

# **Blind Assistant PSP**

## **Core CFW0006**

**User guide**

**Visilab Research Center – University of Messina – Italy**

---

## Chapter 1

# What is Blind Assistant

Blind Assistant is an attempt to realize a software that can help men and children that are *blind* or *visually impaired*. The software is designed to run not on complex and expensive hardware devices, but on simple and cheap devices as Sony Playstation Portable (PSP).

Blind Assistant uses *Nanodesktop tecnology* for windows environment, it is able to speak with the user using *ndFLite*, a porting of an open source speech synthesis engine and it can be controlled by the blind through his voice, thank to the support of *ndPocketSphinx*, a software for voice recognition. Both Flite and Sphinx are projects originally developed by Carnegie Mellon University (CMU) and in a second moment ported under Nanodesktop by Visilab Research Center developers.

At the actual stage of the development, Blind Assistant is able to provide the following functions:

- a) face recognition
- b) position recognition
- c) optical char recognition
- d) color scanner
- e) mail reader
- f) data matrix scanner (see <http://www.libdmtx.org/>)
- g) voice clock/calendar

Blind Assistant can recognize the names and the positions of the people that are present in that moment in your room, using complex algorithms (as Intel EigenFaces or PentLand PCA).

The face recognition technology is provided using *ndOpenCV*, a porting of the original Intel OpenCV libraries realized by the author of the program.

Filippo Battaglia's version of ndOpenCV improves the Intel code as the algorithms have been modified in way to use EMI (Enhanced Mathematical Interface), an API integrated in Nanodesktop and written totally in assembly. EMI is able to use Sony VFPU to speed up the needed computations.

Blind Assistant can also recognize the room where the blind is. This task is executed through a component called *BlindServer*, that runs on an x86 server. The client/server architecture of Blind Assistant allows the program to execute heavy and complex algorithms like *SIFT*. In this way, we can recognize in which room the blind is.

Blind Assistant is also able to read a text. In this case, the software uses an OCR that runs in remote (on the x86 BlindServer) or locally on the PSP, in combination with the voice synthesizer ndFLite.

A color scanner is able to recognize the average color of the image that is grabbed by the camera.

A voice mail reader is able to read the text of a mail to the blind.

The data matrix scanner can be used for the recognition of objects, walls, obstacle, tickets etc.

The clock/calendar is useful if the blind wants to know what time/day it is.

## The author

Blind Assistant has been developed by Filippo Battaglia, an electronic engineer of the Visilab Research Center, graduated at the University of Messina (filbattaglia@libero.it).

---

### The current version

This is an official version of Blind Assistant. The program will work *only on custom firmwares*.

In the zip package is contained a version for PSP-SLIM and a reduced version for PSP-FAT.

Starting from version CFW0006 of the program, both versions share the same source code. However, the FAT version has several functionalities disabled, because of the smaller amount of memory that is available.

The author has tested Blind Assistant on the custom firmware version 3.71 M33-4, 3.80, 3.90 M33-3, 4.01 M33. The program should work also in the following versions of the custom firmware.

This version of Blind Assistant has been designed to use the *Sony GoCam camera* [<sup>1</sup>].

---

<sup>1</sup> In alternative to Sony GoCam, you could use *Eyeserver technology*. This choice is obsolete and now it is deprecated. The binary code that we've provided still contains the required components for Eyeserver connection, but the full working is guaranteed no more.

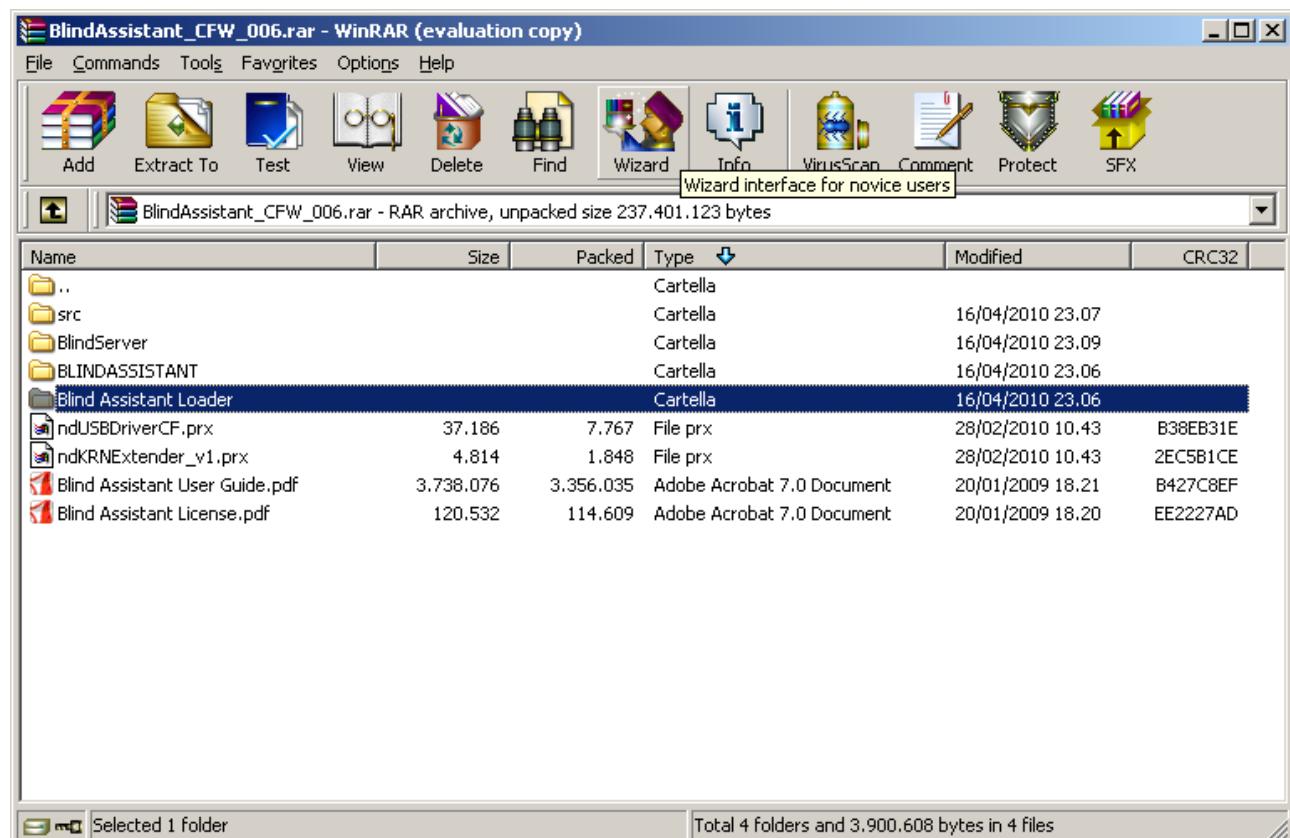
## Chapter 2

# Install Blind Assistant

Download the zip package from Nanodesktop website.

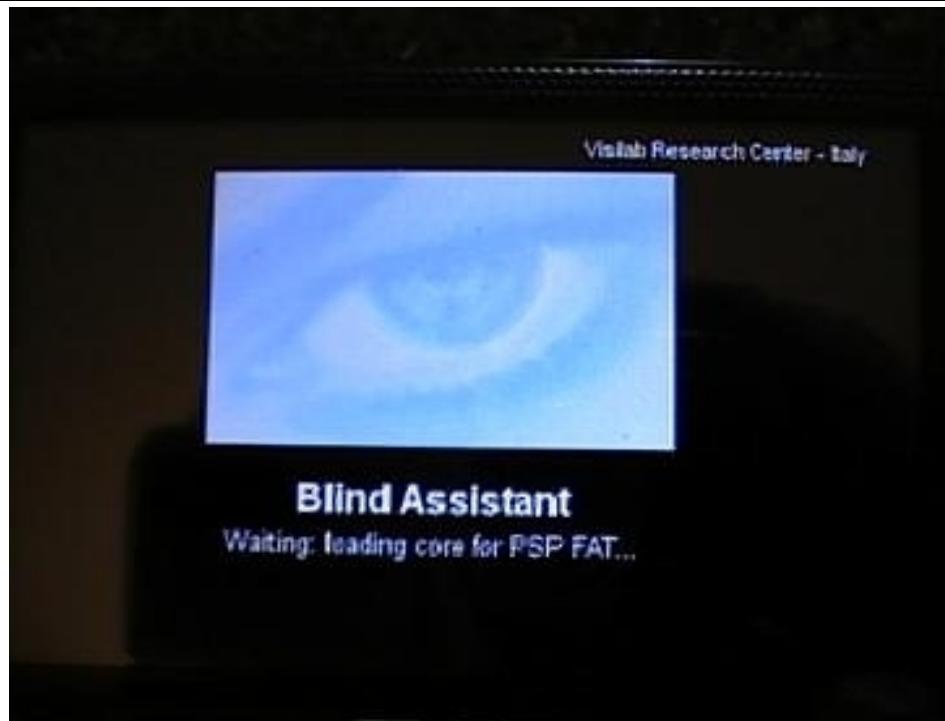
<http://visilab.unime.it/~filippo/BlindAssistant/BlindAssistant.htm>

The content of the package is this:



- Copy the files *ndKRNExtender\_v1.prx* and *ndUSBDriverCF.prx*, in the root folder of the memory stick.
- Copy the folder *Blindsight Loader* into the directory PSP/GAME of the memory stick
- Copy the *Blindsight* folder into the root directory of the memory stick

Now, connect the GoCam to Sony PSP and run the homebrew **Blindsight Loader** from the dashboard of the PSP.



At beginning from BlindAssistant CFW0004, the program is provided with 2 different *cores*: one for PSP FAT and one for PSP-SLIM. Each core is optimized for the features of the console: for example, in the core for FAT some features don't exist, since the system has less memory than SLIM.

Blind Assistant Loader shall load automatically the right core for your model of PSP.



This is Blind Assistant core program:



Blind Assistant in action!



---

## Chapter 3

# Use Blind Assistant

Starting from Blind Assistant CFW005, the blind is able to control his unit simply using his voice. This is possible thank to the *ndPocketSphinx* library, a technology released with Nanodesktop 0.3.6.

If you want to obtain a list of commands available on the actual release of Blind Assistant, you can go to the *Misc* menu, and select the item *List of voice commands*.



The system shall begin to list the voice commands that are recognized by the system:



The actual commands are:

- Enable no function mode
- Enable face recognizer [<sup>2</sup>]
- Enable position recognizer
- Enable optical char recognizer
- Enable color scanner
- Enable data matrix scanner
- Enable mail reader [<sup>3</sup>]

These commands allow to switch the working mode of the system. If you want to change the mode, you must press the SQUARE button of the console. The system shall enter in *listening mode*:



A short beep shall inform the user that the system has begun to listen from the GoCam Microphone. Say the command that you want and the system will react within few seconds.

It is possible that the recognizing process fails: in this case, Blind Assistant shall say that *The command is not recognized*, and it will return to the normal mode. Press the SQUARE button again to execute a second time the procedure.

## Manual switching

The voice recognition system can fail in several occasions: the causes can be noise, the poor quality of the microphone integrated in GoCam, or... simply your particular voice.

Starting from CFW0006 version, a manual switching mode has been added, allowing the blind to switch between the several functionalities without using his voice.

The key of this new function is the couple of trigger buttons at the back of the console.

Pressing the right button, the system automatically passes to the following function, while pressing the left button, it passes to the previous function. [<sup>4</sup>]

<sup>2</sup> Not available under PSP-FAT

<sup>3</sup> Not available under PSP-FAT

<sup>4</sup> If a function is not available on a given platform, the system jumps it and switches automatically to the following.

## Switching mode using the GUI

It is possible to select manually the desired mode, too.  
Simply go to the *Fnc* menu, and select the function that you want.



## Clock/calendar

In any moment, the blind can obtain the date and the time from Blind Assistant. In this way, he doesn't need to bring a Braille clock with him [5].

Simply, keep pressed for 2 seconds the button START of the console: the system will communicate date and time:



The **clock/calendar** function can be invoked in any moment.

<sup>5</sup> The date and the time communicated by Blind Assistant are the same stored in PSP custom firmware. Our program has no dialog box to adjust date/time so, if you want to modify them, use the interface that is provided by PSP custom firmware.

---

## Chapter 4

# Face recognition mode

The face recognition mode allows to the blind to recognize the name and the position of the people present in that moment in the room. **This functionality is available only for PSP-SLIM users.**

The system works on a set of 10 people: for each of them, there are 8 stored samples: this means that the system can work on 80 images in memory. The recognition system uses PentLand (PCA) and Viola-Jones technology and works in real time.

### Important note

In the versions of Blind Assistant up to CFW0005, the data related to the several faces were loaded in a single passage at startup.

Starting from Blind Assistant CFW0006, a new memory manager has been implemented: it loads the data for the faces only when they are required.

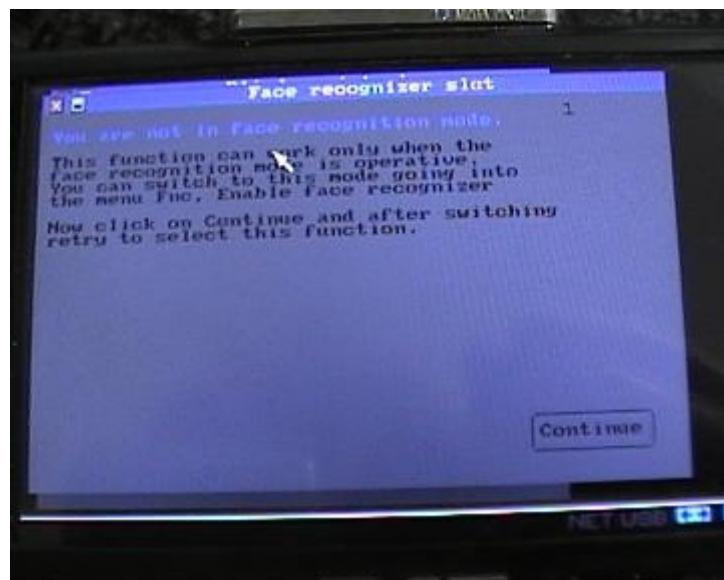


When the user exits from the *face recognition mode*, the data are automatically unloaded and the memory is made available for the other modules.



A consequence of this new way to manage the memory is that: you can start the training wizard, the test dialog-box, and load/save wizard, **only if you're already in face recognition mode.**

So, you must switch to **face recognition mode** before starting the training procedure for the new samples. If you haven't done this, you'll receive a message like this:

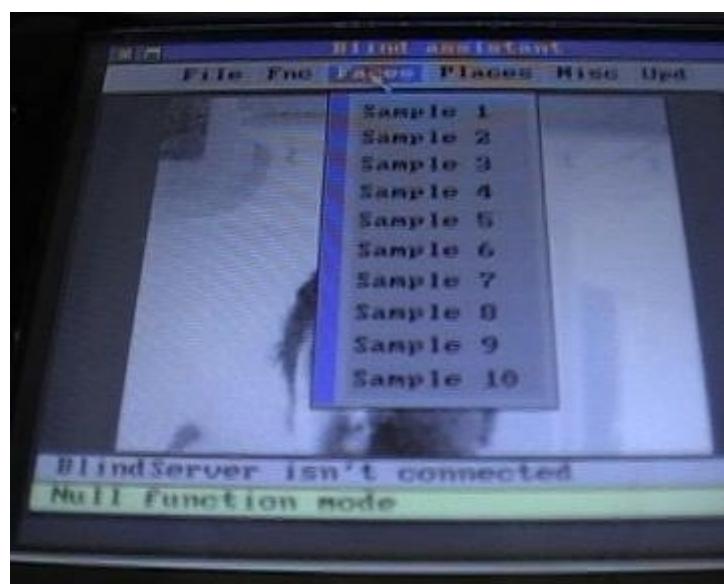


## Training

The first thing to do if you want to use the face recognizer module is the *training* of the system.

This procedure is driven through a wizard, and it is designed to be executed by the personal that assists the blind. It is necessary because the program needs some informations about the names and the faces of blind's familiars.

Switch to **Face Recognizer Mode**, using the voice interface or the manual switching (see the previous chapter). After this, go to **Faces** and choose a **Sample slot**:



The system will indicate that there is no data in that slot.



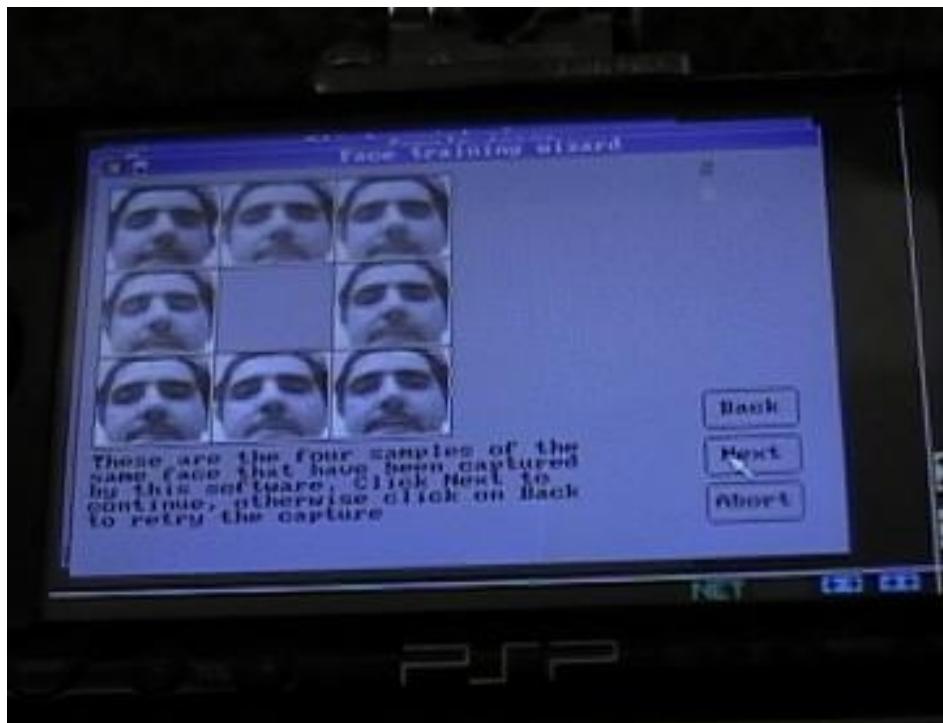
Click on **Training** button



Now, stay in front of your camera: when the system detects your face, it puts in evidence its using a red rectangle. When you see that the image has a good quality, click on **Next**. As you can see, the sample image is divided in four areas whose brightness is normalized separately: we've verified that this improves the accuracy of the recognition in case of not uniform lightning of the face.

*The operation must be repeated **for 8 times**: Blind Assistant needs 8 copies of each face to work correctly. The better results are obtained with images that are slightly different, with small changes of brightness and pose.*

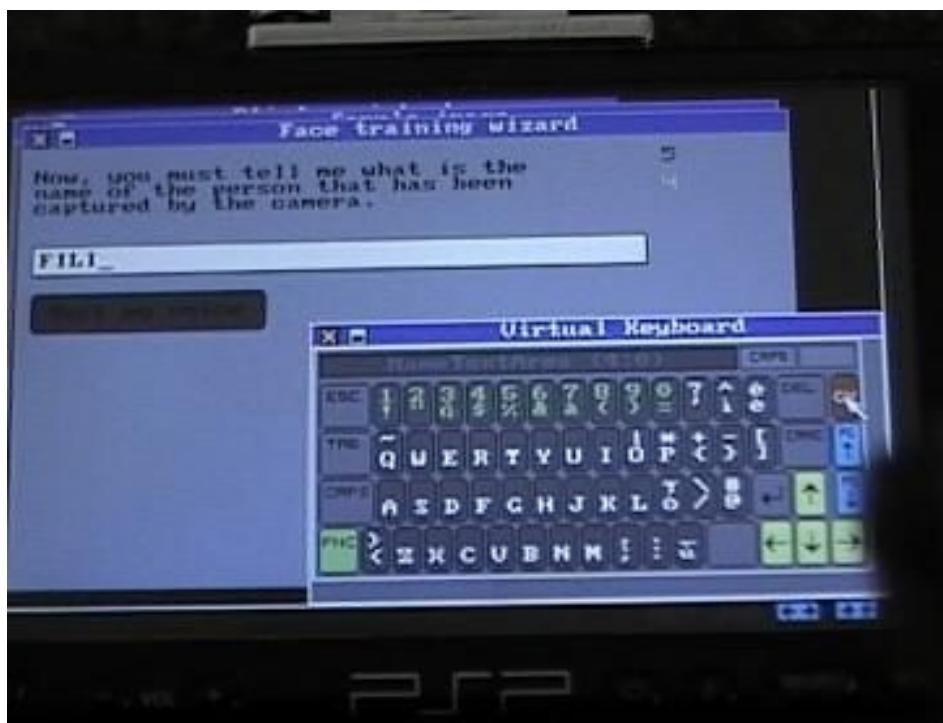
When the 8 samples have been grabbed, Blind Assistant will visualize a summary screen like this:



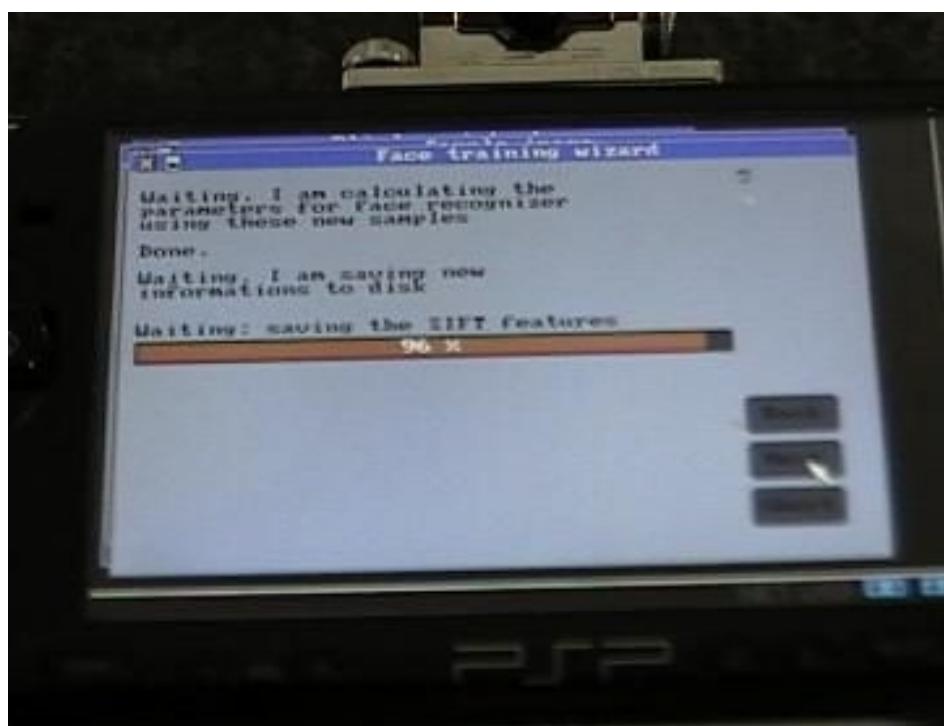
Blind Assistant will try to compensate automatically the variation of the brightness: in any case, we can suggest to you to save images with **different levels of brightness**. This can improve the recognition rate of the system.

Click on **Next** now.

Now, Blind Assistant needs the name of the person that you have registered. Use the virtual keyboard to enter this name.



Click on **Ok**, and after click on **Next**. The system will begin the elaboration:



Ok, the system says that the image has been stored:



Now click on **Exit** and return to the main interface.

The system is already in **face recognizer mode**: the system will begin immediately to recognize the face. Point the camera to a person: the face will be recognized.



When the system will recognize a face, it'll say the name of the person and the position.

The video engine cannot work with dark images: if the brightness isn't sufficient, the system will indicate the trouble. You can setup the correct brightness threshold in the *Engine Options* dialog box.

