

# A Neural Control System of a Two Joints Robot for Visual Conferencing

DAYA BASSAM and ISMAIL ANIS

*Department of Computer Network and Telecommunications Engineering, University Institute of Technology – Lebanese University, IUT de saida, B.P. 813, Saida 961, Lebanon.  
e-mail: B.dayaa@ul.edu.lb*

**Abstract.** This paper presents a real time interactive conferencing system that allows people to communicate via animated images. The employed distant cameras act as a remote vision system. The cameras simulate the movement of an observer and return images covering the surrounding view of the pictured person. This paper also describes the neuro-command system that controls a robot with two joints (degrees of freedom) that carries the used camera. This neural system translates the head movements. A methodology from image processing has been adopted for the classification, and facilitation of the phase of neural network learning process.

**Key words.** artificial neural network, conferencing, image processing, pattern recognition, robotics

## 1. Introduction

Picture analysis through image processing requires substantial development on all levels of operation. Namely, the initial state of the picture taken, its compression, recognition and also its interpretation. The data-retention time (delay) of these phases influences incontestably the quality of processing tasks and the time of execution. However, the quality of service and the processing time are often the major concern of real-time applications. A successful computer vision system should account for granting these requirements. The analysis and interpretation of pictures is subject of important research and scientific studies [7–9, 11, 12].

Researches in this domain bypass the simple setting of compression to the classification, since it is about extracting high-level structured information. It allows us to recognize and to index pictures. The complexity of processing required by these applications motivated several researchers to focus on the particular problem resolution [10, 13, 14]. Although the literature in image processing is abundant by classic methods of segmentation [1, 6, 12, 15], no general approaches are known to exist.

The neural technique already used to command robots [2–5] is used for head orientation recognition: this includes the *Face, Down, High, Right* and *Left*. After the acquisition of the observer's picture, a processing technique has been developed to extract the minimum necessary information to make neural network learn. The learned network will be used to transmit the command to a remote camera.

The intelligent system capable of recognizing some pictures is used to command a robot with two joints to carry another distant camera. As all the scenes that take place in front of us, where the received picture changes according to the movement of the head, a distant camera will be commanded (controlled) according to the movement of the head of the client at the other side. This work can be integrated into the development of the visual conference technology. Several applications have already been developed. "Videovu" (Video and Audio over the Internet and over LAN. Integrate tools), "ShowMe" (Video and Audio over the Internet, "Integrate wb" (WhiteBoard)), "Picture Window" (Video and Audio over the Internet) are examples of such applications.

Our application, which enables remotely controlling to a camera held by a robot with two joints, may lead to more interactive instruments. In fact, our system allows people from different sites to monitor themselves remotely, in real time, at the same time communicating parties will be exchanging animated pictures under a supervising controller. The distant camera works as human eyes and the picture transmitted by the network changes according to the orientation of the observer's head, such a property makes, visual conferencing a more interactive and dynamic process.

This paper treats two problematic aspects: the pre-processing of pictures acquired and the recognition of the orientation of the head for the purpose of improving the quality of classification, facilitating the neural training, and minimizing the turnaround time of the neural system.

Our final objective is to present our intelligent system in a visual conference to improve the quality of the interaction among different communicating sites. To reach this objective, we present a robot with two degrees of freedom allowing to direct the camera in four directions. This robot with two joints has been assembled using two stepper-motors with an electronic controller (Driver).

We will describe the implementation and functionality of the robot in Sections 2 and 3. We will then propose the configuration and the preparation of the neural network. The different aspects of computer programming will be presented in Section 4. Final comments and the general conclusion are provided in the last section.

## **2. Robot Construction**

The implementation of the robot with two joints (Figure 1) is done using two step-motors. Every motor is equipped with a gear system to drive the two joints. At the tip of these two joints, a wrist carries the camera and directs it in the four directions. A system of electronic command Driver (Figure 2) constitutes the interface between the parallel port of the PC and the motors of the robot. Once the support of the robot is manufactured, we construct the command circuit of the two motors. This interface joins the information transmitted through the parallel port and transforms it to signals that drive the motors of the robot.

