories. We had to find other solution. Our process was divided into two parts. Suppose that our PC has made the data processing. Our task is to make decision. In order to make decision information are needed. In our experimental environment, the primary and the calculated information are coming from the PC (Fig. 5-2). The master and the recent pixels are coming through a serial port by speed of 115,2 kBaud. This speed is approximately one eighth of the speed of normal USB 1. Since the camera cannot move with a high speed, a normal, low speed (9,6 kBaud) serial line is used to position the camera. As it is written in the section 4, after the three master matrixes, the recent pixels are transmitted through the channel. The goal of our decision-making is to turn the camera left or right three or four times in every second.

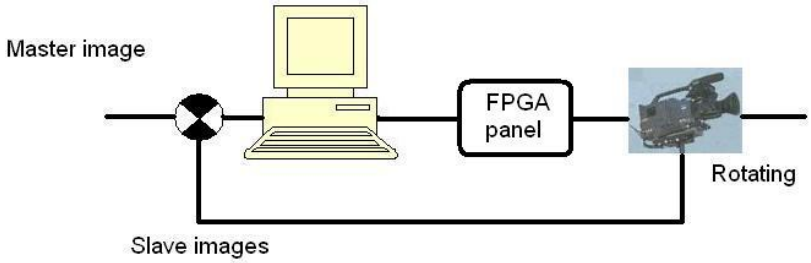

Figure 5-2
Split model with FPGA

We followed the way that was written below. We have already wanted to implement only the device of the decision-making in the FPGA panel. Considering to Section 4, the model was cut into two parts between Layer II and LAYER III. It must be done because in our FPGA there were not enough memories to implement all features of the existing model as this is calculated below. For this reason, this solution is called half solution in Section 1 compared to the existing model (Fig. 5-3).

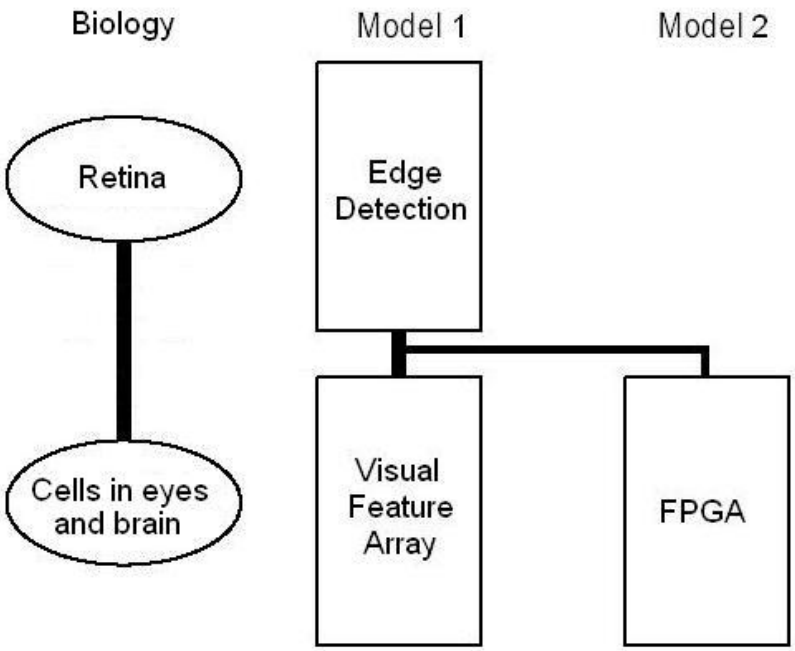

Figure 5-3

Structure of the biological and the experimental models

In the half model, which was implemented a serial model was made because of the lack of memory. The circuit downloaded to the FPGA contains five blocks with three inputs and two outputs in Fig. 5-4.

Figure 5-4

Circuit downloaded to the FPGA

The inputs are the clock and reset signals as well as the incoming pixel information of I, D and O. Since all three matrixes are transferred from the PC to the FPGA instead of the I matrix only, the speed of the serial model is one third of the normal one.

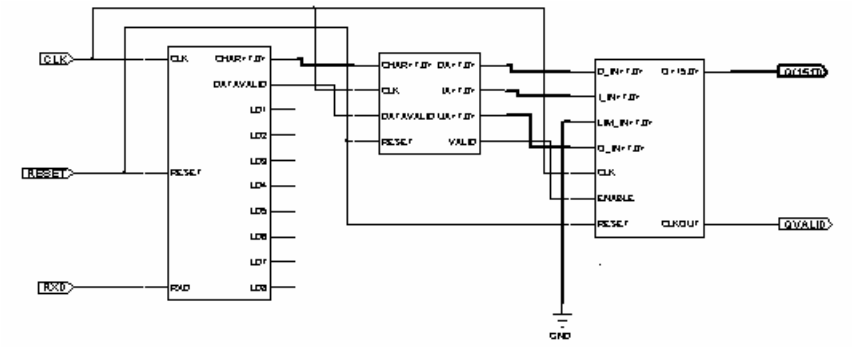

Figure 5-5
Inside structure of the serial model

Receiver: The incoming signals arrive to the receiver block of FPGA. The task of this block is to enter signals to our system. After having validation every incoming I, D or O pixels send to the next block.

Calculator: The first 81 three bytes incoming information are the master pixels. These 243 bytes build three matrixes for master I,D and O pixels. After the master matrixes the pixels of the actual picture are coming. Using the calculation method described in section II, result of each 81 pixels is figured. It takes a vector of 312 elements. This vector of 312 elements is transported to the decision maker block.