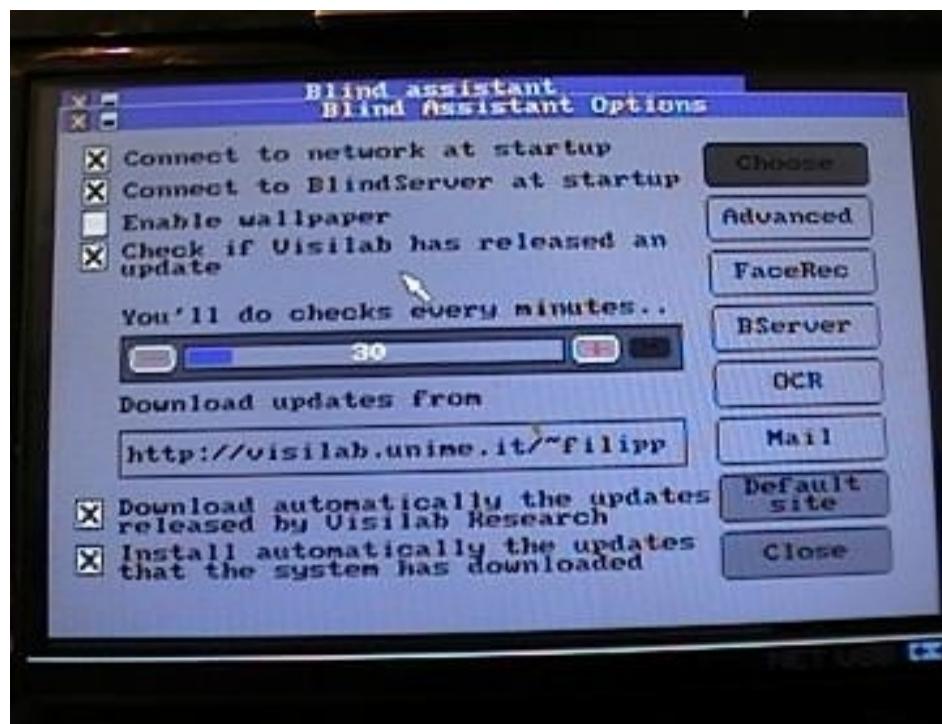
## The Options DialogBox

Now, we can see the dialogbox of the options. Go to **Misc/Options**

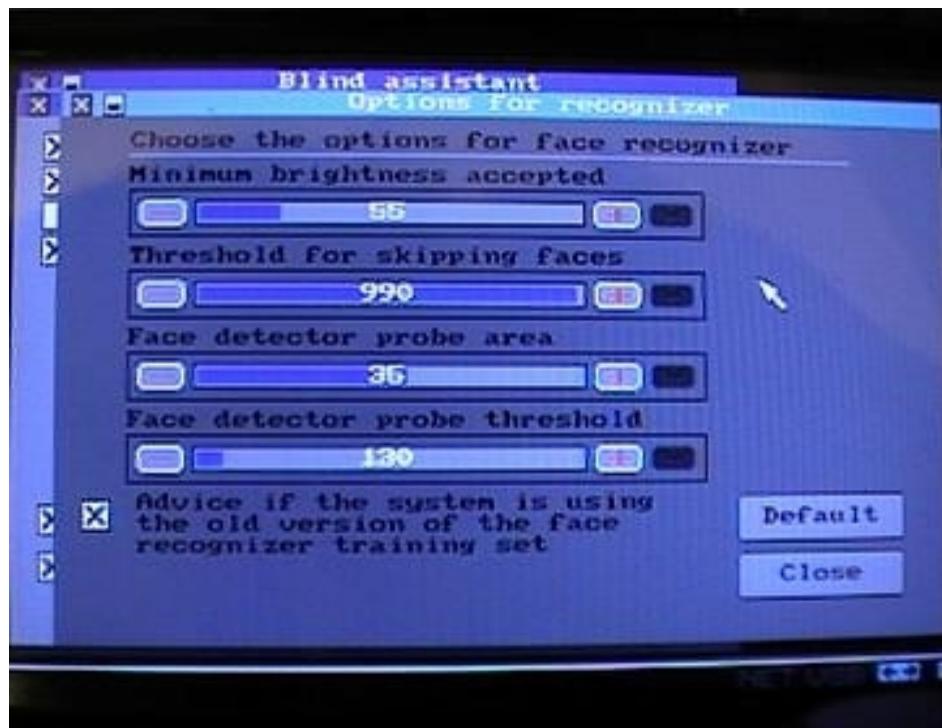


**Note:** when the option dialog-box is active, the video engine is suspended and no recognition operation is performed.

This is the **Options** dialog-box:



Press the **FaceRec** button.



There are 4 trackbars and a check box.

The former controls the *minimum accepted brightness*. When the brightness level of the image is lower than this value, the system will refuse to work and it will say that **the image is too dark**. This is necessary because the algorithms that are used in the program work very bad under low light conditions.

The second trackbar controls the *minimum threshold for skipping faces*. The result is considered to be

significative, only if the ratio between the distance of the nearest element and the distance of the second nearest element, is lesser than this value (divided by 1000). This means that, if the distances between the various faces are very similar, the result will be considered not significative and it will be skipped. The system believes to have recognized the correct identity of a person, only if the distance of the respective face is very lower than the others. If you raise this value, the results will be accepted by the system more frequently, but the recognition will become less reliable. If you lower this value, the result will be skipped more frequently, but they will become more significative.

The third trackbar controls the face detector probe area. If you do lower this value, the face detector will become slower and more sensible, but the system could exchange for a human face some particular on a wall. We advice you don't change this value.

The forth trackbar controls the threshold of the face detector. If you do lower this value, the face detector will become slower and more sensible. If you raise this value, it will become faster but less accurate. The choice is yours: we can suggest you don't change the value if you haven't a real need.

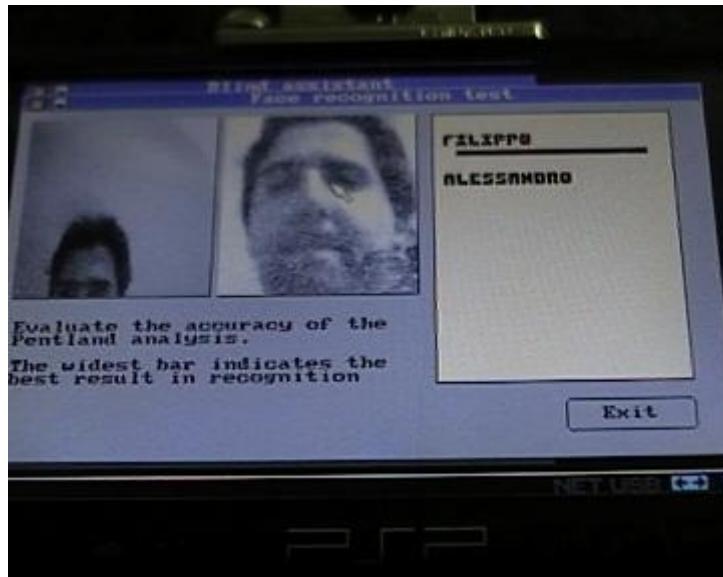
The last checkbox, states if the system must automatically advice the user that it is using an old version of the *cvhaar cascader*. It is advisable don't change this option.

## Test the face recognizer

You can test the face recognizer: in Blind Assistant there is a diagnostic mode that allows to see how changes the distances between an unknown element and the stored samples. In this way, you can see if the data in a face-slot are inaccurate and they must be reinsert repeating the training procedure.

The test system can be started only by the personal that assists the blind, as it is accessible only through the GUI. Move the mouse cursor and go into menu **Misc**. Select the item **Test the face recognizer**.

The following dialog box will appear [6]:



The bars in the rectangle at your right tell us the "score" obtained by any person in the database: the widest bar indicates the best result (the person that should be notified if the face recognizer was in action).

---

<sup>6</sup> The test system can be started only if you are already in face recognizer mode.

## Chapter 5

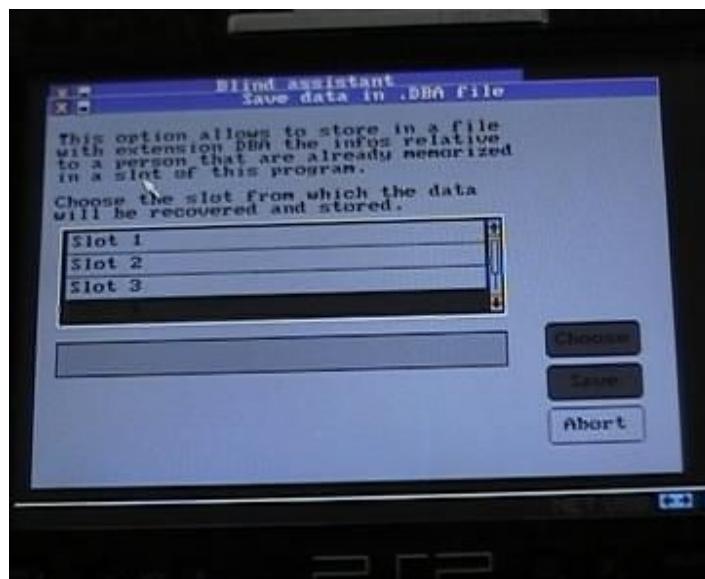
# Load or save the face data

Blind Assistant allows you to save or load the data about a person that have been stored in the training set, in a file .BDA. In this way, you can recover quickly the data about a person and you can save it in your memory stick.

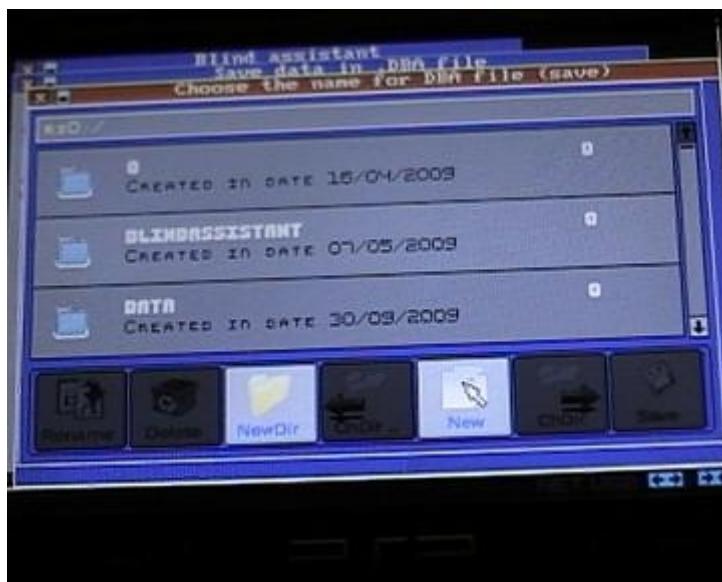
You can also transfer the data about a person, using Internet. A Blind Assistant unit can load the informations about a person from a file .BDA that you have downloaded by the net, and so it can become able to recognize a new face, without repeating the entire training procedure

## Save the face data

The first thing to do is to transfer the data, that you have stored in a slot, into a BDA file. Choose **File/Save [?]**

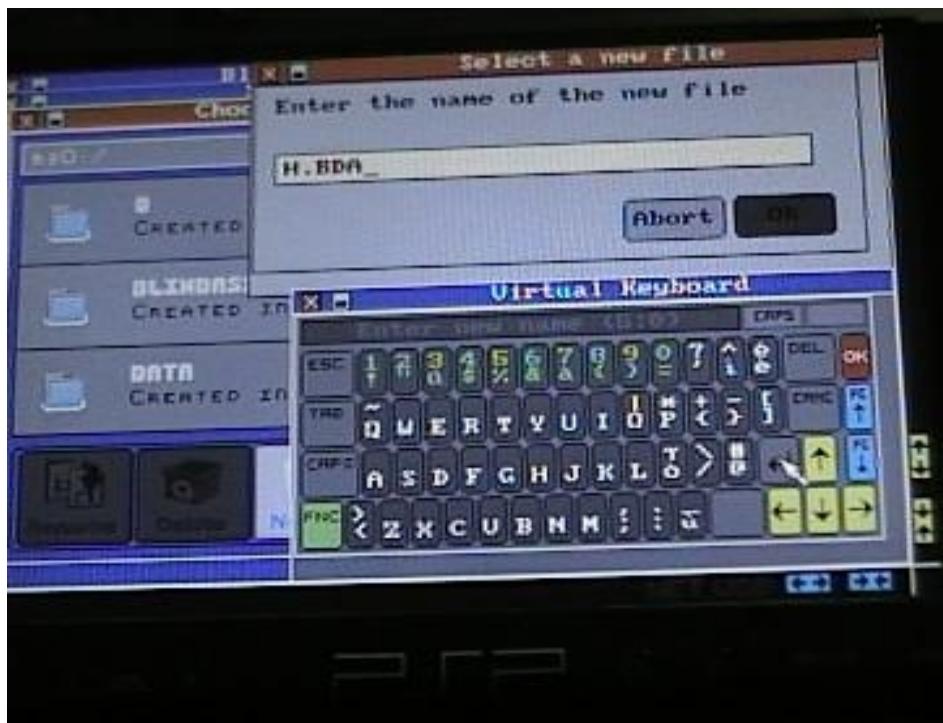


Now, choose the slot that contains the data that you want to store into the BDA file.

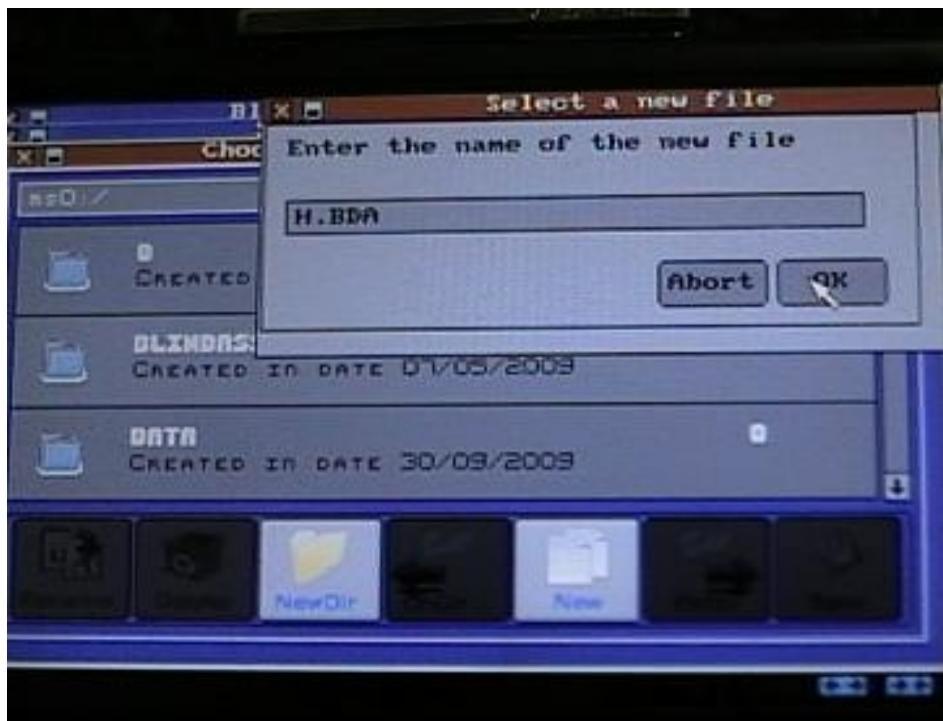


<sup>7</sup> The save/load operations are allowed only if you're already in **face recognizer mode**

Use the Nanodesktop file manager to choose the name of the new BDA file. If you press the button **New** in the manager, it'll launch the virtual keyboard and this will allow you to enter the new name:

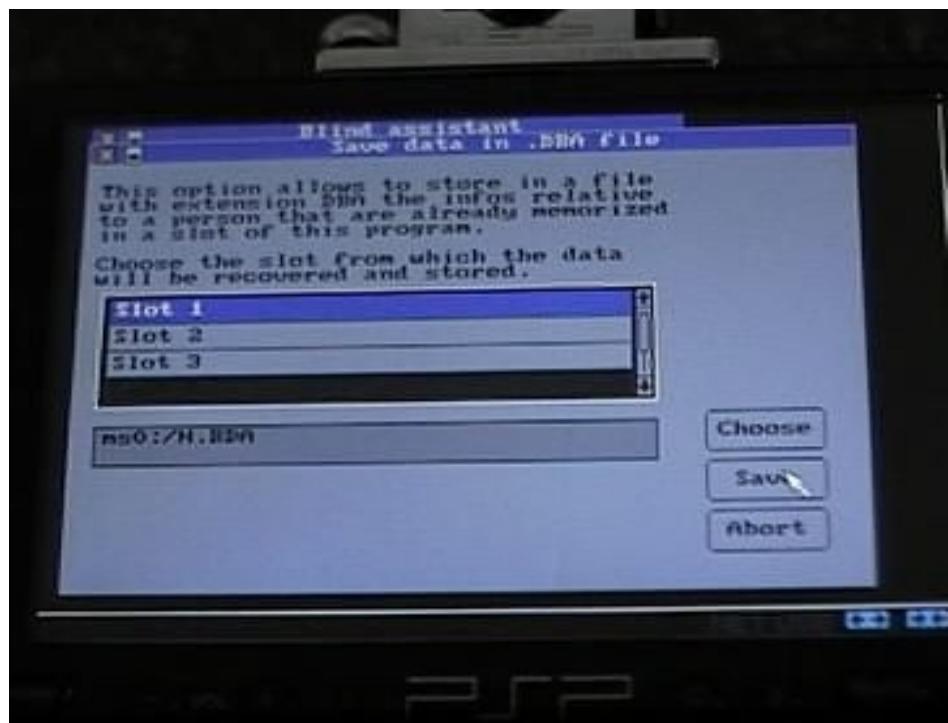


Click on the **red OK** button in the virtual keyboard. The entered name will be confirmed and the keyboard shall disappear:

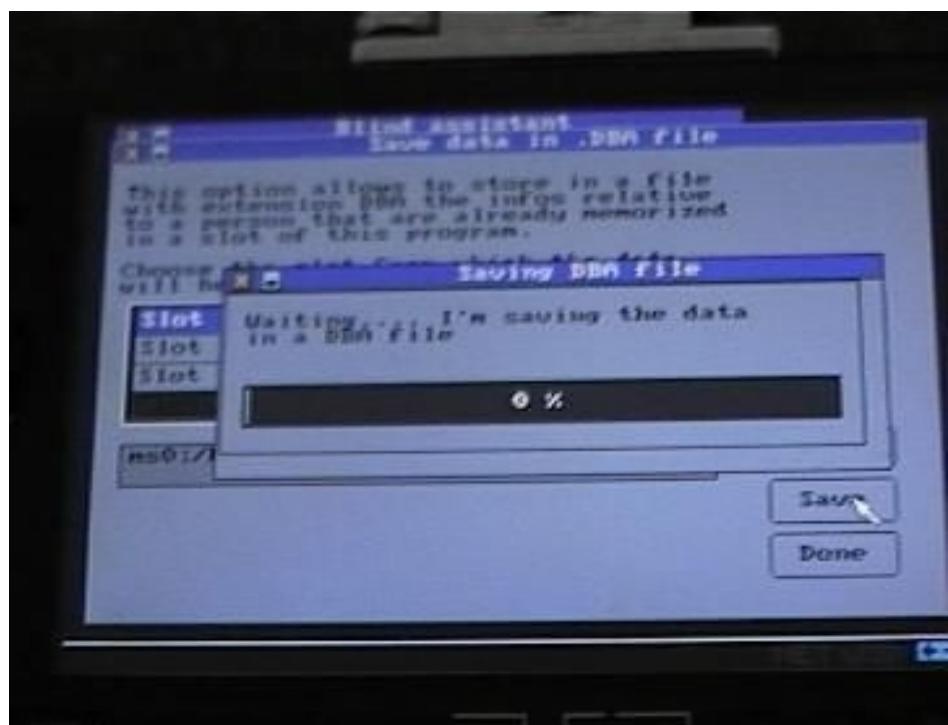


Now, press the **OK** button in the window.

And now, click on the **Save** button:



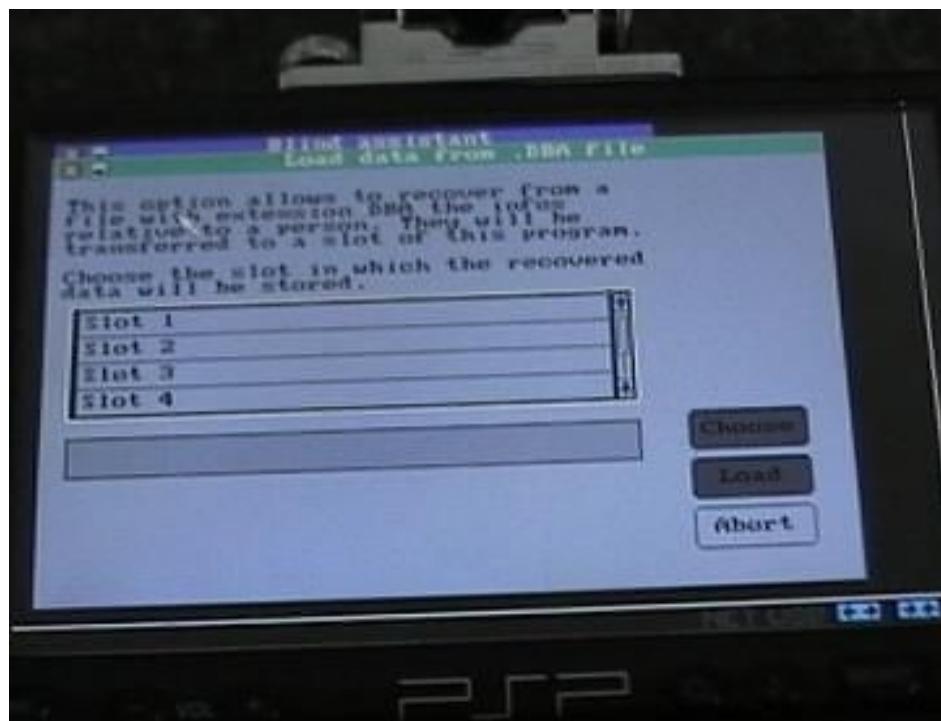
Nanodesktop will create the new file:



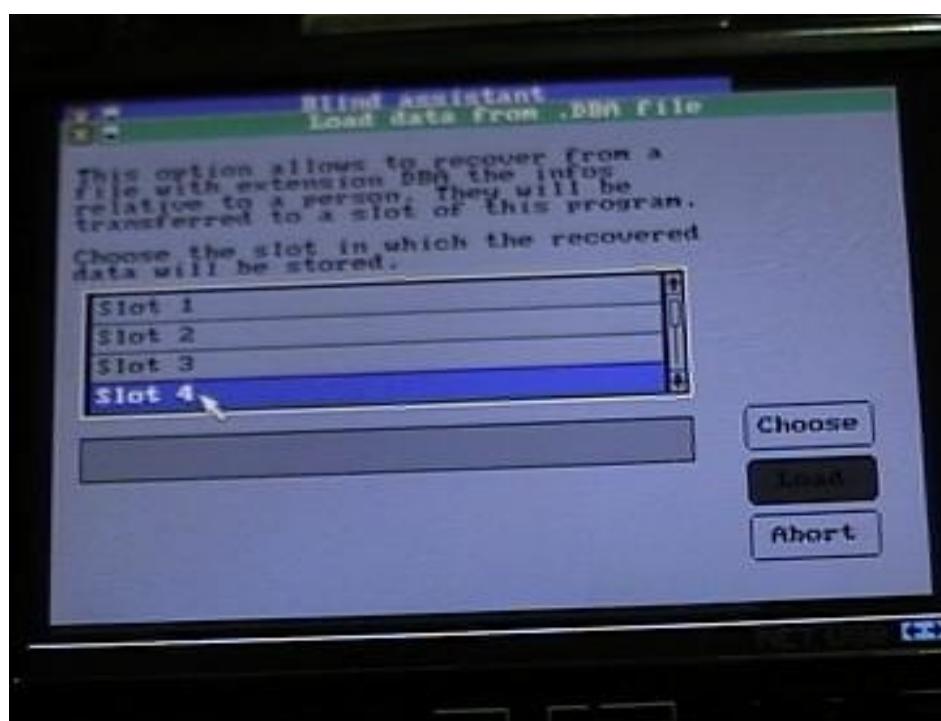
It is done! Now, a BDA file is present on your memory stick

## Load the face data

Ok, now we'll learn how to import the data of a person into the Blind Assistant program, simply loading a BDA file from the memory stick. Go into the menu **File**, item **Load**

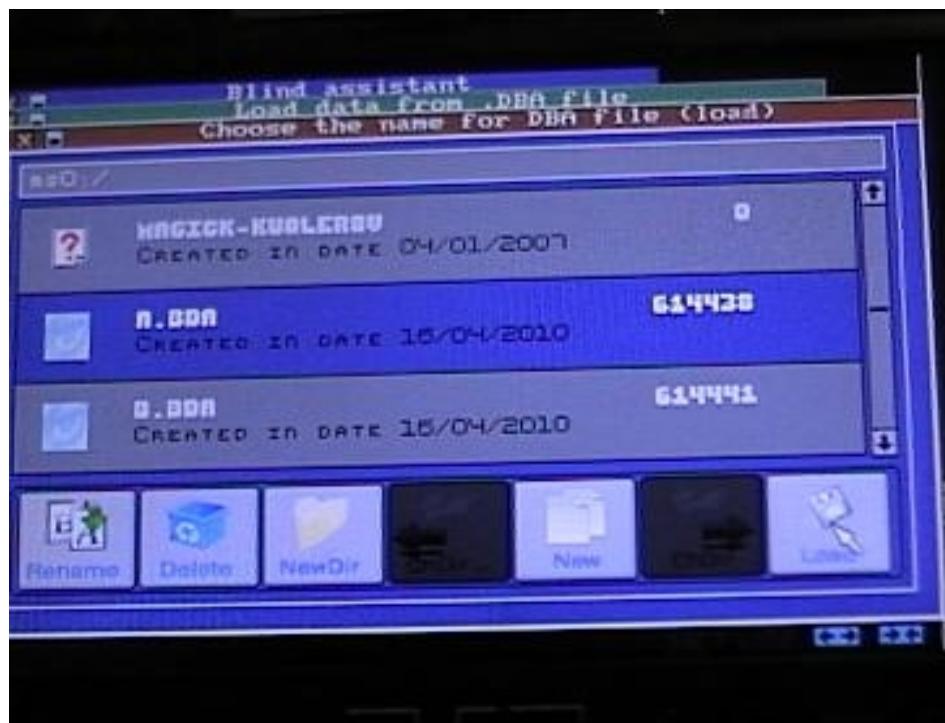


Choose the slot where you want to load the new data.