Combining the robot and its command circuit brings us to the system as illustrated in Figure 3.

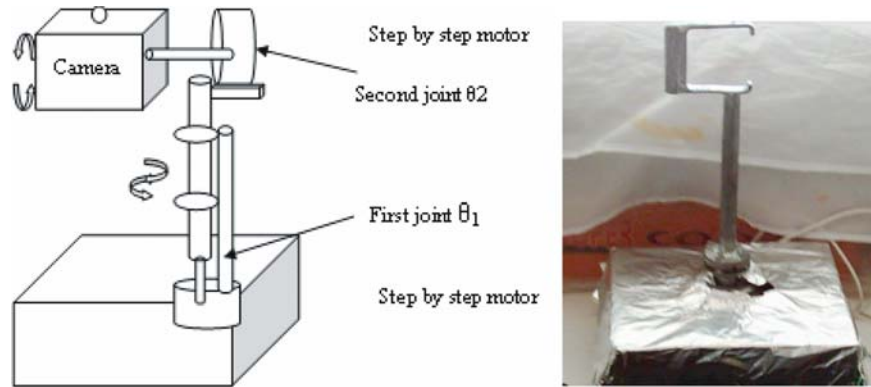


Figure 1. Realized robot with two joints holding the camera in its handfull.

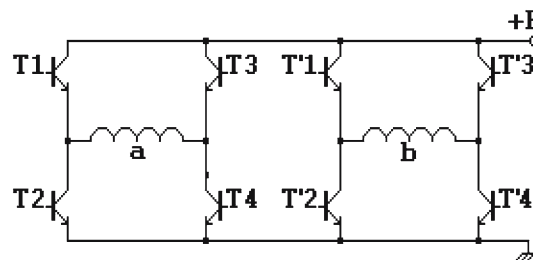


Figure 2. Electronic command circuit ordering the two step-motors of the robot. It is the interface between the PC and the robot carrying the camera.



Figure 3. Real diagram of the robot with two joints with its circuit of command.

### 3. System Function for the Robot Guidance

The relative data of the picture coming from the camera of the observer's (server side) must undergo an intelligent processing phase. This phase is used to deduce the decision made by the neural system (or the command) to transmit (via the computer network joining the two sites). It will be translated into a numeric command, then transmitted toward the robot with two distant joints at "client side". The general structure of our system is depicted in Figure 4.

This robot follows the movement of our head which we can, through this robot, visualize in real time a remote scene. Also we can control the vision field with the orientation of our head which, in turn, determines the orientation of the remote camera (installed on the robot) in an appropriate way. Note that a user can orient his/her head in 4 directions.

The computer captures the image with its camera and analyzes the situation of the head. It formulates the command which is then transmitted to the second camera located on the remote robot.

Accordingly, we state five different cases (Figure 5):

- (1) When the user looks directly to the PC, the user visualizes the distant scene transmitted by the network and taken by the other camera carried by the robot.
- (2) When the user turns his head to the left, the robot will directly turn the camera to the left and the user will visualize a new distant scene to the left of the previous scene.
- (3) Same movement as in (2) with an orientation to the right instead.