Decision maker: This block is responsible for the positioning camera. Using the elements of the incoming vector this block calculates the new position of the camera selecting the space of minimal value in the vector of 312 elements. The value of the coordinate of the new position is transmitted to the serial port with speed of 9600 Baud. This channel also uses an RTS flag for signaling data transfer.

6 Working Method of the Serial Model

Our original purpose was to implement the neural model described in Section 4. Because of lack of memory, the model was reduced and the neural model was almost forgotten. What is remained?

During the implementation some weakness were found. One of the weakness is that no multidimensional matrixes are in Verilog, in our hardware description language. The upper range of a multidimensional matrix is two. Two-dimensional matrixes can be declared, but a three-dimensional one cannot. Since a two dimensional 8-bit matrix is a three dimensional matrix in Verilog, this matrix cannot be declared for our FPGA. To solve this problem a simply linearization method was used for transforming the master matrixes.

Real problems are awake after getting the master matrixes. One picture coming trough the incoming channel takes 9 times 320 pixels, namely 23040 bits. If I, D and O pixels of a picture are stored which were about 70kbits, no free space remains for the further calculations in the FPGA. This result explains why the serial data processing was chosen. Every incoming three bytes value gives a piece of result ($R_{i,j}$) in our result calculation described in section 4. A special shifting calculation method was stood up to be able to avoid the standard, neural based multidimensional calculation method.

After received master matrixes of the recent pictures the serial calculation method is started. The incoming pixels are compared to the adequate master pixels. In order to do this with the just incoming pixels we do not have to know any other pixels. We have to know the master pixels and the just incoming pixels (MD, MI, MO, I,D and O). Each appearance of the just incoming pixel is calculated. It gives nine results, which are stored in nine different variables. After each incoming image (9x9 pixels), the result is sent to the decision maker block. After 2880 incoming pixels, one turn of the calculation is finished and a new position of the camera is fixed.

7 Differences between Neural and Serial Model

The bottleneck of our neural process is the “huge” memory needs. In order to follow the neural method all incoming information of a picture must be saved. It can be done in the PC, but it cannot in the FPGA. To do this the minimal memory need is 320x9+81 bytes, namely 23688 bits. Some inbuilt memory of the FPGA can be used, but reading and writing processes take too much time, if we want to use memories organized by bytes and not bits. Another problem in case of memory organized by bytes is the massive additional hardware using bytes instead of bits. To put these memories and not only the half process, but the total one into the FPGA, at least a device with 1 million gates can be used. Unfortunately, our device with 200 thousands gates is too small to solve the total process.

This “huge” amount of memory is not a problem in a normal PCs. In the PC, the tasks are done one by one. In the first step as it is written in Section 4 at Layer I., the computer collects incoming information. The next step, using all collected figures, calculates the next position of the camera (Layer II, Layer III, Layer IV). In the last step (Layer V), the camera is positioned.

The serial method uses very different ways. In the first step collects the necessary figures only. In this case, these are the master pixels. Reading parallel the new incoming pixels, the calculation part of our process calculates particularities of the result vector. For this reason this calculation method is very fast, because after having read the last recent pixel we need only a few clock signals to produce the new position of the camera.

During the test using 115,2 kBaud serial cable, only 1,3 new position were produced in a second. Do not forget that three times more information were transferred trough the channel than it was necessary. The transfer speed also can be increased. Since only 3-4 new positions are needed in a second, this speed is enough.

Conclusion

In Section 6, a question was asked: What is remained? The answer is quit simply. A new method is remained.

- The serial model can be implemented in an FPGA.
- The pixel matching method is similar to the neural model, rather than the same processing in the PC.
- The process works almost at the same speed than it worked in PC, though the speed of the PC is fifty times more than the speed of the FPGA.

Only one question was remained. Why are PCs used? They are used because many effective implementations are ready for PCs. Their surfaces are intelligent and quick enough. In many cases, these PCs are not necessary elements of the processes. Several times many, more cheaper solutions are suitable. This paper showed an example when PCs can be changed into FPGA. The effective rate of the process speed is remained the same. The rate of cost-benefit is much higher then in the former solution when PCs was used. A PC takes about EUR 300 and the FPGA used in this solution is less then EUR 4. In our process, the slowest part was the motor of the camera. Camera can be moved 3-4 times in a second. Using this parameter the transfer rate in our part of the process is a bit higher then 100 kBaud.

Why do we use a more expensive solution instead of a cheaper one?

In the closed future, we hope that the full model is implemented in a more effective FPGA. As our figures showed, approximately 1 million gates are enough for the total implementation.

References

[1] Péter Korondi, Hideki Hashimoto, ”INTELLIGENT SPACE, AS AN INTEGRATED INTELLIGENT SYSTEM”, **Keynote paper** of International Conference on Electrical Drives and Power Electronics, Proceedings, pp. 24-31, 2003

[2] D. Hubel. Eye, Brain and Vision. W. H. Freeman & Company, 1995

[3] Barna Reskó, Péter Szemes, Péter Korondi, P. Baranyi, Hideki Hashimoto: “Artificial Neural Network based Object Tracking in Intelligent Space”, SICE Conference, Sapporo, Japan, 2004, pp. 1398-1403

[4] Gyula Max, Péter Szemes: Limits of a Distributed Intelligent Networked Device in the Intelligence Space, Proceedings of the 5th International

Symposium of Hungarian Researchers on Computational Intelligence, pp. 173-182, Budapest, 2004

[5] Barna Reskó, Péter Baranyi, Péter Korondi, Péter Szemes, and Hideki Hashimoto „Artificial Neural Network based Adaptive Object Tracking in Intelligent Space" *Power Electronics and Motion Control Conference* Riga, Latvia, CD ROM 2004