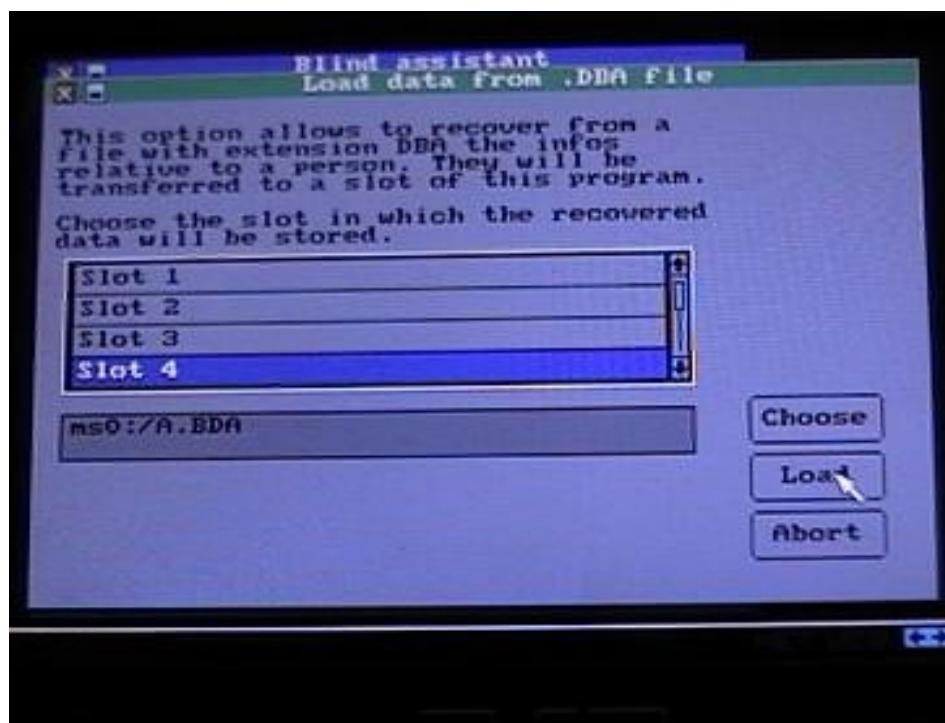


After that, press the **Choose** button.

Select the BDA file that you want to load.

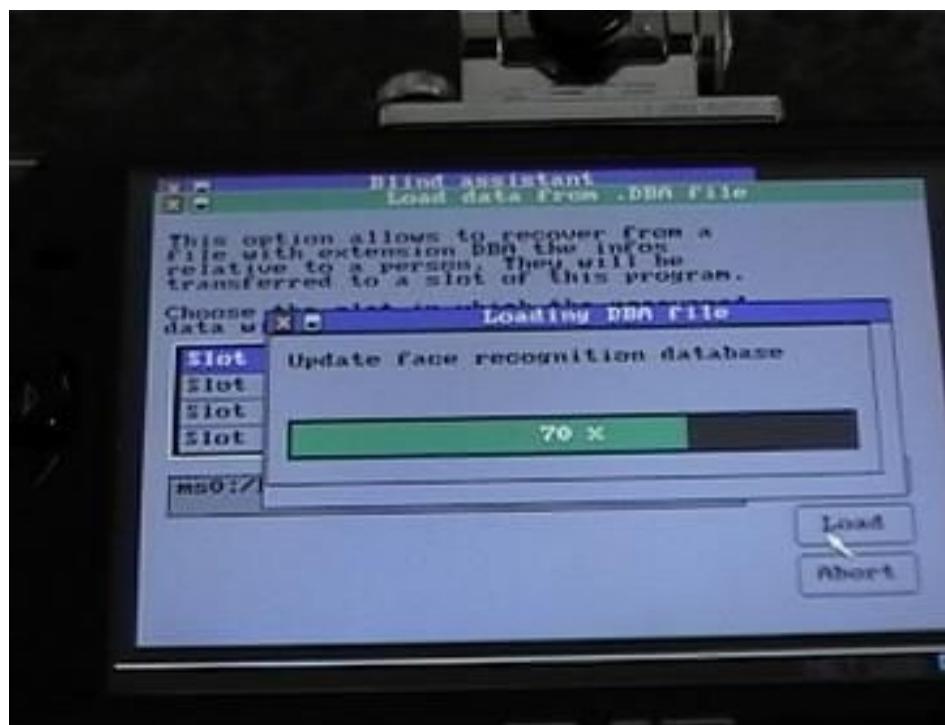


Pressing the **Load** button, you choose the file and confirm its name. The file manager will close and the previous dialog box appears:



Note that the chosen file *ms0:/A.BDA* is evidenced by the system. If you click on **Load** button, the system will begin to load the informations contained in the file that you wish into the memory slot

In this picture, we can see the loading process in progress...



At the end, the system will be able to recognize the new face that we've inserted in the database through the loaded .BDA file.

## Chapter 5

# Place recognition

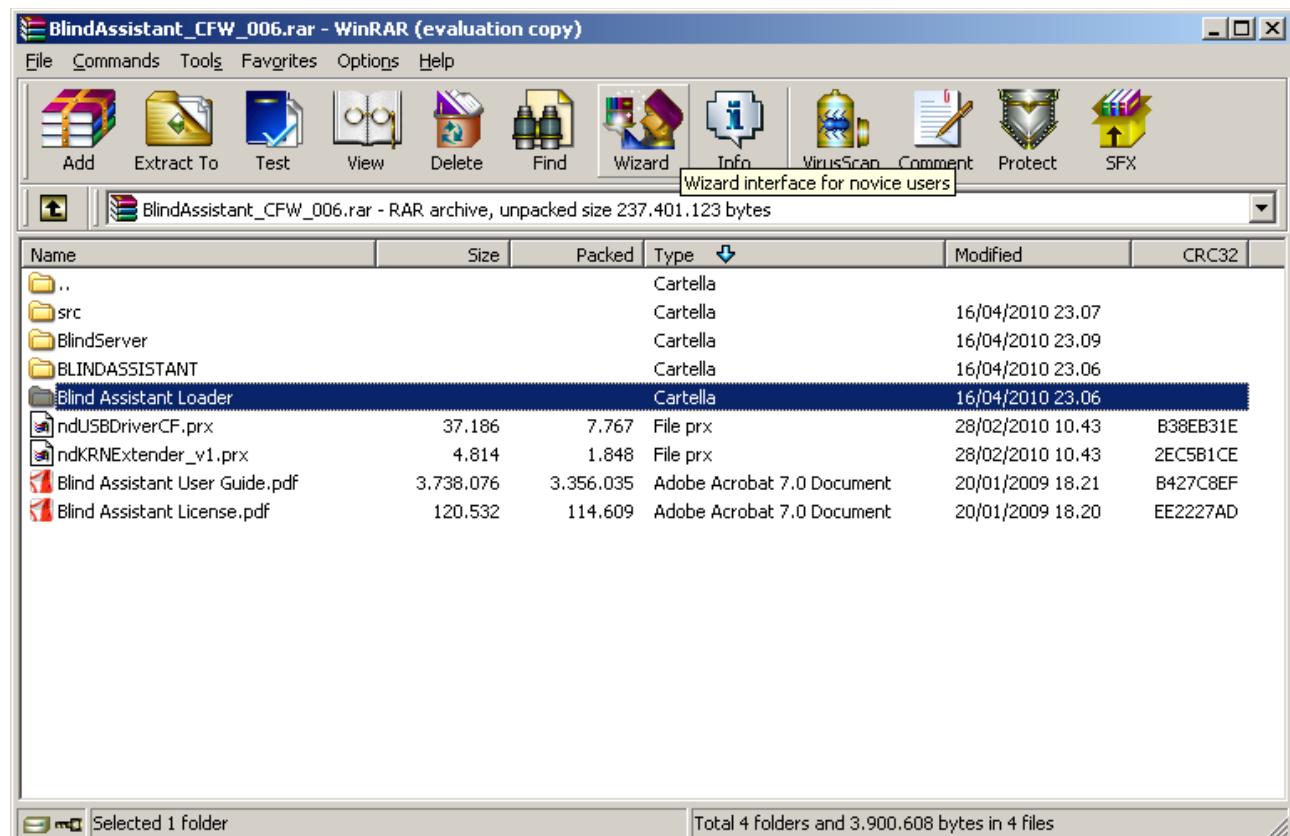
Blind Assistant can be used also to recognize the place where the user is. This feature has been added starting from core CFW0002.

The recognition of the place is executed through a SIFT comparison between the image sent by Sony PSP through the wi-fi connection, and a set of samples that have been stored in the server in a previous period.

Unfortunately, the process of recognition requires a very, very large computation power, so the small processor of the PSP device cannot execute alone the entire algorithm. For this reason, Filippo Battaglia and Visilab Research have developed a software for x86, called **BlindServer**, that runs on a x86 processor and that is connected to PSP through a WI-FI connection.

The entire system is a client-server architecture, where *BlindServer program runs on a x86 PC and it is the server component; Blind Assistant program runs on the PSP and it is the client component*.

**BlindServer** is released in the same package of BlindAssistant. It is distributed in three version: the first is for generical x86 cpu, the second is for processors that are compatible with SSE instructions set, and the third is for processors that are compatible with SSE2 instructions set.



## Step 1: Install BlindServer under Windows

In order to install BlindServer, you must have a PC with Microsoft Windows operating system and a wireless network connection. The communication between client [<sup>8</sup>] and server can happen in two ways:

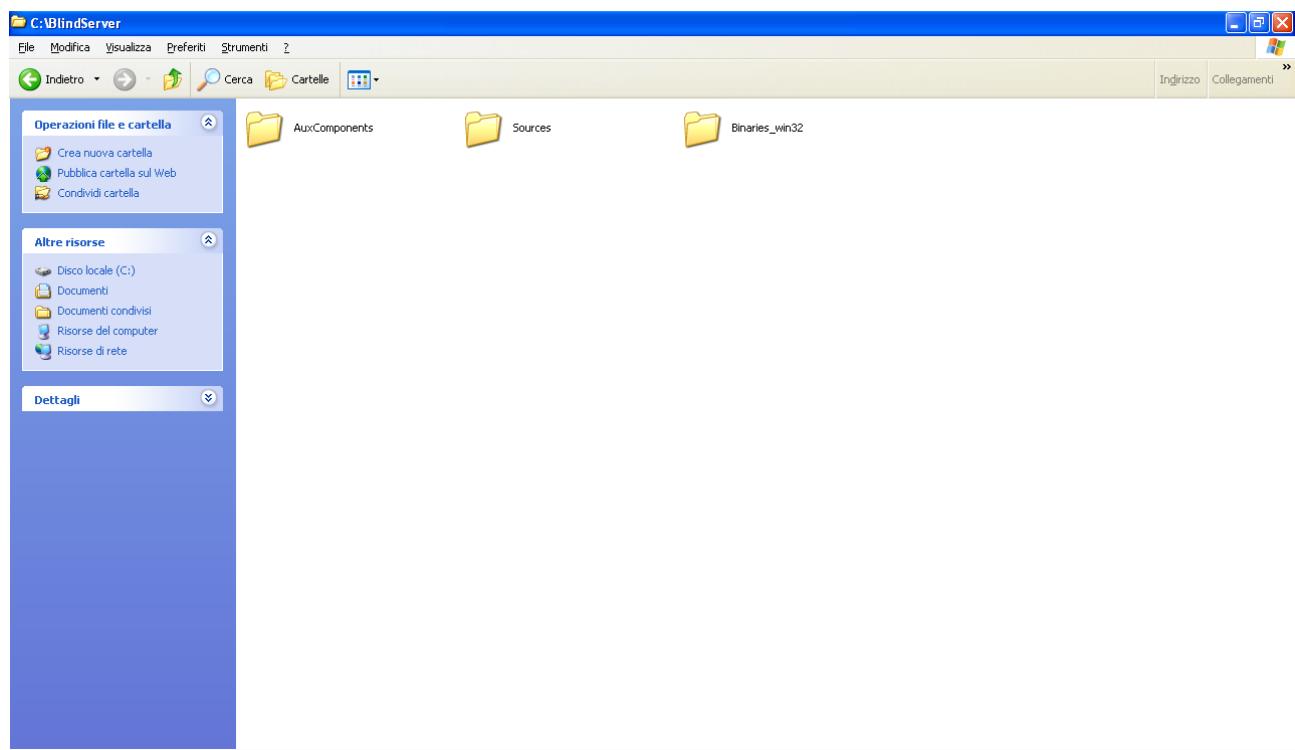
a) if you have a working wi-fi network in your home, the PSP can connect to the x86 server using it. It is imperative that the PC running BlindServer uses a *static IP address*

b) you can use BlindAssistant client also in an open space, as a road, or a park. If your PSP uses a public WIFI connection (like *The Fonera*, for example), BlindServer can be contacted by your console via internet. It is possible, but you must ensure that *your internet provider assigns a static IP address to the PC that runs the BlindServer program*. In this way, the BlindServer will be identified on Internet by a fixed IP address.

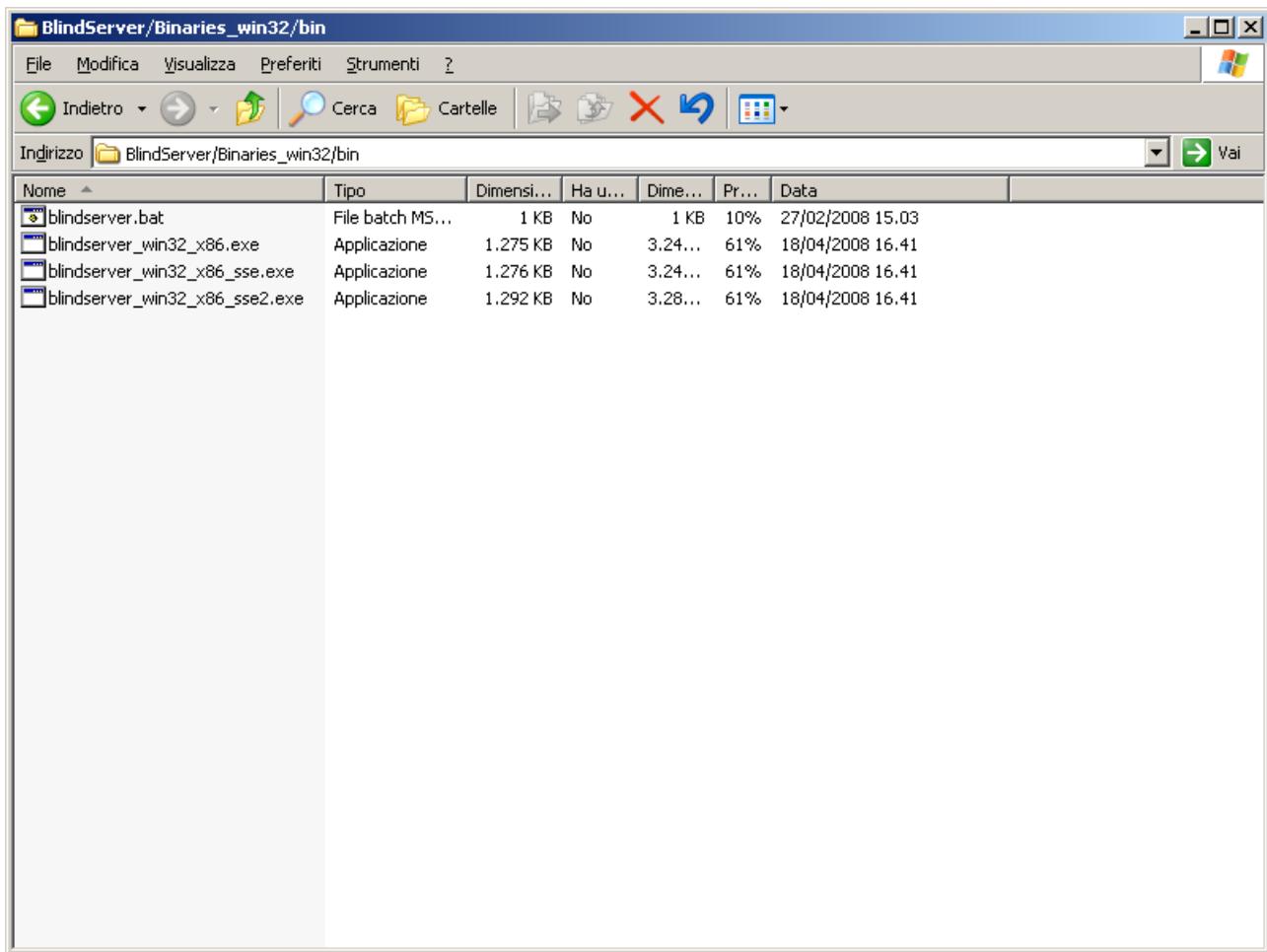
Create a directory in your C:\ folder and call it with a name (for example BLINDSERVER).

**md BlindServer**

Copy the content of the folder BlindServer (contained in the zip package) in the folder [C:\BLINDSERVER](#) of your PC.



Go to the subfolder *Binaries\_win32\bin*:



You can see that there are 3 versions of BlindServer.Exe.

The first is designed for a *generic x86 processor, like a Pentium II, or an AMD K6-3*.

The second executable is compiled for processors that use SSE instructions (like Intel Pentium II/III, or Athlon).

The third executable is compiled for processors that use SSE2 (like Intel Pentium IV or Athlon64).

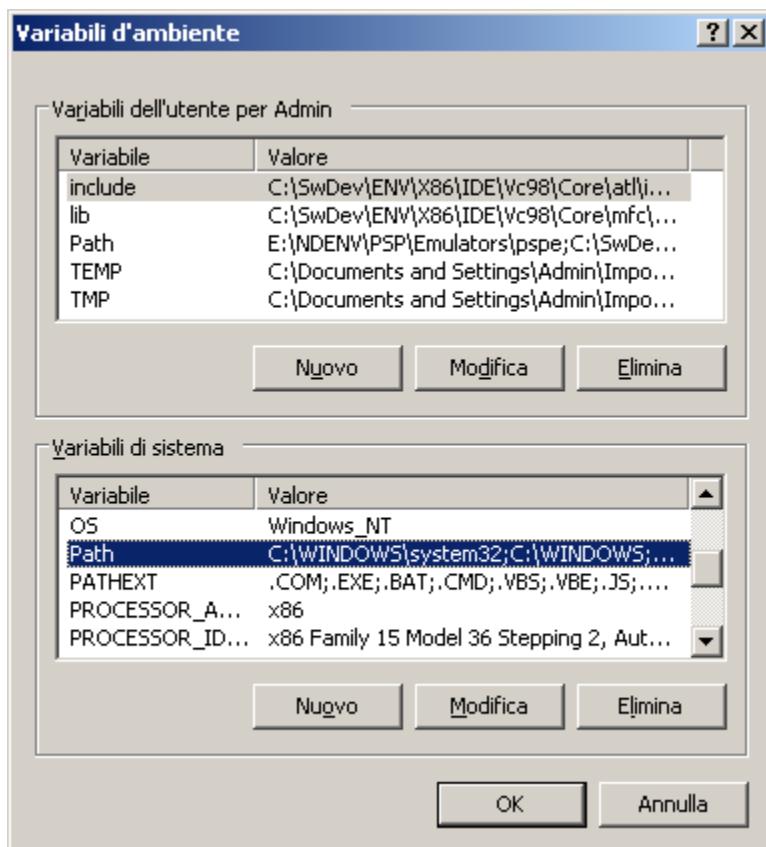
*The user must choose the right executable for his processor.*

## Step 2: Install Cygwin

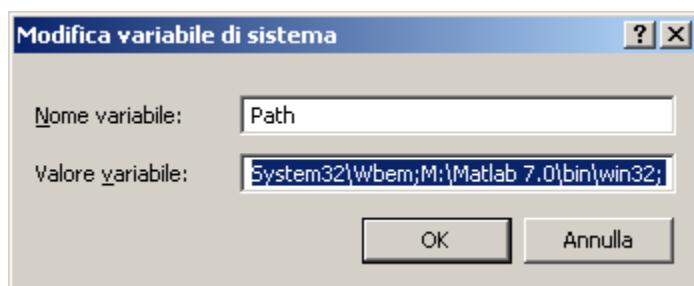
Go to Cygwin website and install a standard Cygwin distribution.