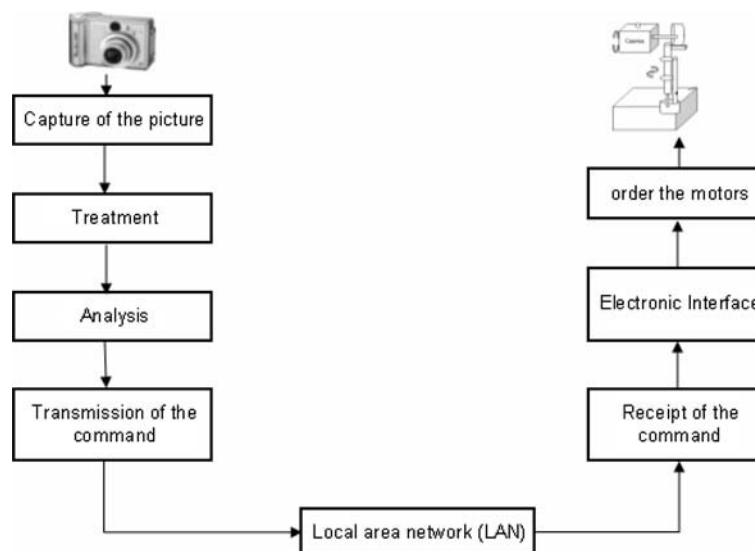


Figure 4. General structure of command of our intelligent robotic system.

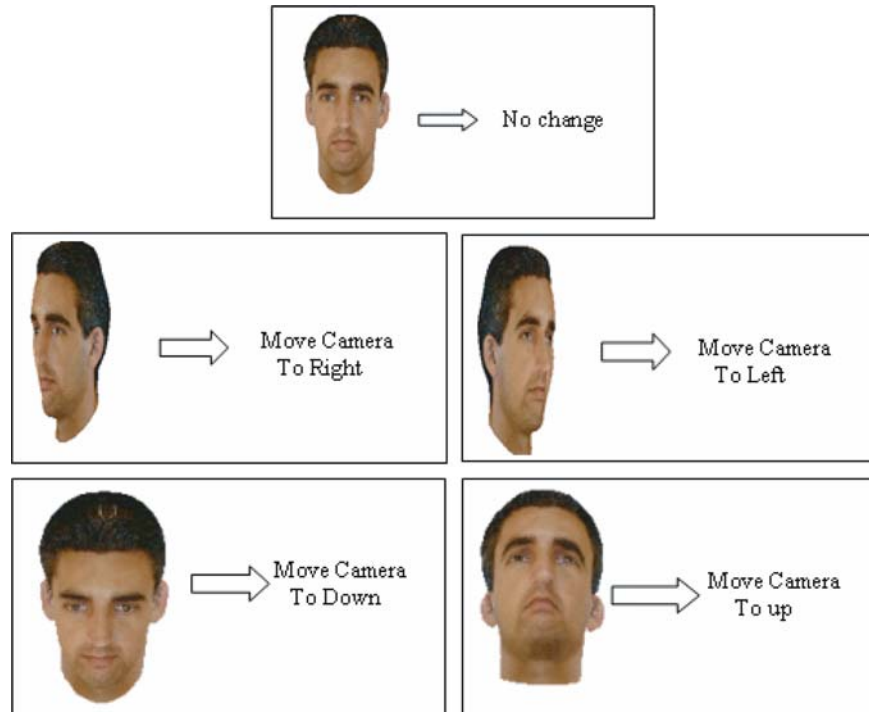


Figure 5. Different cases of command of the distant robot according to the orientation of the human head.

- (4) Same movement as in (2) with an orientation downwards (down) instead.
- (5) Same movement as in (2) with an upwards (up) orientation instead.

The following section will present the programming aspects and components of our system.

#### 4. Programming Implementation (data processing)

This section is of twofold: the first part involves the configuration and preparation of the neural network undergoing a phase of training; and the second part executes concurrently achieving the following tasks:

- (1) Reading of connection weights (results of the first part).
- (2) Capturing of picture from the digital camera.
- (3) Reduction of the size of the captured picture.
- (4) Recognition of the captured picture to deduct its decision.
- (5) Transmission of the decision through the computer network to order the remote robot.
- (6) Command of the stepper-motors of the robot.

#### 4.1. PROGRAMMING OF NEURAL NETWORK

The neural network was implemented using the “MATLAB” software package. Learning is the phase of development of the neural network which is modified by the behavior of the network. Throughout the greater part of process described by the algorithm (Section 4), the modified variables during the training are the weights of connections. Learning is the modification of weights of the network in response to respond to the examples and experiences obtained. It is often impossible to decide *a priori* the values of weights of connections of a network for a particular set of application data. After the learning process, weights are fixed: so it is also in the phase of utilization.