## Step 3: Configure Windows

Now, you have to change the PATH variable in Windows. Go to *Control Panel/System/Environment Path*:



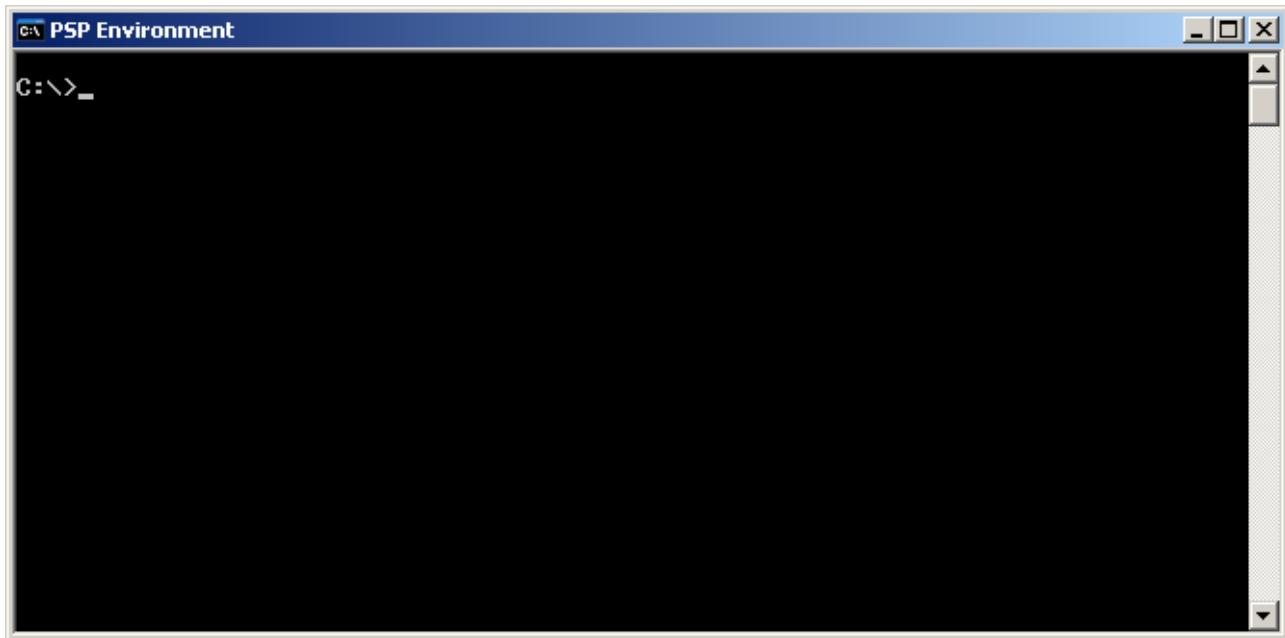
Change the System Path:



Add the folder for cygwin dll: for example, if your cygwin installation has been done in the folder **C:\CYGWIN**, add to the PATH variable, after the ; character, the string **C:\CYGWIN\BIN**.

## Step 4: Start BlindServer

Now, open a *Command Prompt* session and go to the BlindServer folder:



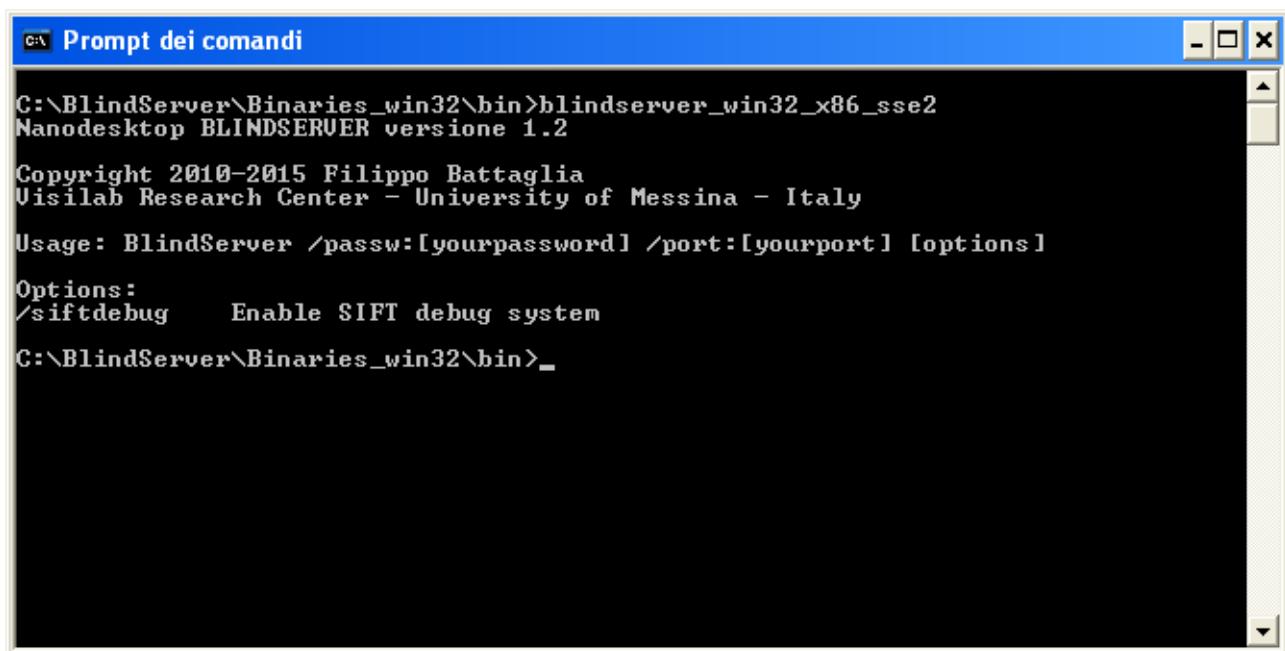
Type:

```
cd C:\BLINDSERVER\Binaries_Win32\bin
```

And after, start the BlindServer program:

```
blindserver_win32_x86
```

(or `blindserver_win32_x86_sse`, or `blindserver_win32_sse2`, if you want to use SSE/SSE2 acceleration).



## Step 5: Define port and password

As you can see in the previous paragraph, BlindServer is a CLI [<sup>9</sup>] program. It requires only two parameters: *the port number*, a number assigned by the operating system that allows the communication between the PC and the mobile program, and the *password*, a string that allows the authentication to the server, avoiding unauthorized uses.

If BlindServer is connected to internet, and if it is visible on the network (suppose that you are using the program from a distant location using a public WI-FI network), in theory other unauthorized PSP clients could access to your server.

For this reason the mobile client, under PSP, **must authenticate itself to the BlindServer** before using it.

You have to choice a password, and you have to store it into the BlindServer and into your BlindClient copy on PSP: **the correspondence between the two strings ensures that a specific client can connect only to a specific server**.

So, now decide the password and the port that you want to use. For example, you can type

```
c:\blindserver\binaries_win32\bin\blindserver_win32_x86_sse2 /passw:FILIPPO9872 /port:44075
```

To use BlindServer for SSE2 with password **FILIPPO9872** and port **44075**.

### NOTE 1:

*Be careful to this: the BlindClient communication packets can be stopped by your Windows firewall. So, you must configure your firewall for allowing the passage of the packets through the port 44075 (or the other port that you have choosen). See the documentation of your firewall for further informations about this.*

### NOTE 2:

If you have choosen a very long password, it can be frustrating to write the entire CLI string each time you want to start BlindServer.

Fortunately, you can avoid to do this. It is sufficient to write the complete string in a file batch, as BlindServer.Bat. So, if you want to start BlindServer, you can only type

### BlindServer.Bat

and BlindServer.Bat will recall the correct file .exe with the stored password and port. An example copy of BlindServer.Bat is already prepared in your **BlindServer\Binaries\_win32\bin** folder.

You can use a text editor (like NotePad or TextPad) and change the BlindServer.Bat file, in way of setting the password and the port that you want.

## Step 6: Configure the client (network settings)

Ok, now you have to configure the client.

The first thing to do is to configure a network connection using the PSP dashboard. You must create a network profile in **infrastructure mode**. Further informations about this can be obtained by Sony PSP documentation: we can only report here some screenshots about the configuration.



Create a network profile in *infrastructure mode*.

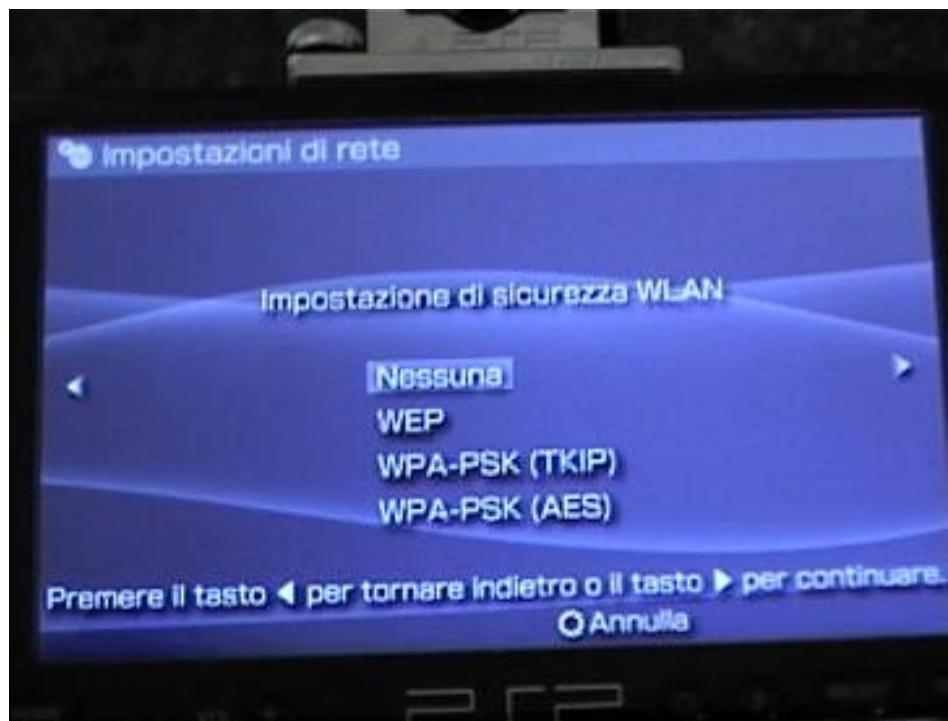




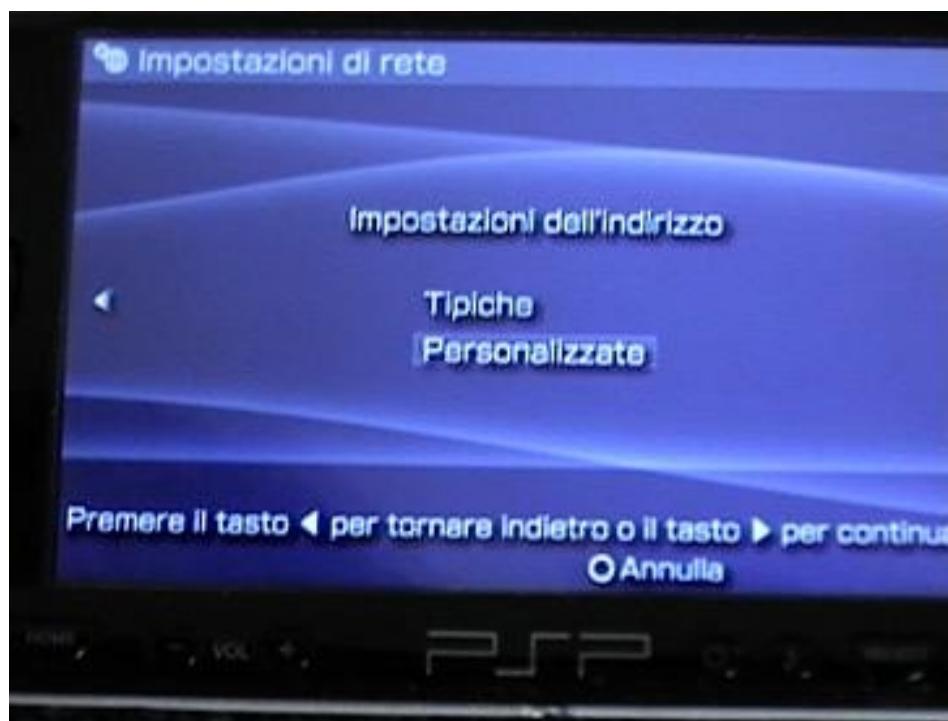
Set the correct SSID code for your Wi-Fi network



Set the correct security protocol for your WI-FI network (none, WEP, or WPA)



Set the correct address settings:



Choose the correct settings for the IP address (in my example I use **manual settings**, but you can choose the option that you prefer...)



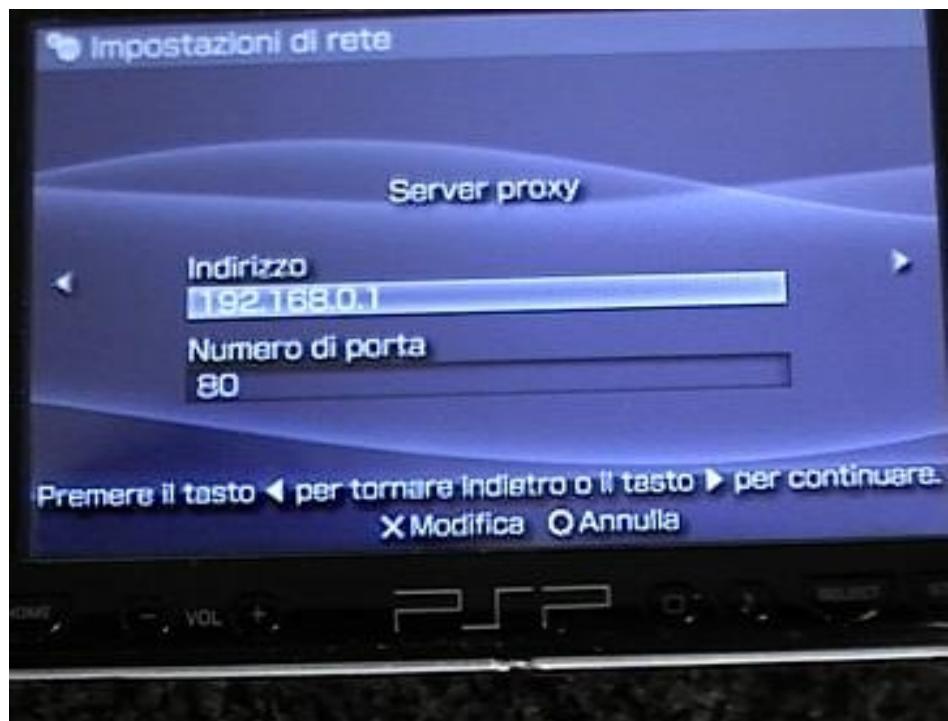
This is the mask for the network IP settings: the user must enter the correct data:



If you want to use the internet connection (for example, for **autoupdate feature**), set here the settings of your proxy server.



For example, I use a proxy server at 192.168.0.1 port 80.

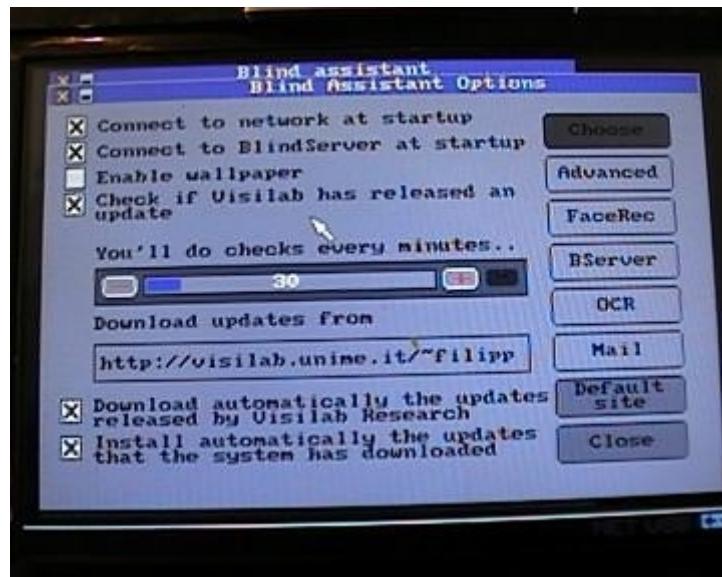


Now, complete the process of creation of the PSP network profile and test it using the function provided by your PSP dashboard. If it is all ok, you can start BlindServer.

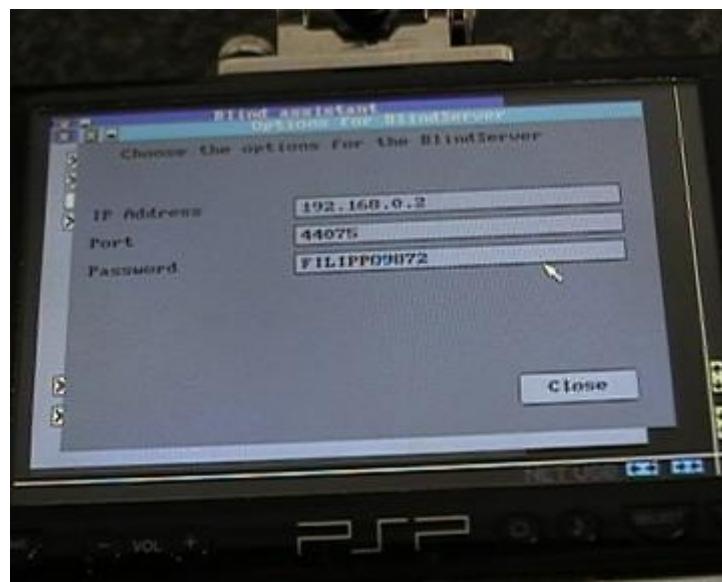
## Step 7: Configure the client (Blind Assistant)

Ok, we have configured the dashboard of the PSP for accessing to the network. Now, we have to configure Blind Assistant program in way to connect to the PC server.

Start Blind Assistant, and choose **Options** in the **Misc** menu: a window will appear:



Press the **BServer** button.



Here, you have to enter some informations about the address of your server in the home network.

The first textbox contains the IP address of the PC where BlindServer runs.

The second and the third textbox contains the port where BlindServer is listening and the password for accessing into it.

## Step 8: Establish a connection between BlindServer and the client

Ok, now you are ready to establish a connection between BlindServer x86 and Blind Assistant PSP.  
**The procedure is used only for testing the connection.**

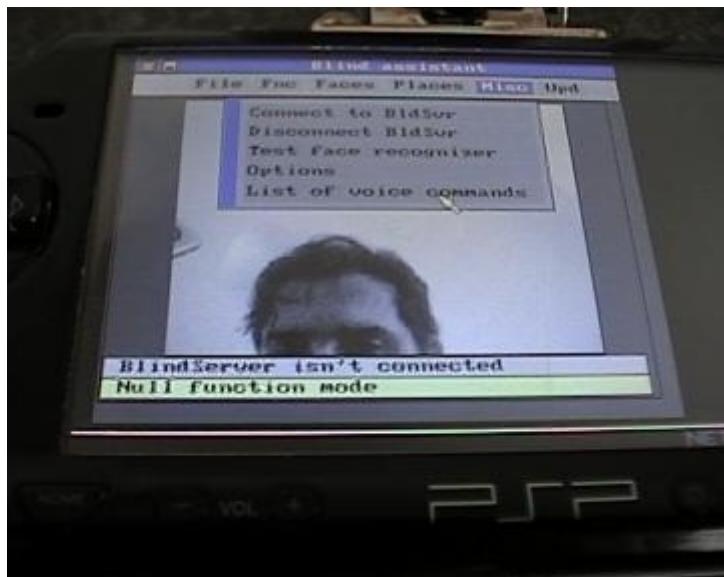
See this photo:



As you can see, the status bar is signalling that **BlindServer is not connected**. In this situation, *place recognition and Tesseract OCR method* aren't available.

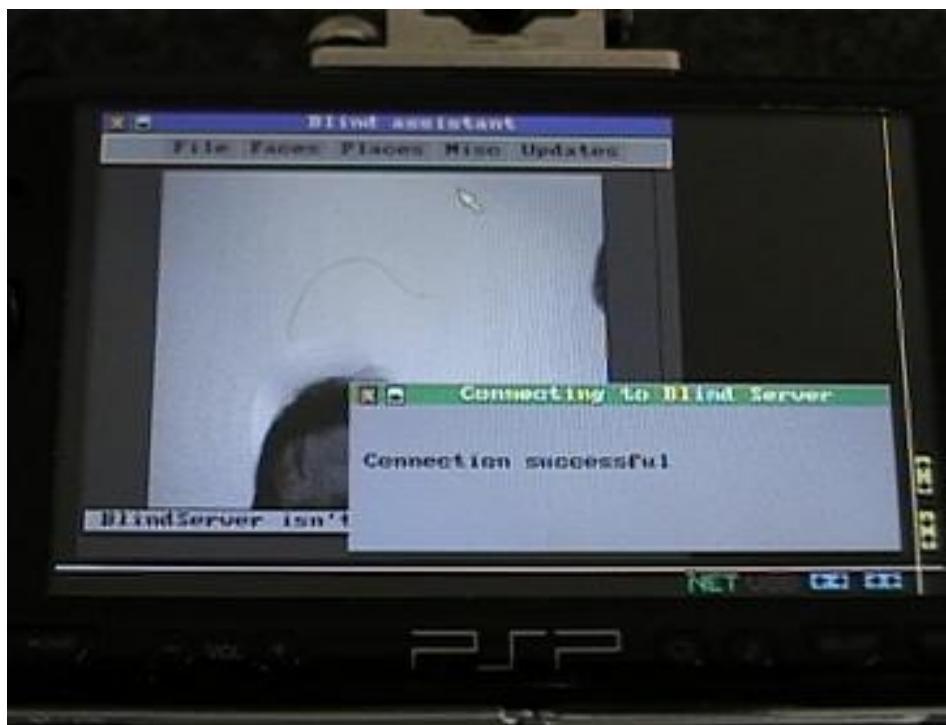
If you want to use one of these services, you have to connect to BlindServer. First, let's see the procedure for the *manual connection*. Ensure that BlindServer is running on a x86 PC and that wi-fi network is operative.

Go to the **Misc** menu and choose **Connect to BlindServer** option:

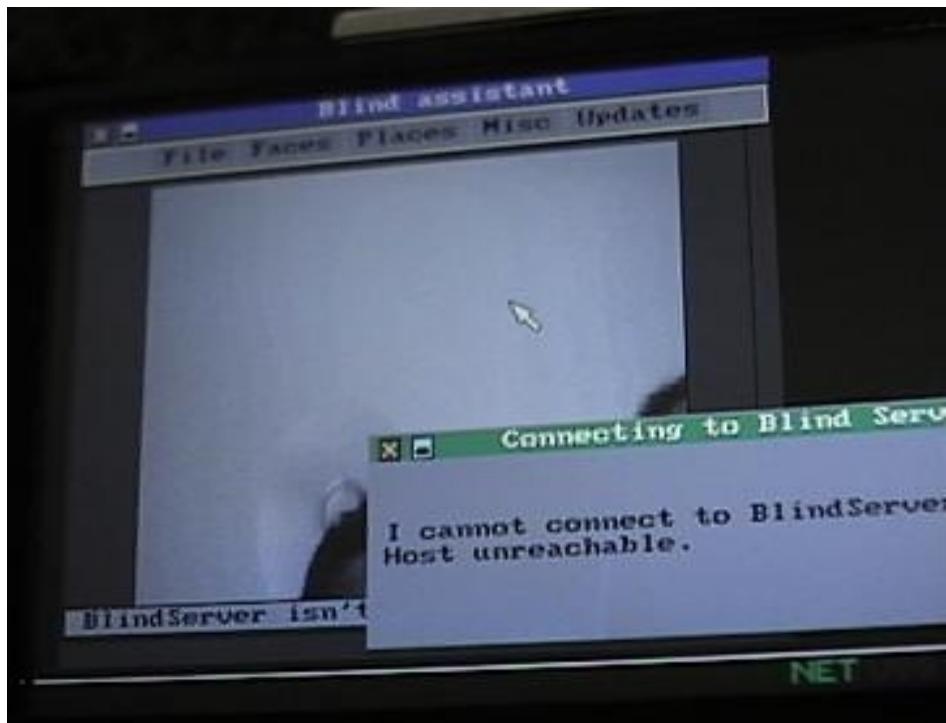


The system shall try to connect to BlindServer, using wi-fi network.

In this photo, the system has successfully established a connection with the server:



In this photo, instead, the system hasn't been able to connect to the BlindServer (in this case, the trouble is that the x86 server isn't executing the BlindServer program or that the network adapter is off or badly configured):



When you want to stop the connection between Blind Assistant and BlindServer, go into the **Misc** menu and choose the **Disconnect BlindServer** item.

## What can I do if the connection doesn't work ?

If the connection between BlindServer and BlindClient doesn't work, you can check these elements:

- In your network, there must be **at least a system (a router or a PC) that is executing the access point functions**, since you are using PSP network drivers in infrastructure mode. If you have only a personal computer with a wi-fi adapter and your console, look for some particular software that is able to emulate the functionalities of an access point via software. A program like this is made available, in some cases, by the producer of the wireless adapter. In other cases, it can be bought by some specialized vendors separately.
- Check if BlindServer is running at the moment of the client connection
- Check your IP addresses. Both server and client must have separated IP. The IP must be *compatible*, for example 192.168.0.X for server and 192.168.0.Y for client
- Check the password and the port: they must be the same in the client options dialog-box and in the server command line.