The neural network used includes 3 layers: an input layer, a middle layer, and an output layer. It is a multilayer network having in the first layer, the input layer, equivalent to 1291 neurons representing a picture, i.e. a matrix of 1291 pixels. Every pixel is represented by a neuron. The second layer, the middle layer, contains 100 neurons. Increasing the number of neurons in the middle layers, it increases the efficiency of the network in phases of learning and recognition. The last layer is the output layer; the number of neurons in this layer and in the input layer depends on the processed application. For example, our first test recognizes five pictures (orientation face Up, Down, Right, Left, or Face). It is sufficient to have 5 neurons or 5 bits for the output. We only have 5 prototypes to learn. Actually, these 5 neurons of output permit to code (in binary) not only 5 prototypes, but 32 prototypes (i.e.  $2^{**}5$ ). Yet it is preferable to have 5 orthogonal codes as a basis of training.

For example, for the 5 bits of output of the network we will choose:

00001 ' for orientation “Up”  
 00010 ' for orientation “Down”  
 00100 ' for the right orientation  
 01000 ' for the left orientation  
 10000 ' for an other picture or face

Before proceeding to the phase of training (the one prepared under Visual Basic), a database containing 50 pictures representing several shapes of a face of high orientation and 50 pictures representing shapes of low orientation, and similarly for the remaining cases. These pictures undergo two operations: 1 – reduction of the size (to limit the size of our neural network), 2 – quantification of values of every pixel to shades. This data constitutes the basis of training of our multilayer network.

##### 4.1.1. Learning Phase

The neural network chosen must learn to distinguish the customer’s face in its five positions (high, low, right, left, face). For a good generalization, the training set must include several samples for every position.

There is no fixed number of samples. Consequently, there is no fixed number of neurons of the network. The different tests bring us to a multilayer network composed of 100 neurons for the middle layer which includes 5 neurons in the output layer, corresponding to the number of cases to recognize (face, Up, Down, right and left).

The learning phase is done once under “MATLAB”, which possesses libraries of neural networks. Synaptic weights of the network are taken randomly at the beginning.

Learning, using the method of “Levenberg-Marquardt”, assures a convergence toward the desired value with error of approximately  $10^{-6}$ . The training set includes the following stages:

1. Capture of images from the digital camera. We prepare the basis of training for the different cases of the face.
2. Reduction of the image size to  $50 \times 50$  pixels. This corresponds to the number of neurons of the input layer of the network.
3. Transformation of the picture to grayscale to take the necessary information, facilitating the training without losing the main features of the detected picture.
4. Conversion of every pixel in the picture is to type long that is 32 bits in Visual Basic.
5. Normalization of values is done to obtain values between 0 and 1 because the transfer function of neural network provides the output values between 0 and 1.
6. Save the synaptic weights values obtained after the learning process.

At the end of this phase, we get the values of the synaptic weights for every layer, where each line corresponds to the values of one neuron.

Once the decision is taken, for the captured pictures in line, by the neural network, the command should be sent from the client to the server to position the distant robot.

#### 4.2. SERVER SIDE PROGRAMMING

In this section, we take care of the server that is connected to the robot with two joints carrying the camera. This stage includes the following tasks:

- Initialization of the parallel port (Figure 6).
- Initialization of the socket so that we can receive the numeric data of “client” (Figure 7). This data represents the five different commands that we must transmit to the robot carrying the camera at the “server side”.

We must note that the address of the parallel port is displayed on “text box” and it differs from one Windows version to the other. We must avoid modifying the two joints by hand which might affect the circuit’s electrometric strength.

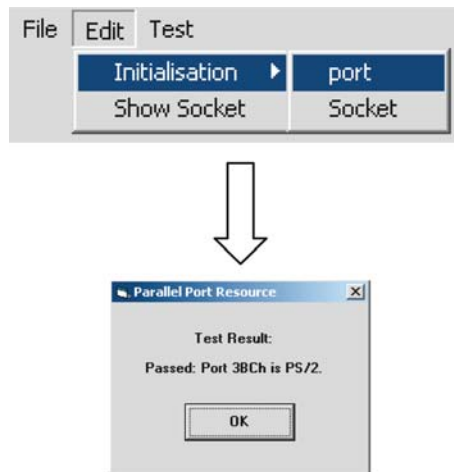


Figure 6. Initialization of the parallel port.

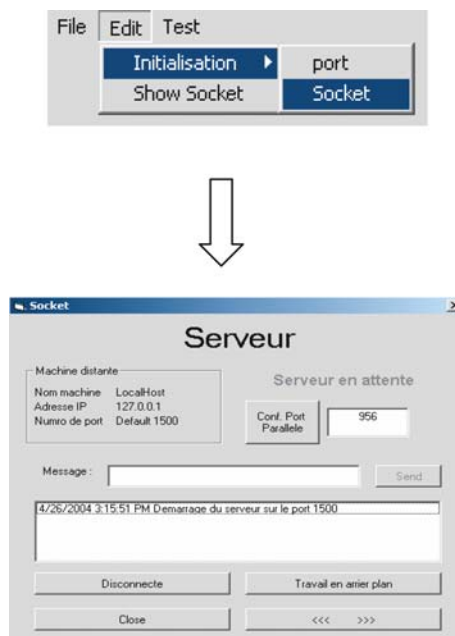


Figure 7. Initialization of the socket of the parallel port.

#### 4.3. CLIENT SIDE PROGRAMMING

Once the learning phase of the neural network is finished (Section 4.1.1.), the phase of recognition must be implemented on the “client side”. It permits



the recognition of the orientation of the present face to transmit (client  $\rightarrow$  server) the numeric command via the computer network to order the robot at the “server side”. In addition, the server redirects the robot’s camera to transmit in the opposite direction (Server  $\rightarrow$  Customer) the new scene according to the orientation of the client’s head.

The face recognition phase permits taking the following decision (or the appropriate command word to transmit). From the client side, a graphics interface permits the visualization of the distant scene with the supplementary information: The five outputs of the neural network and the face at the client side.

In this mode, there are several stages to follow:



*Figure 8.* Outputs of neural network according to the orientation of the head. We notice that when the face is in the position “face”,  $s_{(0)}$  is near to 1 whereas  $s_{(1)}$ ,  $s_{(2)}$ ,  $s_{(3)}$  and  $s_{(4)}$  are near to 0. When the face turns to the right, it is  $s_{(1)}$  that will be near to 1 and all other outputs will be near to 0.

- **START:** To begin capturing the picture from the digital camera “client side”
- **Input  $L_w$  and  $I_w$ :** For the acquisition of synaptic values of neural network after the learning phase.
- **Recognition:** After that we take the picture and directly compress it, we feed it into the function of neural network simulator to take the decision. At the end

Position of the camera  
Before moving the  
head to the right.