## Step 9: Switch to *position recognizer mode*

Ok, now we've learned how to connect Blind Assistant and BlindServer through a wi-fi network.

The moment is arrived: we must begin to use BlindServer. As first operation we'll try the *place recognition* function.

BlindServer works in the following way: the software receives the images from the mobile device (i.e by the PSP). The image is received by the server using the wi-fi network, and after it is uncompressed.

Now, BlindServer determinates the *SIFT features*, a set of mathematical vectors that can be associated to interesting details of a gray-tones image.

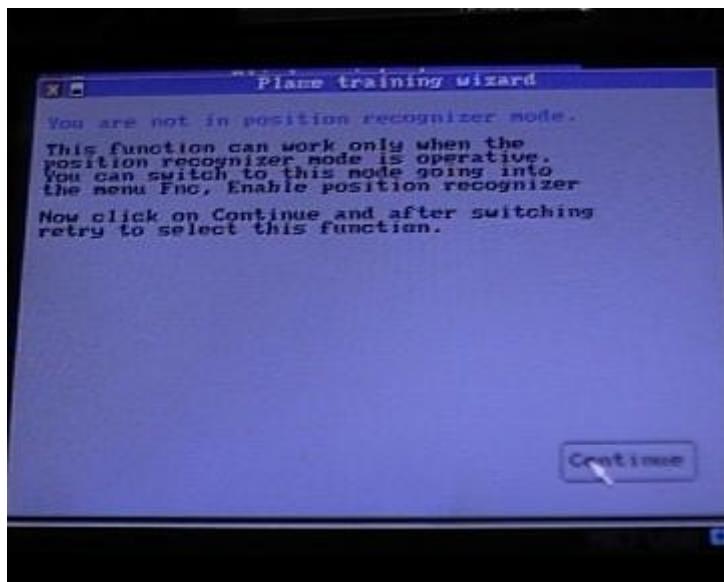
After, it compares the features of the received image with an internal database of the different rooms of the blind's home. Finally, it decides in which room the blind is, in base to a criterium of maximum similarity.

The response is retransmitted to the client via wi-fi and the PSP says to the blind where he is.

Before using the program, the **personal that assists the blind must train the software**, storing into the server the images of all rooms that must be recognized.

The training procedure is performed through a wizard that is run on the mobile device. However, starting from the CFW0006 core, the access to the training procedure has been modified, because of the new memory architecture. So, before starting the training procedure, you **must switch BlindAssistant into the *position recognizer mode*.**

If you try to enable the room training wizard, before enabling the position recognizer mode, you'd obtain the following error message:



So, the first thing to do is to switch the system in the position recognizer mode.

You can do this in three ways (see chapter 3). The blind can use the **manual switching** (pressing the right/left throttle buttons at the back of the console until the required function is enabled), or the **voice switching** (pressing the SQUARE button, in way that Blind Assistant goes into *listening mode*, and saying the command **Enable Position Recognizer**).

The personal that assists the blind (and that, we presume, will execute the training procedure) can also select manually the **Position Recognizer Mode**, using the application GUI.

Simply go into the menu **Fnc**, and select the item **Enable Position Recognizer**:



The system shall switch to position recognizer mode. Blind Assistant will say this with a voice message "Position recognizer system enabled".

### Step 10: Start the training procedure

Well, start BlindServer on x86 computer (see page 28). Now, go to the **Places** menu, and choose **Place 1**.



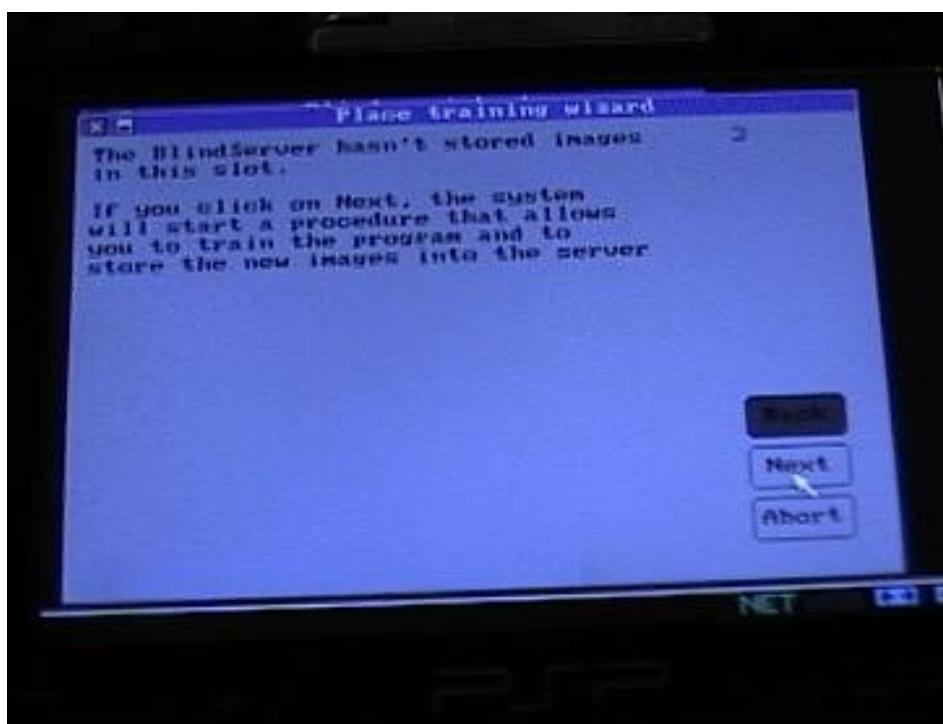
The PSP sends a *message packet* to BlindServer, using a particular protocol. Nanodesktop will start a wizard that allows you to train BlindServer by your PSP.

The system will start the **place training wizard**. Note that Blind Assistant will try automatically to establish a wireless connection with the BlindServer. If the connection fails, the wizard is interrupted and an error message is shown on the screen.

If, instead, the connection is successful, you'll see this message on the display and the wizard procedure can continue:

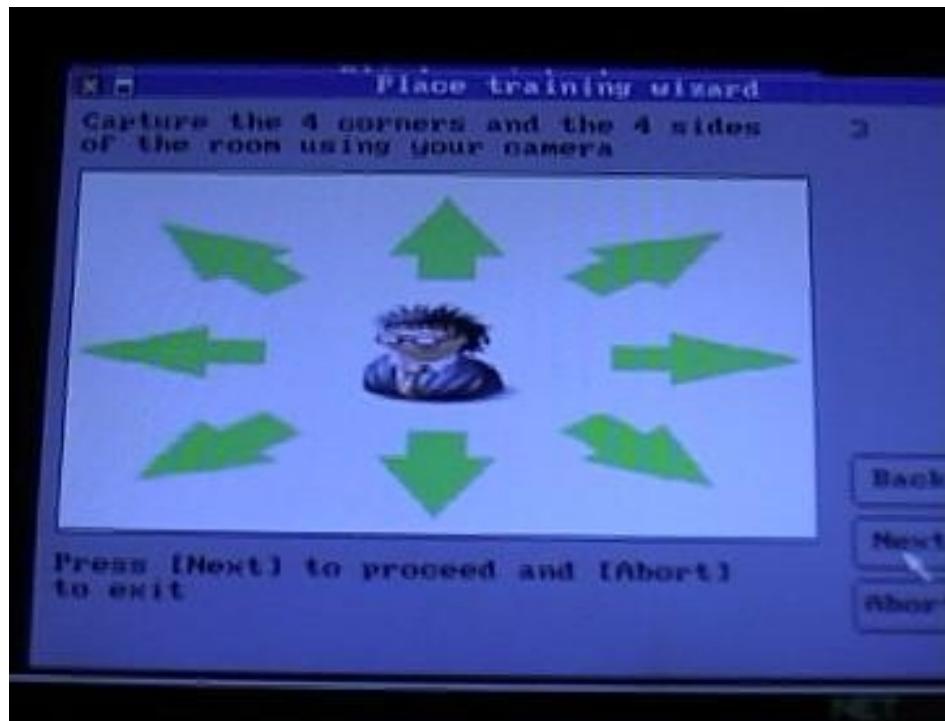


Click on the **Next** button.

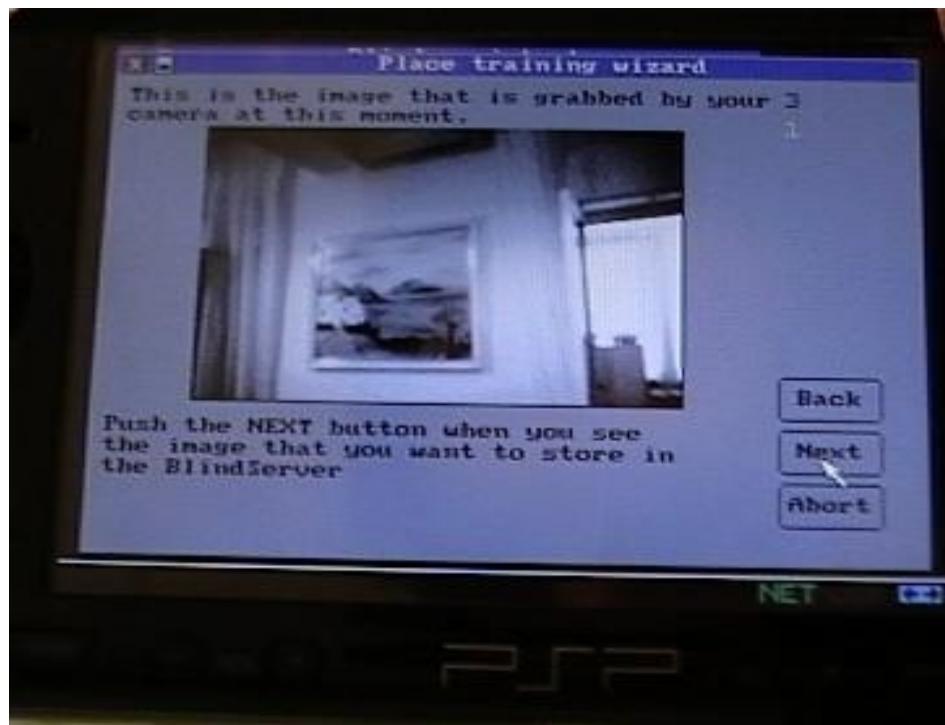


The server is automatically questioned by Blind Assistant: here it is saying that there is no *set of samples* that is stored in the slot 0 of the server. Well, click on **Next** and you'll begin the training.

For each room of your home, BlindServer requires a set of 8 picture. They must be grabbed from any

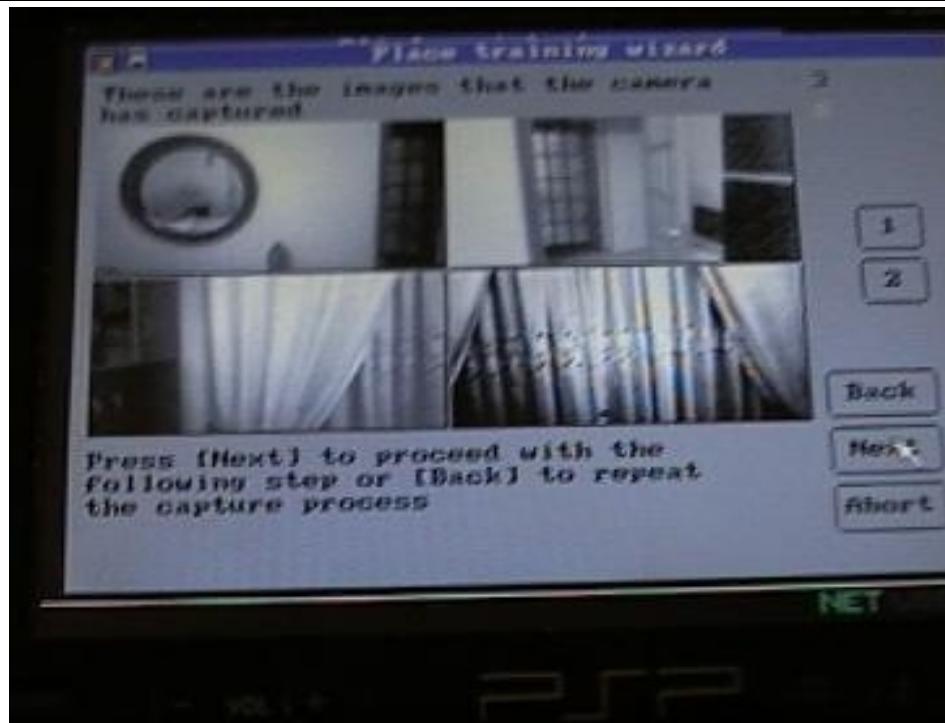


corner and from any side of the room (as the figure shows).



Click on the **Next** button. The image will be stored in a temporary buffer on the PSP. Point your PSP to another corner of the room and click again on the **Next** button. The entire operation must be repeated for *8 times*.

When you'll finish, the system shall show a summary of the room, like this:



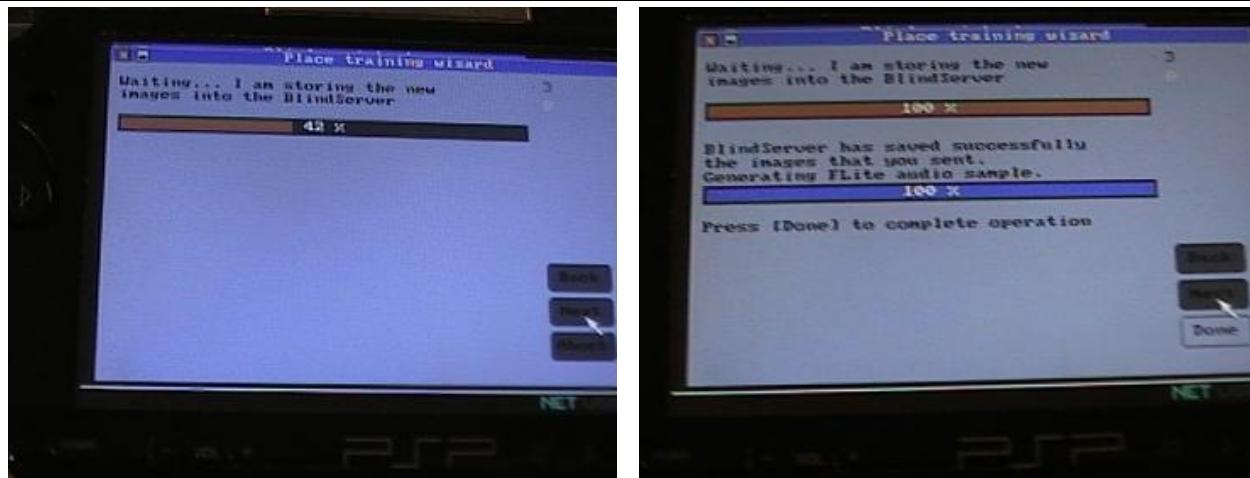
Click on **Next**.

Now, Blind Assistant requires a further information: you have to indicate the name of the room.

Click on the textbox (the rectangle in gray), and type the name using the virtual keyboard. When you have finished, click on **ENTER** or on **OK** button.



Using the button **Test my voice**, you can test how the name of the room will be spelled by the software. Click on **Next**.



The wizard is nearly complete.

Now, Blind Assistant PSP is sending (in a compressed format) the 8 images to the x86 BlindServer using the wi-fi connection. After, it synthesizes the name of the room in PCM format and stores it into the memory stick (for cache). Click on **Done**.

**Now, the informations about the given room/environment are stored in the choosen slot of the Blindserver.**

This operation must be done again for each room of your home: the BlindServer requires a training operation for each place.

## Step 11: Use the position recognizer

After that training is completed, the blind can use Blind Assistant to recognize the room where he is.

**Everytime, if that is needed, the blind must switch the system in *Position recognizer mode*, using the voice control (press the SQUARE button and says **Enable position recognizer**) or using the manual switching (pressing the right/left throttle buttons at the back of the console).**

When the switching operation is completed (after few seconds), the client will inform the blind through a voice message (*Position recognizer system enabled*).

This is Blind Assistant in **Position recognizer mode**:



When the blind wants to know where he is, **he can simply point the camera to a side of the room and press the TRIANGLE button**. The system says "Wait" and the status bar reports the message "*Contacting BlindServer*". In that moment, the system tries to establish a connection to the BlindServer: it compresses the grabbed image and sends it to the x86 computer via wi-fi [<sup>10</sup>]



The image is analyzed by the server. This is the log of the server activity:

```
C:\WINDOWS\system32\cmd.exe
Slot 0 Sample 1: SIFT features found 0
Slot 0 Sample 2: SIFT features found 0
Slot 0 Sample 3: SIFT features found 0
Slot 0 Sample 4: SIFT features found 0
Slot 0 Sample 5: SIFT features found 0
Slot 0 Sample 6: SIFT features found 0
Slot 0 Sample 7: SIFT features found 1
Slot 0 Sample 8: SIFT features found 0
Slot 0 Sample 9: SIFT features found 0
Slot 1 Sample 1: SIFT features found 0
Slot 1 Sample 2: SIFT features found 4
Slot 1 Sample 3: SIFT features found 0
Slot 1 Sample 4: SIFT features found 2
Slot 1 Sample 5: SIFT features found 0
Slot 1 Sample 6: SIFT features found 0
Slot 1 Sample 7: SIFT features found 0
Slot 1 Sample 8: SIFT features found 0
Slot 1 Sample 9: SIFT features found 0
Slot 2 Sample 1: SIFT features found 0
Slot 2 Sample 2: SIFT features found 1
Slot 2 Sample 3: SIFT features found 0
Slot 2 Sample 4: SIFT features found 0
Slot 2 Sample 5: SIFT features found 0
Slot 2 Sample 6: SIFT features found 0
Slot 2 Sample 7: SIFT features found 0
Slot 2 Sample 8: SIFT features found 0
Slot 2 Sample 9: SIFT features found 0
Slot 3 Sample 1: SIFT features found 0
Slot 3 Sample 2: SIFT features found 0
Slot 3 Sample 3: SIFT features found 0
Slot 3 Sample 4: SIFT features found 0
Slot 3 Sample 5: SIFT features found 0
Slot 3 Sample 6: SIFT features found 0
Slot 3 Sample 7: SIFT features found 0
Slot 3 Sample 8: SIFT features found 0
Slot 3 Sample 9: SIFT features found 0
Slot 4 Sample 1: SIFT features found 0
Slot 4 Sample 2: SIFT features found 0
Slot 4 Sample 3: SIFT features found 0
Slot 4 Sample 4: SIFT features found 0
Slot 4 Sample 5: SIFT features found 0
Slot 4 Sample 6: SIFT features found 0
Slot 4 Sample 7: SIFT features found 0
Slot 4 Sample 8: SIFT features found 0
Slot 4 Sample 9: SIFT features found 0
Slot 5 Sample 1: SIFT features found 0
Slot 5 Sample 2: SIFT features found 0
Slot 5 Sample 3: SIFT features found 0
Slot 5 Sample 4: SIFT features found 0
Slot 5 Sample 5: SIFT features found 0
Slot 5 Sample 6: SIFT features found 0
Slot 5 Sample 7: SIFT features found 0
Slot 5 Sample 8: SIFT features found 0
Slot 5 Sample 9: SIFT features found 0
Slot 6 Sample 1: SIFT features found 0
Slot 6 Sample 2: SIFT features found 0
Slot 6 Sample 3: SIFT features found 0
Slot 6 Sample 4: SIFT features found 0
Slot 6 Sample 5: SIFT features found 0
Slot 6 Sample 6: SIFT features found 0
Slot 6 Sample 7: SIFT features found 0
Slot 6 Sample 8: SIFT features found 0
Slot 6 Sample 9: SIFT features found 0
Slot 7 Sample 1: SIFT features found 0
Slot 7 Sample 2: SIFT features found 0
Slot 7 Sample 3: SIFT features found 0
Slot 7 Sample 4: SIFT features found 0
Slot 7 Sample 5: SIFT features found 0
Slot 7 Sample 6: SIFT features found 0
Slot 7 Sample 7: SIFT features found 0
Slot 7 Sample 8: SIFT features found 0
Slot 7 Sample 9: SIFT features found 0
Slot 8 Sample 1: SIFT features found 0
Slot 8 Sample 2: SIFT features found 0
Slot 8 Sample 3: SIFT features found 0
Slot 8 Sample 4: SIFT features found 0
Slot 8 Sample 5: SIFT features found 0
Slot 8 Sample 6: SIFT features found 0
Slot 8 Sample 7: SIFT features found 0
Slot 8 Sample 8: SIFT features found 0
Slot 8 Sample 9: SIFT features found 0
Slot 9 Sample 1: SIFT features found 0
Slot 9 Sample 2: SIFT features found 0
Slot 9 Sample 3: SIFT features found 0
Slot 9 Sample 4: SIFT features found 0
Slot 9 Sample 5: SIFT features found 0
Slot 9 Sample 6: SIFT features found 0
Slot 9 Sample 7: SIFT features found 0
Slot 9 Sample 8: SIFT features found 0
Slot 9 Sample 9: SIFT features found 0
Analysis is completed.
```

If the *matching SIFT features* are greater than a stated threshold, the server shall report that the blind man is in a known room. The name of the room is transmitted via wi-fi to the client.

At this point, Blind Assistant has only to communicate to the user the response that it has received by the server: the blind shall listen a voice advice "**You are in ...**" followed by the name of the room.

If you obtain, instead, a message like "**I don't know where you are**", this means that the obtained SIFT features were not significative (under the threshold) and so the results of the test have been ignored by the software.

<sup>10</sup> **IMPORTANT NOTE:** Starting from the core version CFW0006, Blind Assistant (client) shall try automatically to connect to BlindServer every time that the program goes into the position recognizer mode. Moreover, if this initial attempt to connection fails, the client will retry to connect everytime the blind presses the TRIANGLE button. In this way, the connection procedure is totally automatic and the blind has not to worry about it.

## Chapter 6

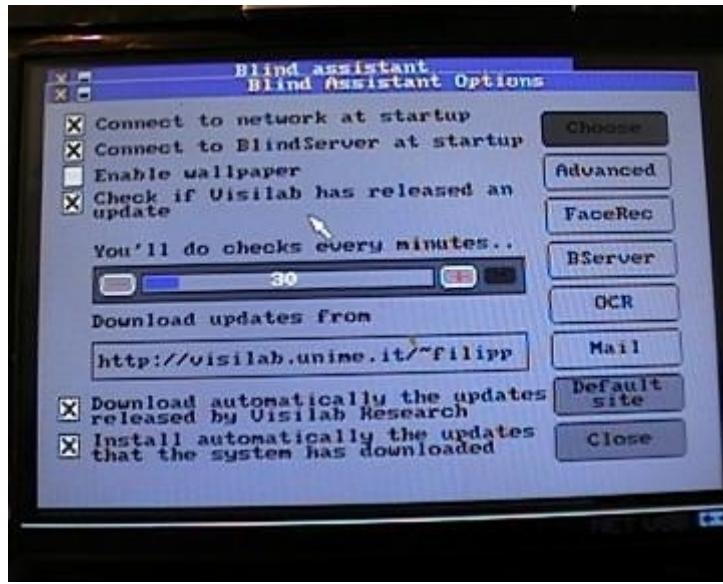
# BlindServer connection at startup

In Blind Assistant is integrated an option making that the program connects automatically to the server at any startup.

When this option is enabled, the PSP will try to connect to BlindServer at startup without any manual operation. Go to **Misc** menu and choose **Options**:



The following window will appear



Enable the options:

- **Connect to network at startup**
- **Connect to BlindServer at Startup**

and click on **Close** button. It's all done.

If you restart the system, you will see a window as the following:



The system is connected now. If the connection at startup should fail, the system will retry to connect to Blindserver automatically when the blind switches to *position recognizer* or *optical char recognizer mode*.

## Chapter 7

# OCR

OCR stands for *Optical Char Recognition*: it is a software that is able to recognize the chars in an image, and to transform them in a text. An OCR can help the blind to read a text or a label, if it is used in combination with a software of voice synthesis. This version of Blind Assistant integrates the support for *OCR recognition*. Now, we'll see how to use this function.