Moving the head to left side



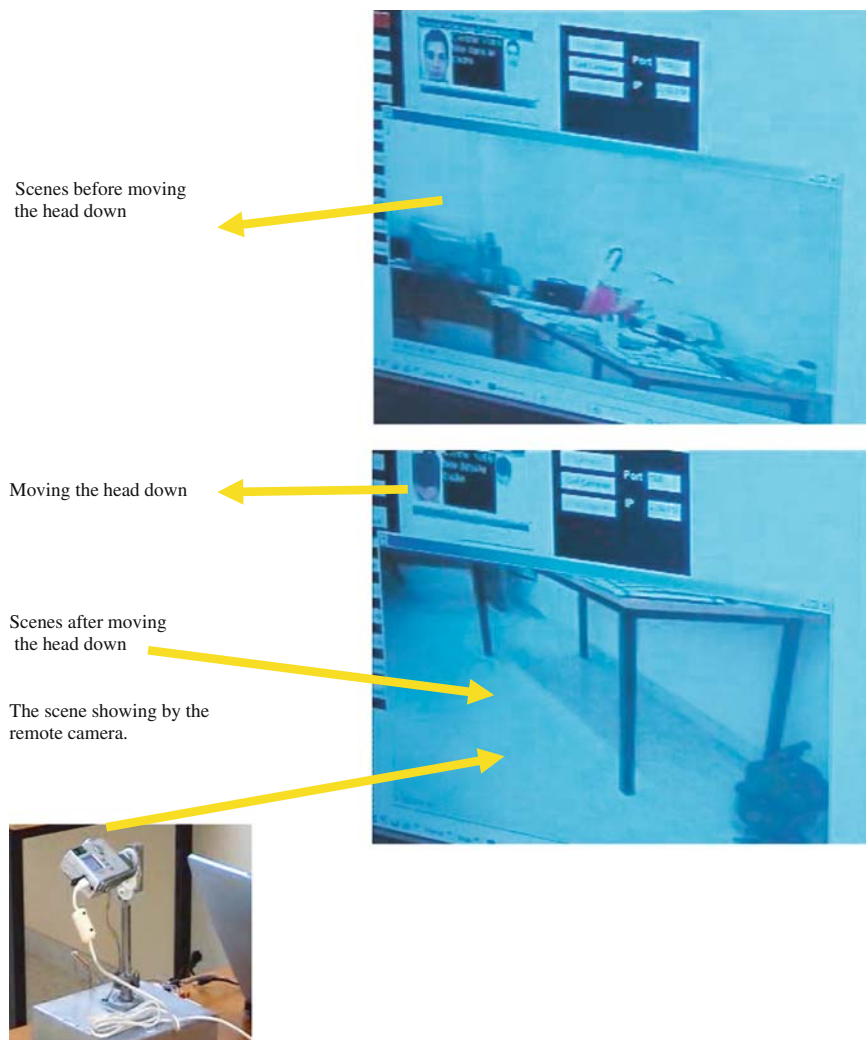
Moving the camera  
According to the  
orientation of the  
head



*Figure 9.* The camera (server side), carried by the robot with two joints, orientates itself according to the orientation of the head “client side”.

we transmit the data to the server. Figure 8, we notice that the results of the neural network appear to us on the notice board.

- **Connection:** It is necessary to be connected to the server i.e. to the robot to transmit the appropriate command via the computer network. Figures 9 and 10 show how the decision influences the command of the robot carrying the camera and the scene received “client side” changes according to this decision taken by the neural network. All the computer programs presented will be illustrated in the form of an organization chart in Figures 11, 12, 13 and 14.



*Figure 10.* Scenes transmitted to the server become dynamic i.e. a scene that changes according to the orientation of the client head. At the top, it is a scene transmitted by the distant server and when we descend the head client side, a new scene is gotten automatically since the distant camera orientates itself to the low.

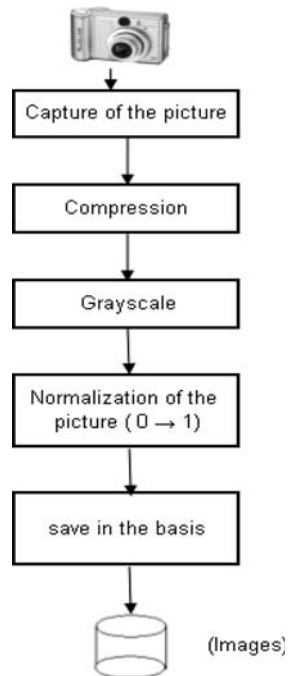


Figure 11. Preparation of the training set that includes faces in the five positions (high, low, right, left and face).

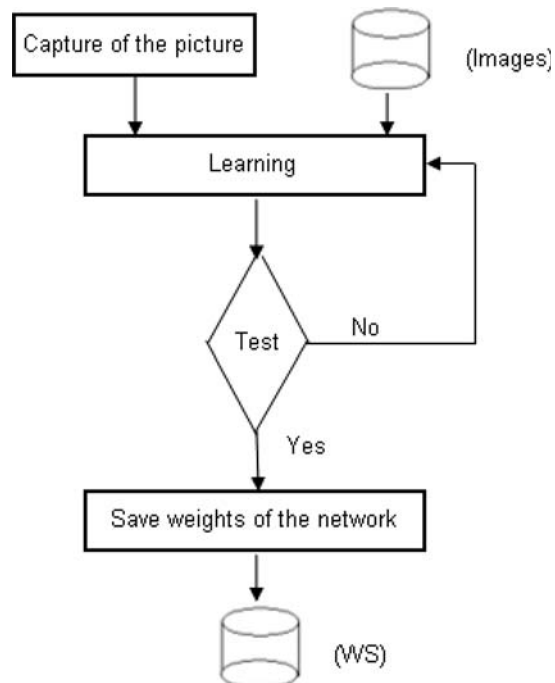


Figure 12. Training of neural network of training set prepared and including faces in their different positions.

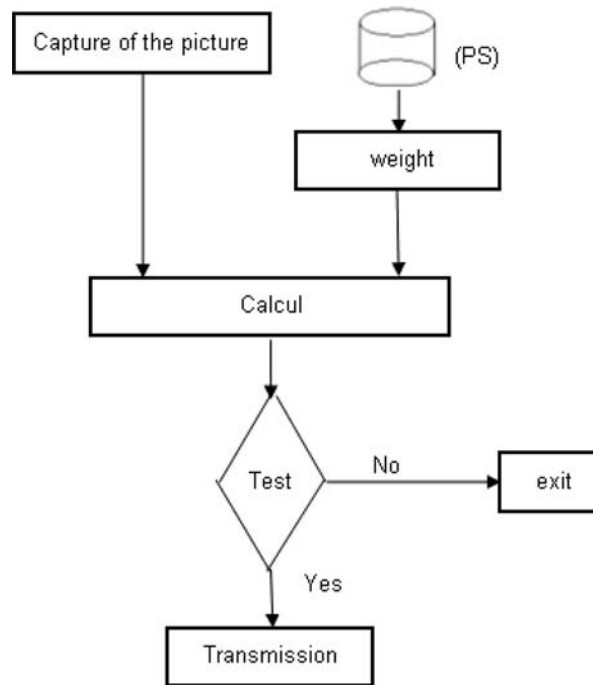


Figure 13. Recognition of a picture by the neural network learned off-line and sending the decision to the server to control the distant camera.

## 5. Conclusions

We have designed and developed robotics system with two joints provided with a camera transferring a scene (image) from one site to another. The distant camera is oriented according to the orientation of the client's head supervising the scene. An intelligent system of treatment and analysis of pictures is used to order the two joints of the robot. When implementing the robotic intelligent system, we faced the following problems:

- (i) The mechanical construction of the robot with two joints while respecting the mechanical norms (gearings – rubbing...).
- (ii) The research and design of the electronic command circuit Driver based on the utilization of transistors.
- (iii) The development of the interface circuit of the PC – motors that must include an opto coupler permitting to protect the parallel port against high voltages and to increase the applied voltage to boundary-marks of motor spools. The opto-coupler must be used since the parallel port of the PC provides the strength (5V to 5mA) that is incapable of saturating the transistor in the driver circuit.

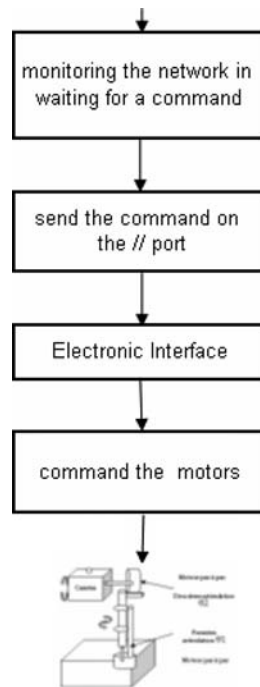


Figure 14. Receipt of the decision via the computer network and orientation of the camera.

- (iv) The modeling of the neural network that permitted us to recognize the learned pictures. The training has been done under MATLAB and the real-time recognition in real time has been performed under Visual Basic.