The OCR integrated in Blind Assistant has been designed for reading a string in a label, or a small text in a paper. It can use two different strategies, called *OCR methods*. They are:

- **Tesseract via BlindServer**
- **ndTesseract (internal)**

When you use **Tesseract via BlindServer**, the image grabbed by the camera (at a resolution of 1024x768 pixel) is compressed and sent, via wireless, to the server. In BlindServer 1.2 is integrated a special version of Tesseract 2.04, the OCR engine developed originally by HP and now hosted by Google Foundation (<http://code.google.com/p/tesseract-ocr/>). Tesseract engine has been chosen because it is the best open-source OCR available at the moment. The response generated by the OCR is, in a successive moment, re-sent to the mobile device. This will provide to read the content of the response to the blind user, through the ndFlite voice synthetizer.

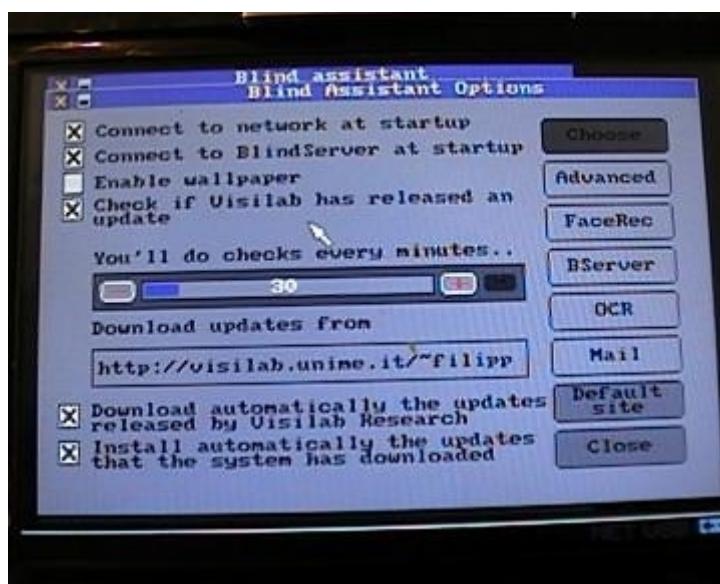
This method is very fast, but it has the drawback that it requires the presence of a working wireless connection and of a personal computer.

The OCR method **ndTesseract (internal)** [11] performs the entire scanning process on the mobile console. As the processor of the PSP is a CPU 333 Mhz, we had to developed a **new, special version, of Tesseract** that is able to run also in a MIPS processor for embedded systems. This OCR engine is called **ndTesseract** (from *Tesseract for Nanodesktop*). This special software, is able to scan an image of 1024x768 within 2 minutes, and using only the local resources of the PSP.

So, the first thing to do is to configure the OCR system, in way to choose the wished OCR method.

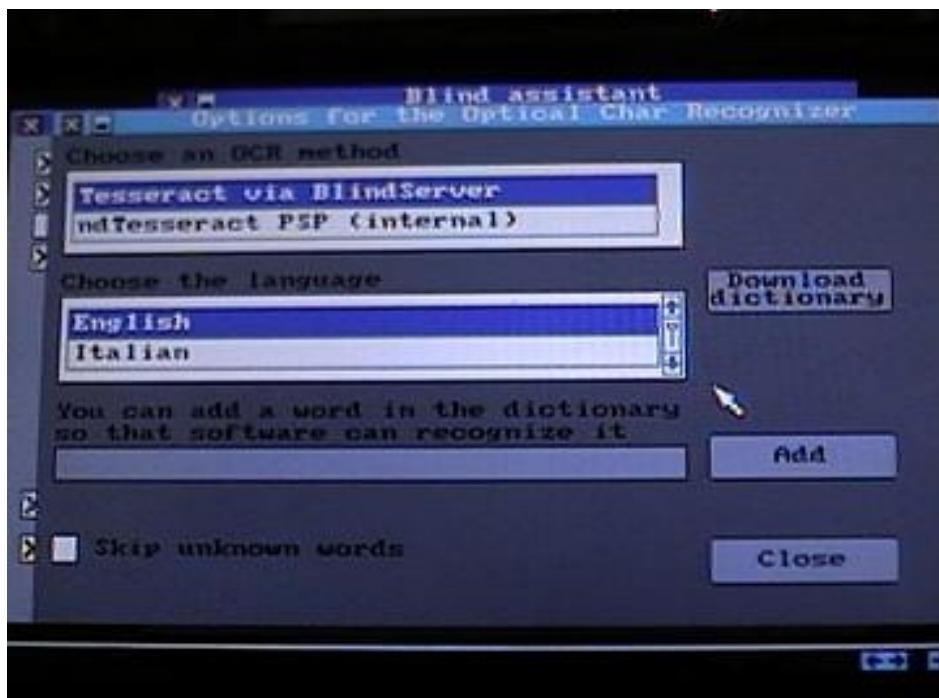
## 1. Configure the OCR

Go into the **Misc** menu and choose **Options**. The following window shall appear:



11 Not available for PSP-FAT (PSP-1000) users

Click on the **OCR** button. A configuration window shall appear on the screen:



In the first listbox, you can choose the correct OCR method.

- **Tesseract via BlindServer**
- **ndTesseract PSP (internal)** [<sup>12</sup>]

In the second listbox, the user can choose the language that is used for correcting the recognized text. In fact, PSP executes an algorithm (after scanning) developed in Visilab Research Lab that is able to correct the wrong words. It needs to know the language in which the text is written, since it uses a dictionary for the correction.

So, you must choose the language. Actually, the supported languages are English, Italian, French, Spanish, German and Dutch.

The option **Skip unknown words** allows to skip the words that are not present in the dictionary. This can give a better comprehension of the sense of the text spoken by the console.

In the dialog box there are some other options and other elements, but we'll return on those in a second time. When the options have been configured, click on **Close**. The system will save the new settings.

<sup>12</sup> This option is not available for PSP-1000 (PSP-SLIM) users. This version of the OCR engine has replaced the old engine ndOcrad. ndTesseract provides performance very large respect than Ocrad.

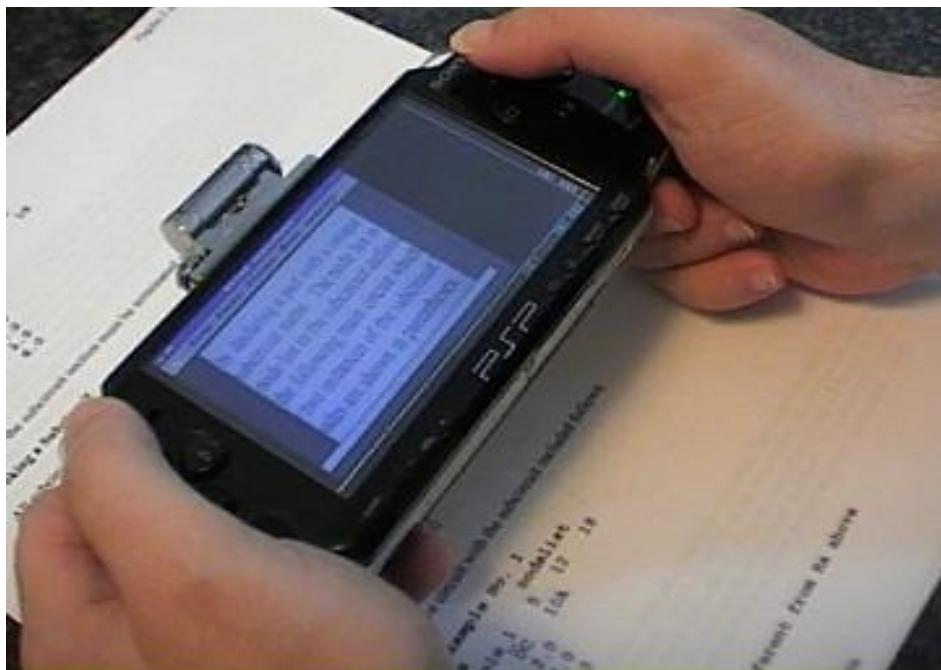
## 2.Use of the OCR (Tesseract via BlindServer)

In this paragraph we'll assume that you are using the OCR method **Tesseract via BlindServer**.

**Switch Blind Assistant to *optical char recognition mode*** using the manual procedure (see chapter 3) or the voice control system (press the SQUARE button and say "*Enable optical char recognizer*").

When you enter in face recognizer mode, the system will attempt automatically to establish a connection with the BlindServer.

Now, put the PSP on a paper, like in the picture that you can see [<sup>13</sup>].



When you want to start the recognition process, **press the TRIANGLE button**: the system will send the image of the paper to BlindServer for the analysis [<sup>14</sup>]

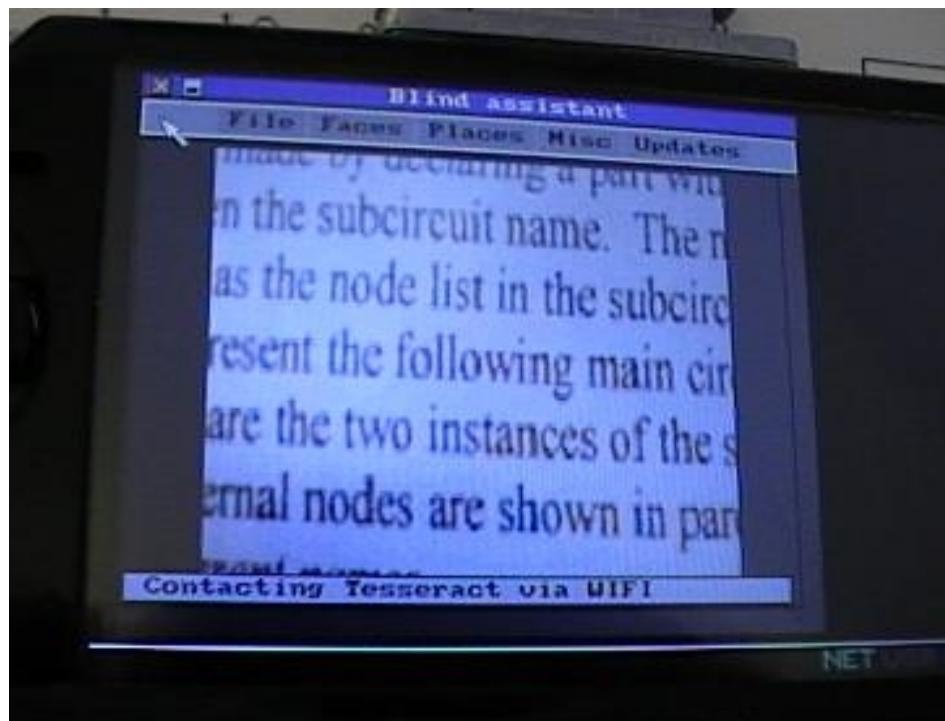
The software performs the following steps:

- a) adaptive thresholding: it removes noise by the image;
- b) resizing of the image and compression without losses;
- c) transmission to BlindServer for Tesseract processing;
- d) retransmission to the client of the resulting text;
- e) correction of the wrong words using a dictionary and an algorithm of maximum similarity;
- f) synthesis in real time of the text
- g) reproduction of the words that have been recognized

<sup>13</sup> It is important that the text is in focus. The PSP GoCam camera hasn't an autofocus function. However, it has a ring that the blind can rotate. The ring has two positions: the blind must change the position of the camera before using the OCR. This is a price to pay for using Blind Assistant in a cheap device like the PSP.

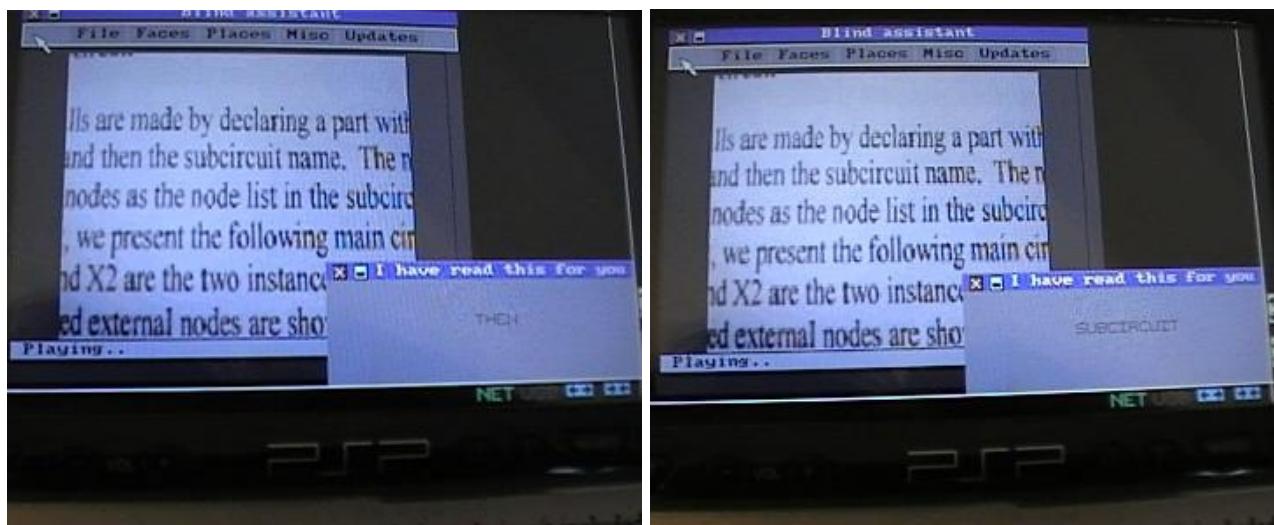
<sup>14</sup> If the server is not connected when the user presses the TRIANGLE button, BlindAssistant will retry to connect automatically. If the connection fails, the user will receive the voice message "Blindserver is not connected".

During each stage of the operations, you'll hear a short beep: it informs the blind that the system is in action. This is the result: you'll obtain this after few seconds:



(Here, the system has contacted Tesseract engine on BlindServer)

After receiving the response by the BlindServer, the system will start to spell the content of the text word-by-word.



(The phase of words spelling).

During the voice synthesis of the text, or during the generation of the words, the user can stop the process of spelling (for example, because he has still understood the sense of the message).

This can be done simple keeping pressed the TRIANGLE button. After few instants, the software will stop the recognition operations and it will return to the normal interface.

### 3.Use of the OCR (ndTesseract - internal)

In this paragraph we'll assume that you are using the OCR method **ndTesseract (internal)**.<sup>[15]</sup>

**Switch Blind Assistant to *optical char recognition mode*** using the manual procedure (see chapter 3) or the voice control system (press the SQUARE button and say "*Enable optical char recognizer*").

This OCR method doesn't need a wireless connection. As drawback, the recognition process can require almost 2 minutes.

Put the PSP on a paper, like in the picture that you can see<sup>[16]</sup>.



When you want to start the recognition process, **press the TRIANGLE button**: the camera will acquire the image and the system will begin the processing (a beep is heard by the blind).

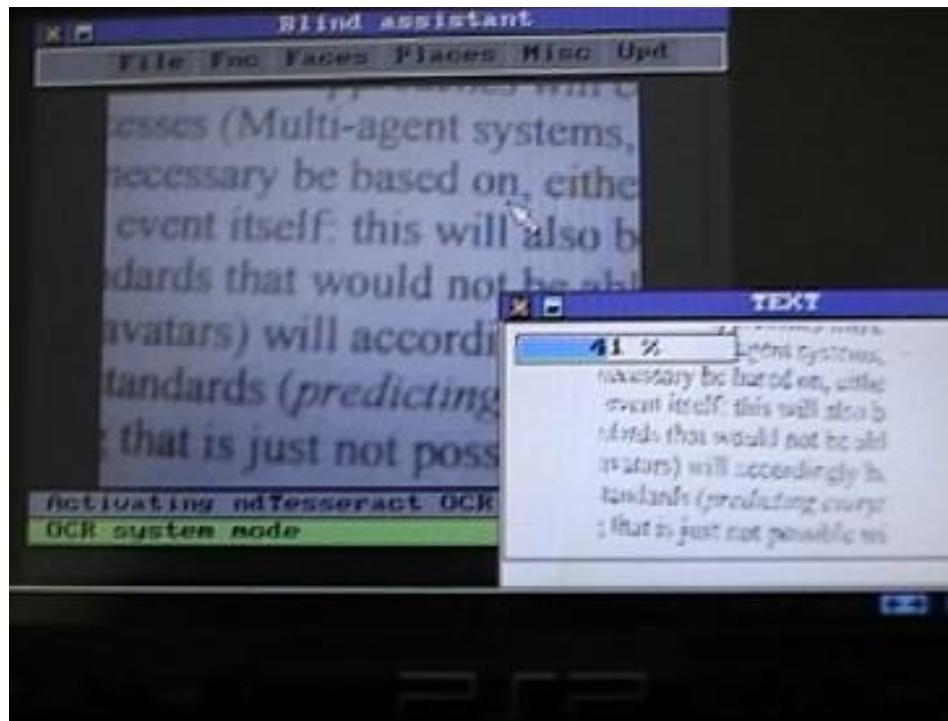
The software performs the following steps:

- a) adaptive thresholding: it removes noise by the image;
- b) erosion of the image: it divides the single letters in the image;
- c) resizing of the image: it makes lower the computation weight for MIPS processor
- d) execution of the *ndTesseract* algorithm on-board the MIPS processor;
- e) correction of the wrong words using a dictionary and an algorithm of maximum similarity;
- f) synthesis in real time of the text
- g) reproduction of the words that have been recognized

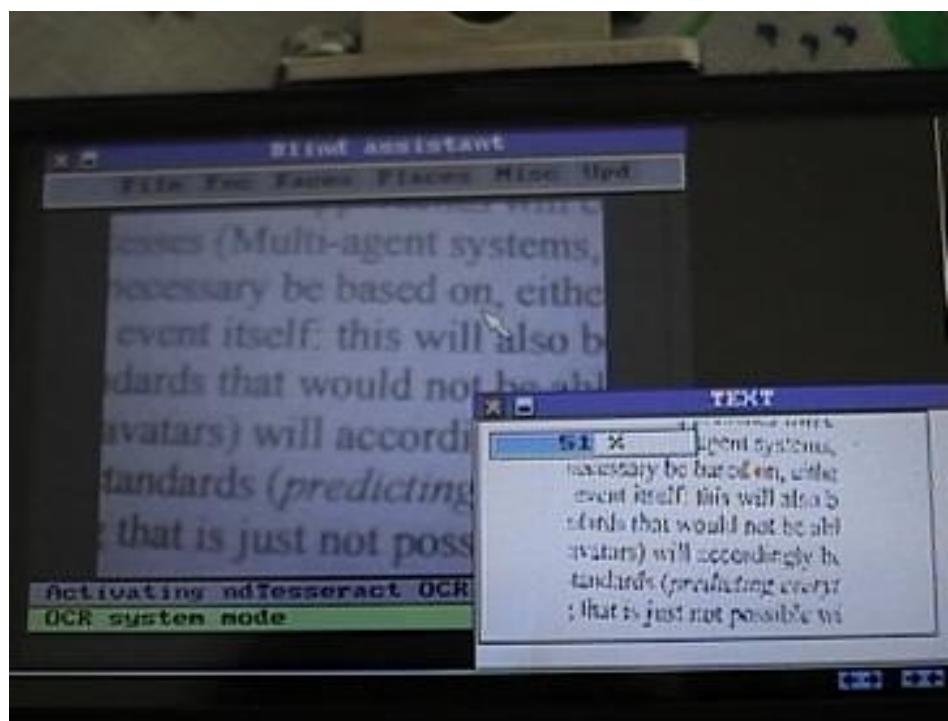
<sup>15</sup> This function is not available for PSP-1000 (PSP-FAT) users

<sup>16</sup> It is important that the text is in focus. The PSP GoCam camera hasn't an autofocus function. However, it has a ring that the blind can rotate. The ring has two positions: the blind must change the position of the camera before using the OCR. This is a price to pay for using Blind Assistant in a cheap device like the PSP.

As the recognition process can keep 2 minutes, a "voice progress bar" has been implemented for the blind.

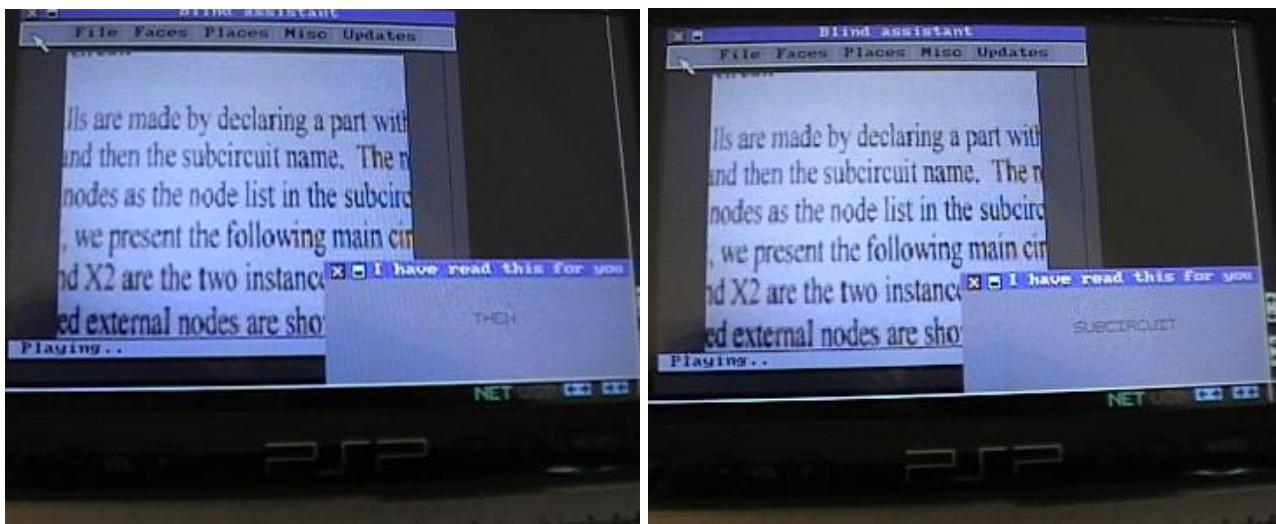


During the scanning process, the blind will hear the voice messages "*twenty per cent*", "*fourty per cent*", "*fifty per cent*", etc.



Anyway, the blind can stop in any moment the scanning process keeping pressed the TRIANGLE button for five seconds. In this case, a beep is emitted and the system will begin to synthesize only the part of the text that had been decoded until the moment of interruption.

After terminating the scanning process, the system will start to spell the content of the text word-by-word.



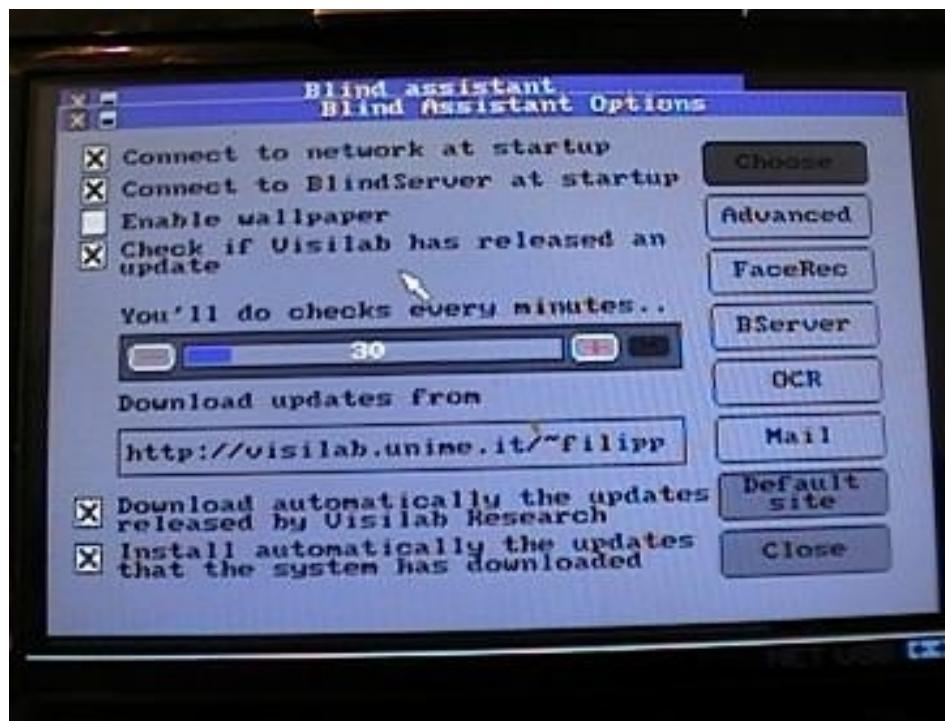
(The phase of words spelling).

Also in this case, the user can stop in any moment the process of spelling, if he has still understood the sense of the message, simply keeping pressed the TRIANGLE button.

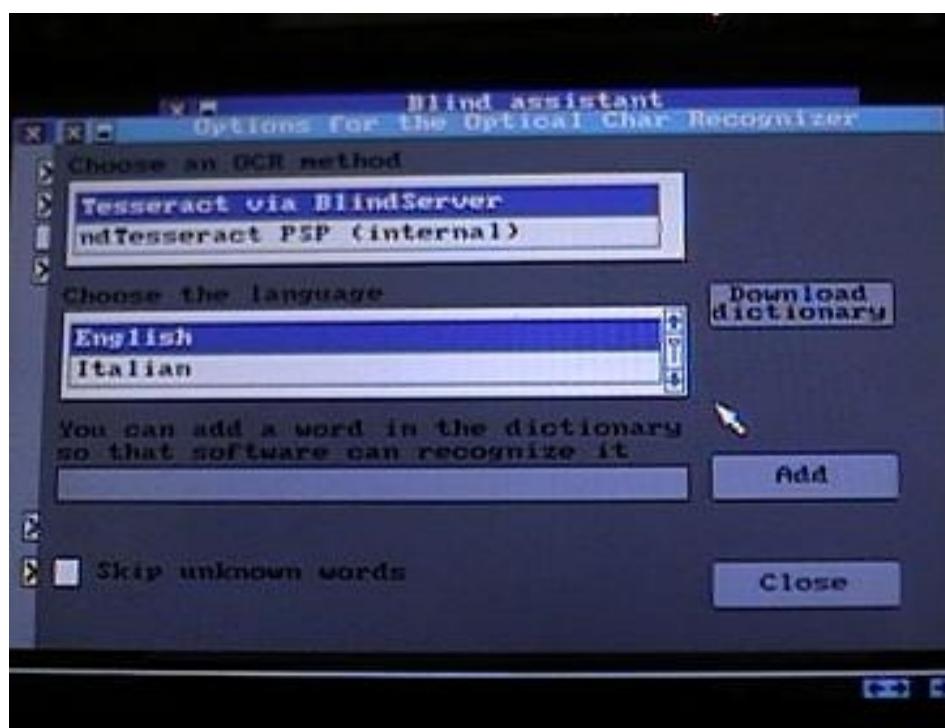
Terminating its job, the software will return to the normal interface.

## 4. More about OCR options

Now, we can return to the **OCR Options window**. In a previous paragraph, we've seen that it can be reached going into menu **Misc**, item **Options**:



and pressing the **OCR** button:



In the **OCR option window** there are some functions that we've ignored before. Now, we'll examine them in detail.

## 4.1 Add a new word to the dictionary

In some cases there is a word that is frequent in a text (for example, the name of the blind or of the company where the blind works). Since this word isn't present in the dictionary, Blind Assistant could try to correct it with a similar term. This could cause an error.

For this reason, Blind Assistant provides a function that allows the introduction of new known terms to a dictionary. Go to **Misc/Options/Ocr** window and click on the textbox called *You can add a new word*.



Virtual Keyboard can appear over the textbox, but in this case it is sufficient to drag its window in a position that is better (push the CROSS button over the title bar of the keyboard window and drag it in the desidered position).

Type the new word in the textbox and click on **ENTER** or **OK** button. After, click on the **Add** button. The system will add the new word to the dictionary and it also generate a new wave file.

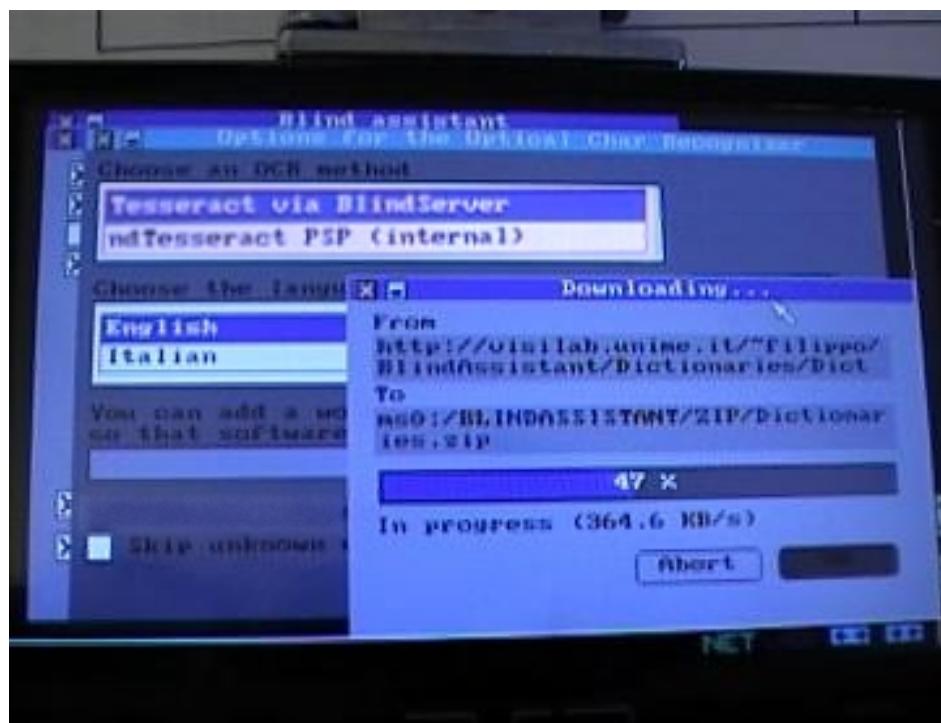
## 4.2 Update the set of the textual dictionaries

Blind Assistant OCR can work only if a dictionary is present. A set of dictionaries is provided with the software, but Visilab can update them periodically.

For this reason, Blind Assistant can download, through Internet, a new dictionary.  
Go to menu **Misc/Options/Ocr**.

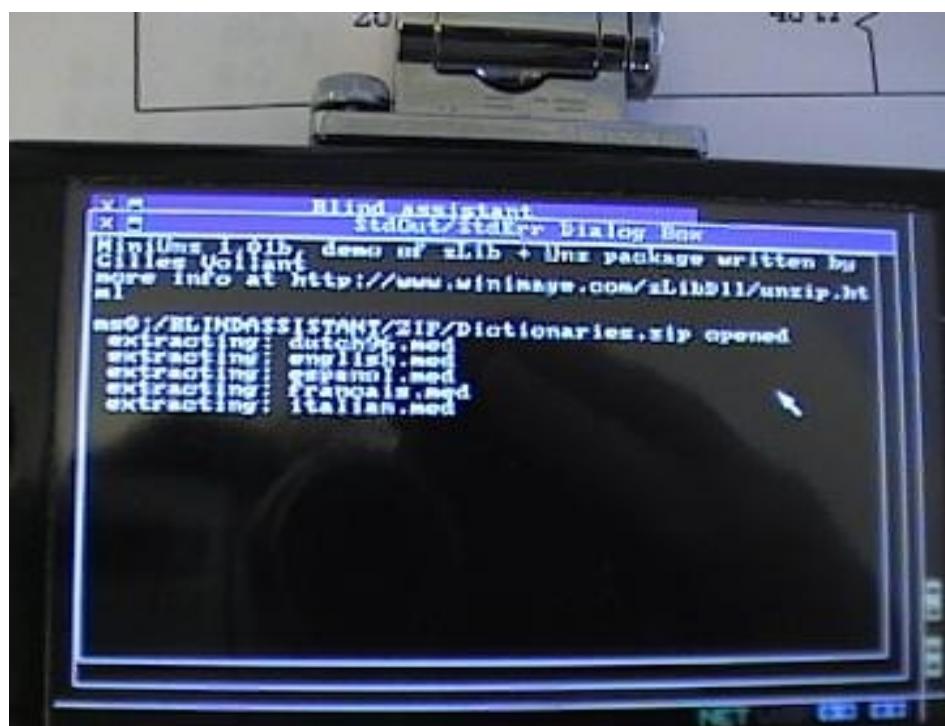


Choose the desidered language (in this photo, it is English) and click on **Download dictionary** button.



The system will connect automatically to the Visilab server and it will download the wished textual dictionary.

The files are compressed in a zip file, but Blind Assistant is able to uncompress that automatically into the memory stick. No other user operation is required.



## Chapter 9

# Color scanner

In some cases, the blind needs to recognize the color of a dress, or the color of an object. At beginning from Blind Assistant CFW0005, the program integrates a *color scanner*: this component allows to know the average color of the image that is grabbed in that moment by the camera.

The color scanner can be enabled using the manual switching system (see chapter 3) or using the voice control system: press the SQUARE button and say the command **Enable color scanner**. This shall switch the program into *color scanner mode*.



If you want to know what is the average color that is present in the image, **press the TRIANGLE button**: after few seconds, the system will say to you that "*The average color is...*".

## Chapter 10

# Mail reader

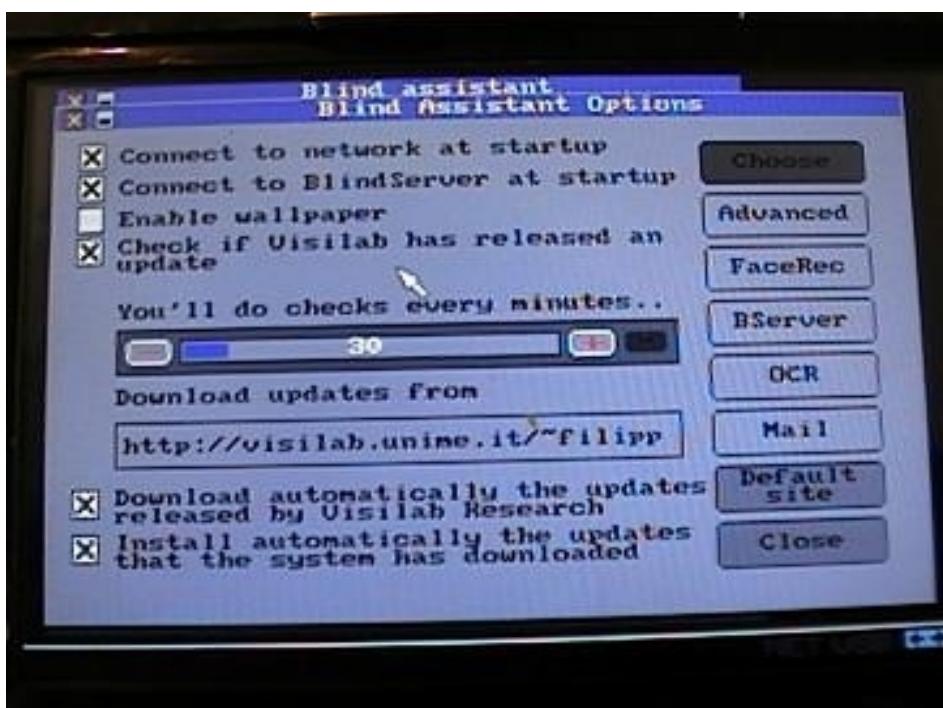
This version of Blind Assistant integrates a small mail client that allows to the blind to receive the mails from a POP account [17]. The messages are read through the wi-fi connection.

## Configuration

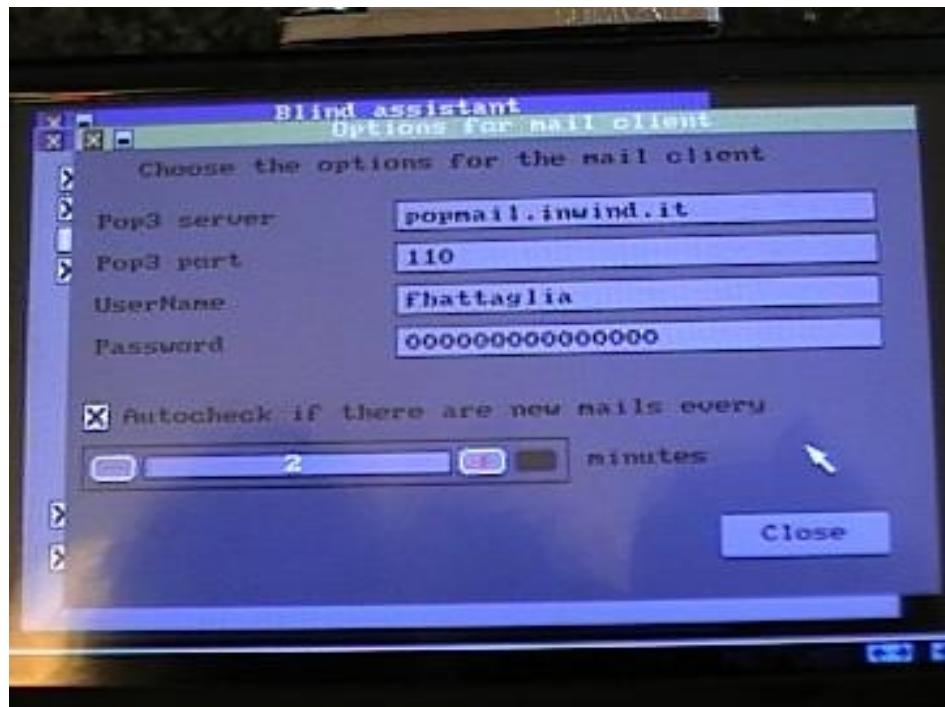
The first thing to do is to configure the POP3 client. Go to the *Misc* menu and choose *Options*:



Click on the **Mail** button:



This shall open the *mail configuration* dialog-box:



Using the text boxes, enter:

- the *address of the POP3 server* (be careful: the servers that use SSL for authentication are not supported. This limitation will be removed in the next versions of the program);
- the *username* and the *password* of the POP3 account;
- the POP3 port (usually it is 110).

Click on **Close** button.

## Mail autochecking

The blind can check in any moment if there are new mails in his account, switching the program into *mail reader mode*, and interrogating the POP3 server.

However, to interrupt the current operation for switching to mail reader mode and to discover that there are no new mails can often be frustrating. For this reason, starting from the version CFW0006, Blind Assistant implements a *mail autochecker*<sup>[18]</sup>.

It is a *daemon*, a program that runs always in background and that checks every n minutes if there are new mails in the POP3 server. If new mails are found, a voice message is emitted to inform the blind.

The service can be enabled or disabled in the **Options/Mail client** window.

It is always operative in background, except that during the ndTesseract scanning process (in this case it is disabled for security reason, as the scanning process is a very impegnative job for the PSP CPU).

---

<sup>18</sup> This service is not available for PSP-1000 (PSP-FAT) users

The check is performed every n minutes: the interval can be set in the **Options/Mail** window.

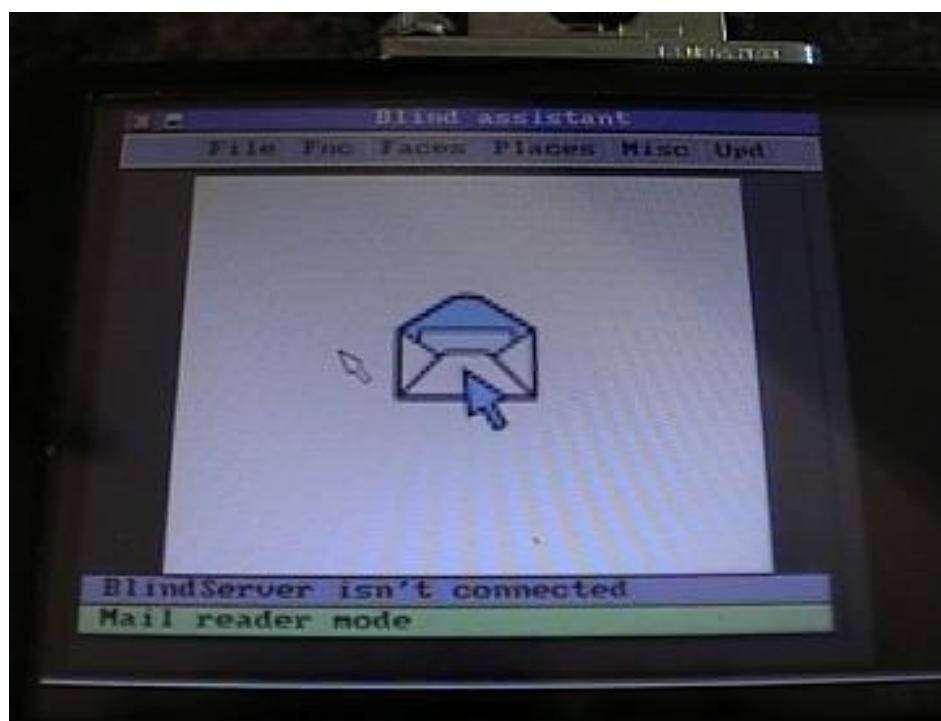
When new mails are found in the POP3 server, a small icon appears in the main interface and a voice message is emitted for the blind: "You have unread mails. Please switch to mail reader mode for accessing to them."



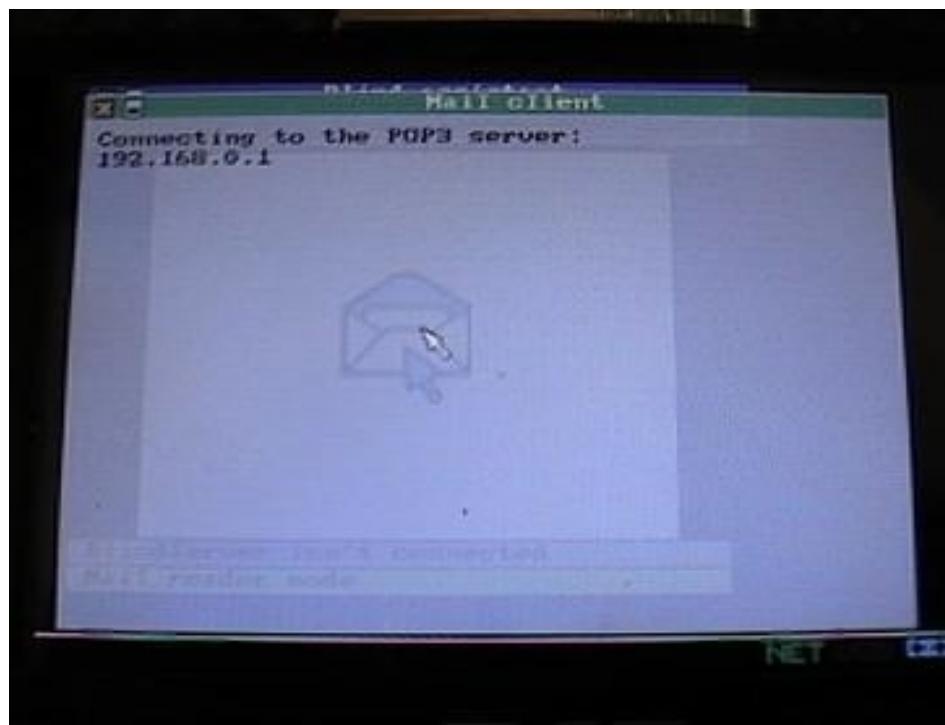
The advice will be repeated for other two times. At this point, the blind can decide to switch to mail reader mode or to continue in his current operation.

### Use of the client

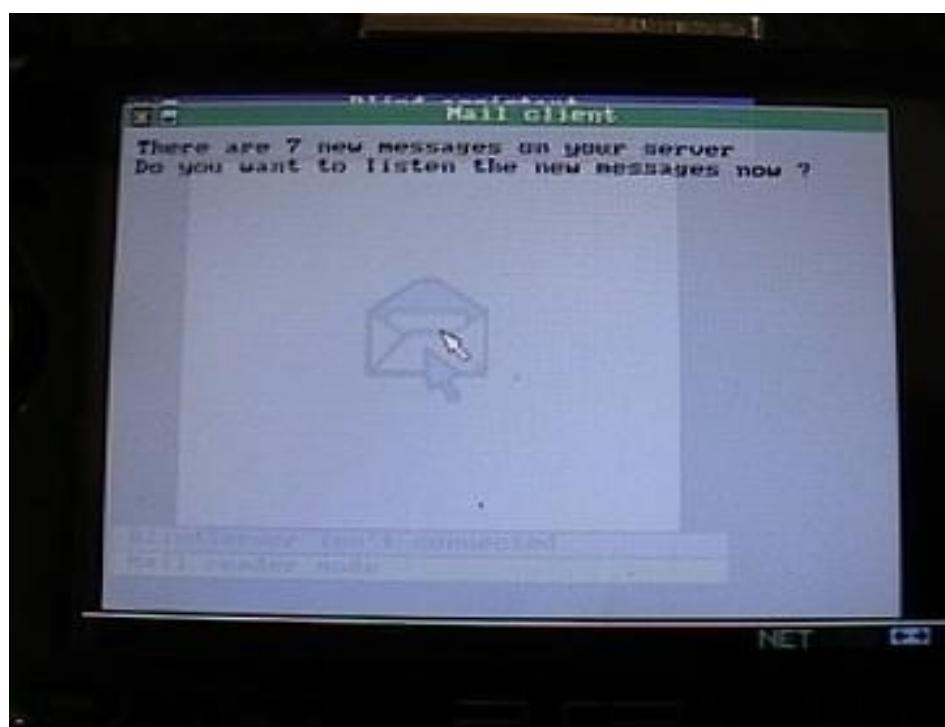
Now, switch the working mode of the device into the *mail reader* mode. Use the manual system (see chapter 3) or the voice control system (press the SQUARE button and say the command: "**Enable mail reader**").



Now, **press the TRIANGLE button:** Blind Assistant shall interrogate the POP3 server through the WI-FI network.