A lot of tests were needed and the results obtained were very satisfactory. The turnaround time of the system is very acceptable. In the context of a visual conference, we constructed a neural model of visual picture processing which was implemented on a robotic system. With the developed system, we succeeded in delivering back a remote image transmitted on a dynamic network i.e. a “server side” picture that changes its orientation according to the movement of the head of the observer “client side”. It has been achieved by utilizing an intelligent system of processing and image analysis using a learned neural network. The demonstrated interactive system of visual conferencing presents obvious advantages, such as the economy of time and money. Typically, remote teaching and similar applications can now be economically envisaged.

### Acknowledgement

This work has been sustained by the National Council of the Scientific Research (CNRS) of Lebanon and CEDRE program.

## References

1. Basri, R.: Recognition by prototypes, *International Journal of Computer Vision* **19** (1996), 147–168.
2. Daya, B.: A multilayer perceptrons model for the stability of a bipedal robot, *Neural Processing Letters* **9** (1999), 221–227.
3. Daya, B., Chauvet, P., Garenne, A. and Chauvet, G. A.: Learning rules for dynamic trajectories in the cerebellar cortex. Application to a bipedal robot, *Proc. of the IASTED, ACTA Press, NCI2004*, pp. 19–24, Switzerland, 2004.
4. Daya, B. and Chauvet, P.: A static walking robot to implement a neural network system, *Proc. of the 15th International Conference on Systems Science (ICSS)*, pp. 251–258, Wroclaw, Poland, 2004.
5. Daya B., Hajjar, M. and Chauvet, P.: A new system inspired from the cerebellar cortex to control a bipedal robot, *WSEAS transactions on Systems* **4(4)** (2005), 461–467.
6. Edelman, S., Ctzu, F. and Duvdevani-bar, S.: Similarity to reference shapes as basis for shape representation, In: G.W. Cottrell (ed), *Proceedings of first Annual Conf. of the Cognitive Sciences Society*, pp. 260–265, San Diego, CA, 1996.
7. El Kharki, O., Sadgal, M. and Ait Ouahman, A.: Traitement d'images Bas Niveau: Algorithmes, Implémentation et Etudes comparative, *SEI' 99*, In: E. Norman (Ed.), Fès, Maroc, 1999.
8. El Kharki, O., Sadgal, M., Ait Ouahman, A., El Himdy, A. and Ait Belaid M.: Automatic cartography from aerial images (Site of Salé, aMorocco), *International Archive of Photogrammetry and Remote Sensing XXXI0II* (Part B2) (2000), 179–183.
9. El Kharki, O., Ait Belaid, M., Ait Ouahman, A., Sadgal, M. and Bijaber, N.: Classification of multispectral image using ISODATA algorithm, *AMSE Journal*, paper N° 01315 (1B) (2002).
10. Garnesson, P., Giraudon, G. and Montesino, P.: *Messie: un système multi-spécialiste en vision Application en imagerie aérienne*, Research internal report, INRIA, Avril 1999.
11. Sadgal, M. and El Fazziki, A.: Un Modèle de Catégorisation dans les Systèmes de Vision, *TELECOM' 2001 & 2èmes JFMMA*, Casablanca, 17–19 Octobre 2001.
12. Sadgal, M. and El Fazziki, A.: Une Architecture Multi-Agents Basée Sur la Catégorisation dans le Traitement d'images, *International Conference on Image and Signal Processing*, pp. 608–617, Agadir, May 3–5 2001.
13. Walter, V.: Automatic classification of remote sensing data for GIS database revision, *In IAPRS*, **32** (Part 4) (1998), 545–552.
14. Weidner, U. and Förstner, W.: Towards automatic building extraction from high resolution digital elevation models, *ISPRS Journal* **50** (1995), 38–49.
15. Won, C. S. and Derin, H.: Simultaneous parameter estimation and segmentation of noisy and textured images using Markov random fields, *CVGIP: Graphical Models and Image Processing* **54(4)** (1992), 308–328.