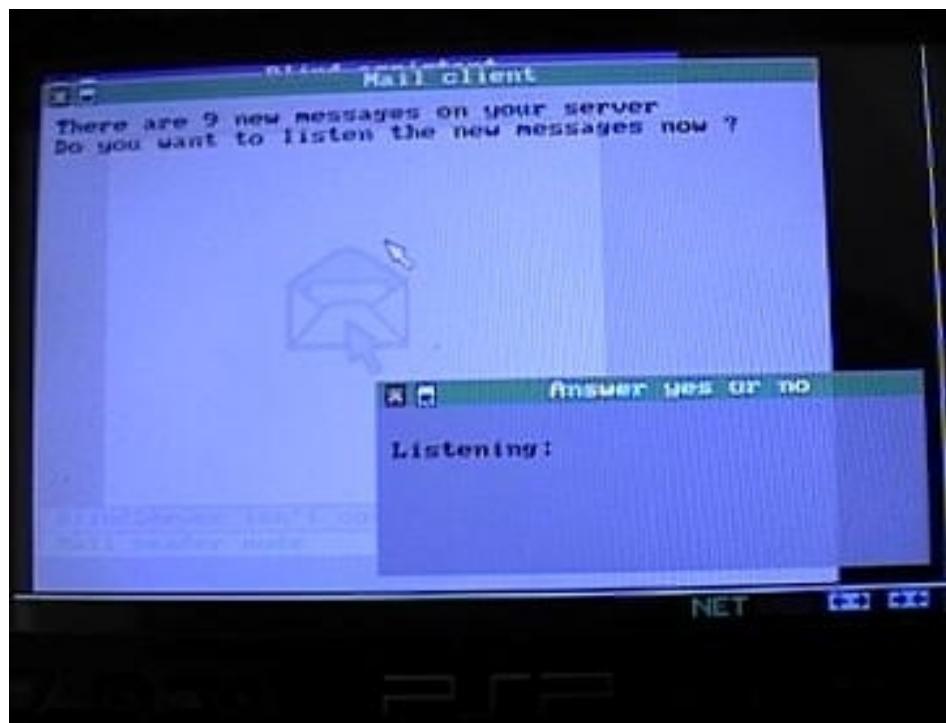


After few seconds, Blind Assistant will communicate (using a voice message) the number of mails that are stored in the server:

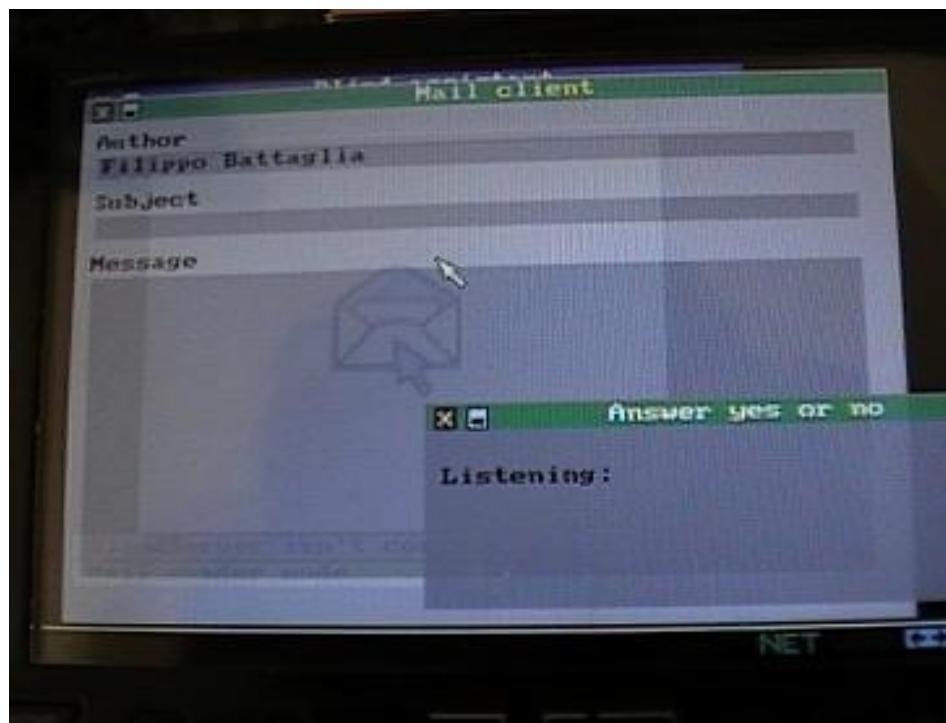


**IMPORTANT NOTE:** The mails are ignored if they have a body larger than 250KBytes. This is a necessary limitation that is useful because messages too large would be difficult to spell with the word by word strategy that is used by Blind Assistant.

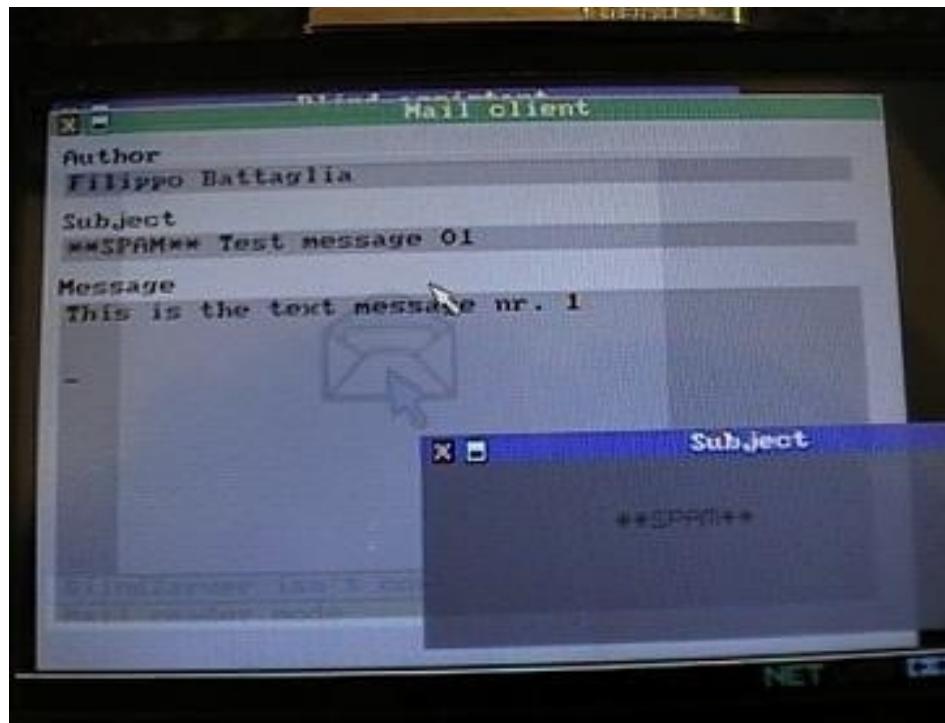
After this, Blind Assistant shall ask to the user if he wants to listen all messages now: he must answer Yes or No, using his voice.



If you answer Yes, the program will begin to manage each message. For each of those, it will communicate the name of the sender, and it will ask if the blind wants to listen the content of the mail.



If you answer Yes again, the system will begin to read to the blind the subject of the mail and the content. The reading operation is done word-by-word



If you want to stop the reading process, you can keep pressed the key TRIANGLE for some seconds.

When the process of reading of the single message is finished, Blind Assistant shall ask two things to the user:

- if he wants to listen again the message (answer Yes or No);
- if he wants to delete the message on the POP3 server (answer Yes or No);

### **End of the process**

When the entire process will be terminated for all mails, Blind Assistant will provide to delete the messages on the POP3 server in base to the commands that the user has given before.

A long beep shall inform the user that the software is in quiet mode again. If you want to check again if new mails have arrived, press a second time the TRIANGLE button.

### **What type of mails is supported ?**

Starting from Blind Assistant CFW0006, the VMIME library has been added to the system core. Thanks to this library, the mail reader is able to manage mails both in plain text and in HTML code.

The mails that contain images, dynamic elements, and other complex elements are not supported by the program. If the mail reader encounters one of these mails, it will communicate that its content cannot be decoded, using a voice message ("Error: I cannot decode the content of the mail").

## Chapter 11

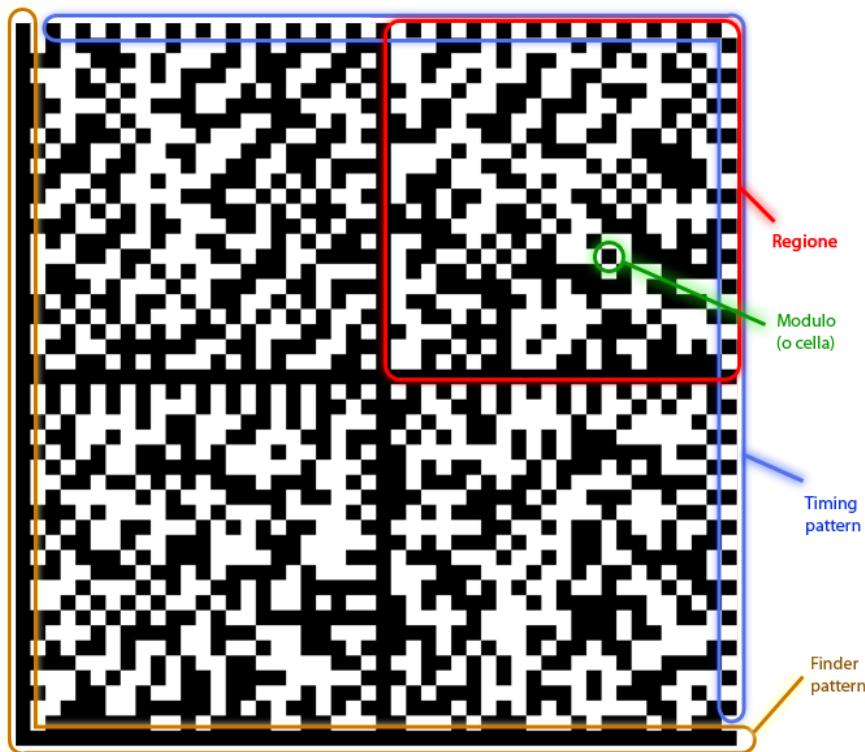
# Data Matrix scanner

In the daily life of a blind, it is often necessary to recognize an object or an obstacle: an envelope, a banknote, a particular device, a specific book. In other cases, can be useful to recognize a wall or a door in a cheap and easy way.

In 2005, a company called **International Data Matrix Inc** (ID Matrix) has introduced a special standard for bidimensional matrices, that can be printed on labels and objects.

The features of this standard have been standardized in the ISO/IEC 16022:2006 document: ([http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=44230](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44230))

Data Matrix standard is free: it can be used by the developers without paying any royalty. This is an example of Data Matrix [19]:



Important features of **Data Matrix standard** are [20]:

- support for messages long up to 3116 ASCII chars;
- robustness against noise;
- invariance respect to rotation and scaling
- error correction

Starting from the version CFW0006 of the core, Blind Assistant integrates a *scanner for data matrix*. This component is able to recognize the data matrices when they are framed by the GoCam camera.

19 from [http://it.wikipedia.org/wiki/File:Data\\_Matrix.png](http://it.wikipedia.org/wiki/File:Data_Matrix.png)

20 You can see the document at [http://www.gs1.org/docs/barcodes/GS1\\_DataMatrix\\_Introduction\\_and\\_technical\\_overview.pdf](http://www.gs1.org/docs/barcodes/GS1_DataMatrix_Introduction_and_technical_overview.pdf)

This is made possible by a library called **libdmtx** (<http://www.libdmtx.org/>), that has been integrated in the new core of the program.

**LibDMTX** is a library able to detect automatically the presence of a data matrix in a frame. The library provides also to decode quickly its content.

## How can I create a data matrix label ?

Blind Assistant doesn't integrate any tool for creating and printing the data matrix labels.

However, you can simply generate an image file that contains your data matrices (encoding your strings) using the utility **dmtxwrite**, downloadable from

<http://www.libdmtx.org/downloads.php>

The tool is available for Windows and for Linux. The Ubuntu users can also download the tool using **Synaptic Package Manager**.

Install dmtxwrite in your PC following the instructions of the site. After, open the bash/command prompt and type a command like this:

```
dmtxwrite dmtxmessage.txt --output dmtxmessage.jpg
```

Where *dmtxmessage.txt* must be replaced with the name of the text file that contains the message to encode, and *dmtxmessage.jpg* is the output image that shall be generated by the tool.

After this, you'll be able to print the image *dmtxmessage.jpg* using your printer. Cut the contours of the matrix using a clipper, and attack the paper to the object, wall or door using two pieces of tape.

## Switch Blind Assistant to Data Matrix mode

Now, you must enable the data matrix reader integrated in Blind Assistant.

The blind can use the manual switching (see chapter 3) or he can put the system in *listening mode* and utter the words "**Enable data matrix scanner**".

The system will answer with a voice message "*Data matrix scanner enabled*". See this photo:

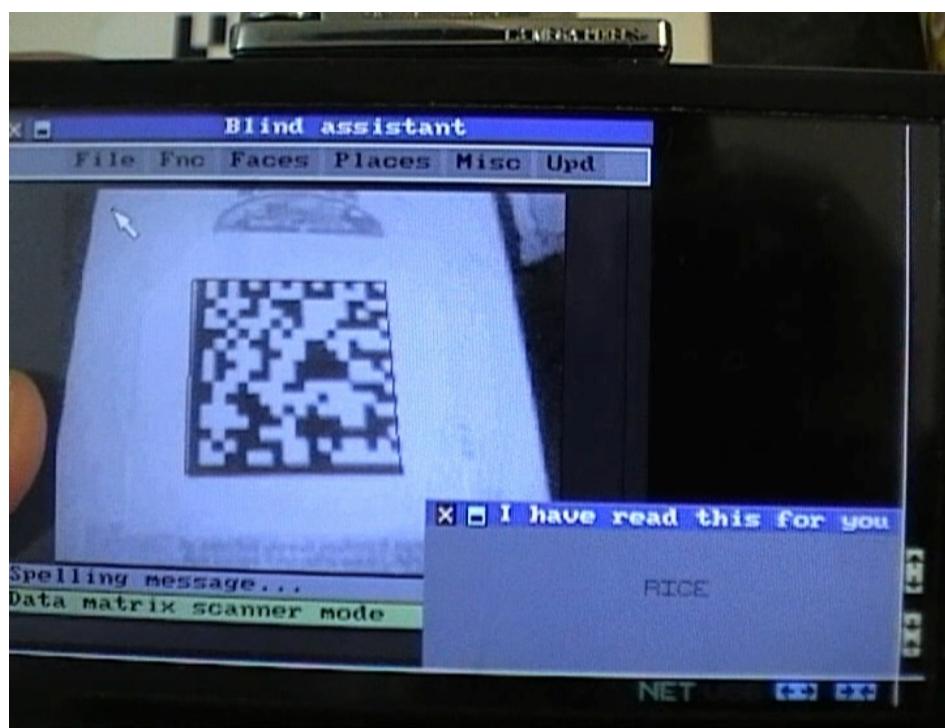


The scanner integrated in Blind Assistant works with a resolution of 640x480 pixel and a frequency of 1 Hz. It is able to recognize automatically if a data matrix is present in a frame.

No operation by the user is required.

In our tests, the system has been able to recognize a label of 5x5 cm. at a maximum distance of 80 cm.

When a data matrix is recognized, a short beep is emitted: the system decodes automatically the matrix, and passes the content to the *ndFLite* voice synthesizer.



The content of the matrix is uttered word-by-word: in any moment, the user can stop the process of utterance simply keeping pressed the TRIANGLE button.

It's all!

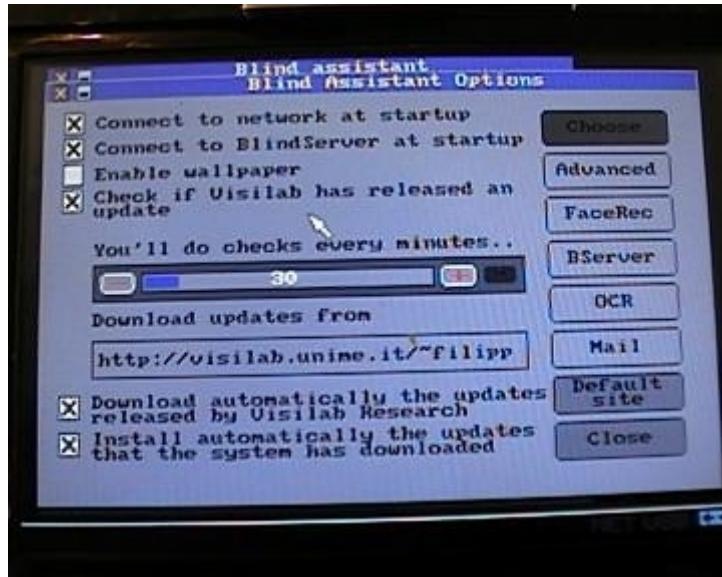
## Chapter 12

# AutoUpdate

Blind Assistant is an open-source software. It is constantly improved by the author, with bug-fixes, new functions, new algorithms and so on.

The software can autoupdate itself using the Internet connection. *This function is performed automatically: so the blind doesn't need to operate manually.*

Go to the **Misc** menu and choose the **Options** item.



In this dialog box there are several options that are connected to the *AutoUpdate* feature.

### Connect to network at startup

When this checkbox is enabled, the program will connect automatically to the network when it is started.

### Check if Visilab has released an update

When this checkbox is enabled, the program will check periodically if Visilab has released a new version of the software. The period between a check and another is determinated by the trackbar that you see in the figure: the default value is 30 minute.

### Download automatically the updates released by Visilab Research

When this checkbox is enabled, the system will download automatically the update packages that are released by Visilab. The packages are downloaded in the folder ms0:/BLINDASSISTANT/UPDATES/

### Install automatically the updates that the system has downloaded

When this checkbox is enabled, the system will install automatically the new core that it has downloaded.

If you want to enable the autoupdate feature, you must enable the following options:

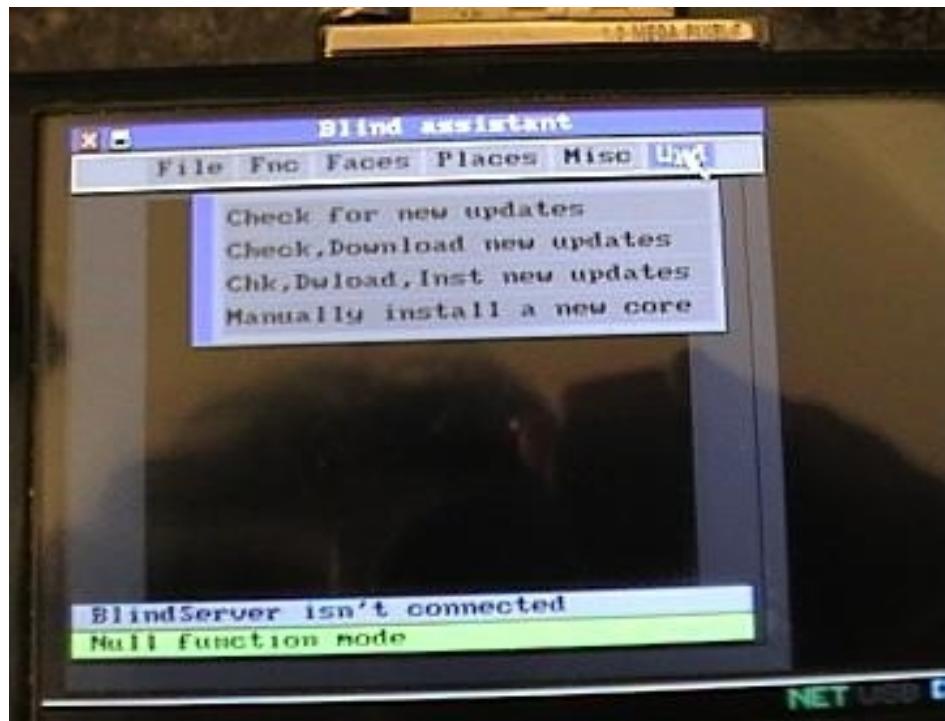
- **Connect to network at startup;**
- **Check if Visilab has released an update;**
- **Download automatically the updates released by Visilab Research;**
- **Install automatically the updates that the system has downloaded.**

When you've finished, click on **Close** button and the system will save the options in the configuration file of the program.

## Manual update

Even if you have disabled the *autoupdate feature*, you can update your system manually in any moment.

Go to the menu **Upd**, in the main interface.



Here, there are three items.

### **Check for new update**

If you select this, the system will perform a simple check in order to understand if Visilab has released a new version of the program. If this event occurs, the system will only generate a message "*Visilab has released an update*".

### **Check and download new updates**

If you select this, the system will check if Visilab has released a new update and it will download it into the memory stick. The update is not installed.

### **Check, download and install new updates**

This option checks if Visilab has released a new update, downloads it locally and installs it automatically.

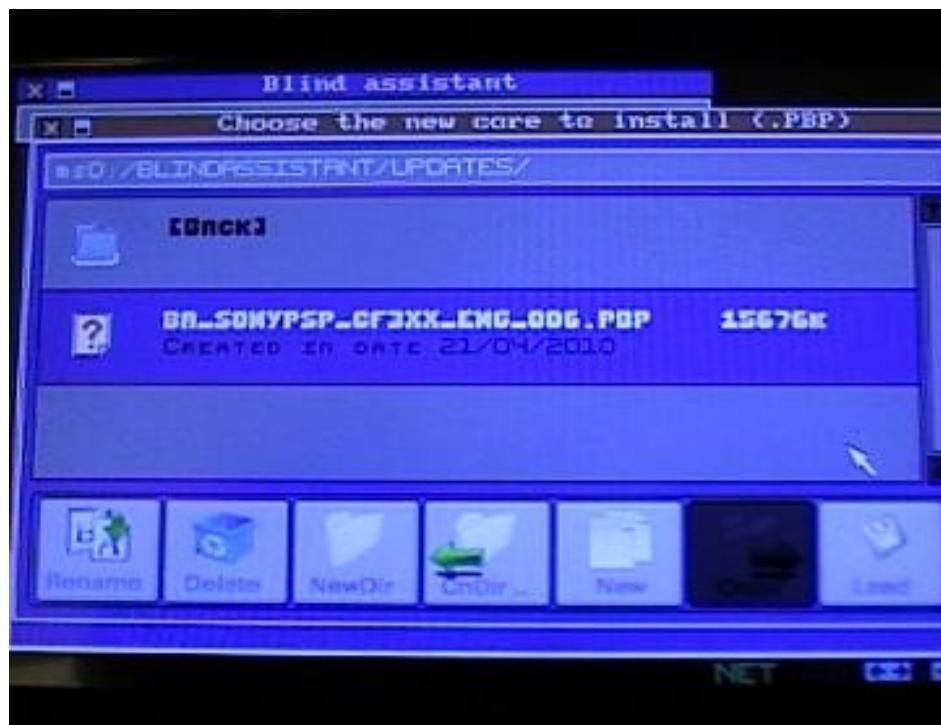
### **Manually install a new core**

Perhaps you have troubles in connecting your PSP to internet through your router. In this case, you cannot use the function of autoupdate.

In this case, you can download a new core and you can install it manually. Copy the new core from Visilab website (for example, you can ask it to the developer of this program) and store it into the memory stick from a PC. After, start Blind Assistant.

Now, go to menu **Upd** and choose **Manually install a new core**.

The file manager will appear on the screen:



Select the core file (.PBP extension) that you want to install. Click on **Load** button and the process of installation will begin.

## Appendix A

# Where you can find help

If you have troubles with this program, you can write to the author ([filbattaglia@libero.it](mailto:filbattaglia@libero.it)), or you can write a message on the psp-ita forum (<http://www.psp-ita.com/forum>).

---

## Appendix B

# ChangeLog

This a list of changes that have been done in this core:

POC

First release (pre-alpha)

CFW0001

Introduced support for GoCam  
Improved speed of recognition

CFW0002

Introduced BlindServer vers. 1.0  
New optimized VFPU algorithm for face recognition  
SIFT code has been removed by BlindClient (is in the server now)  
Blind Assistant is able to recognize where the blind is.  
Blind Assistant integrates OCR features  
Introduce support for fast voice synthesis

CFW0003

Introduced BlindServer 1.1: engine (it has been totally redesigned)  
Introduced /siftdebug option  
New version of BAWCP protocol  
Improved speed of face recognition  
Introduced new cvhaarascade file  
Introduced trackbars for the probe area and the threshold of the face detector  
New criterium for skip the face recognizer results  
Several bugs fixed  
More efficient VFPU code for face recognition

CFW0004

Introduced Blind Assistant Loader 2.0  
Introduced core for PSP FAT  
Fixed compatibility troubles with PSP FAT firmwares  
Incorporated new nd strategy for USB drivers (Nanodesktop 0.3.5)

CFW0005

Introduced ndPocketSphinx support for voice recognition  
Added a colors scanner  
Added a mail reader  
Face recognition uses a four-zones brightness balance system

CFW0006

Face recognizer: introduced a new dynamic memory manager  
Position recognizer: BlindClient can synchronize its database with BlindServer if there are differences  
OCR: ndOcrad engine for on-board scanning has been replaced by ndTesseract  
OCR: BlindServer has been updated to 1.2 version  
OCR: The engine uses a new image filter with erosion now  
Introduced Data Matrix scanner  
Introduced VMIME library in the core: now the mail reader can manage HTML mails  
Introduced mail autochecker  
PSP-2000 and PSP-1000 share the same engine now  
Introduced system fixers: components that fix automatically the problems that are detected in the system  
The ndFLite voice synthesizer use Varangozov's memory manager now  
The video interface uses Nanodesktop 0.5 accelerations  
PocketSphinx voice recognizer has been improved

## Appendix C

# Eyeserver

Blind Assistant can be used also through *Eyeserver technology*.

Eyeserver can be used to connect a normal PC webcam to a Blind Assistant unit. The utility of this is, at the moment, only theoretical, because Visilab hasn't released a version of Eyeserver that works on the embedded processors (the only released version is for Windows), but the things will change in the future.

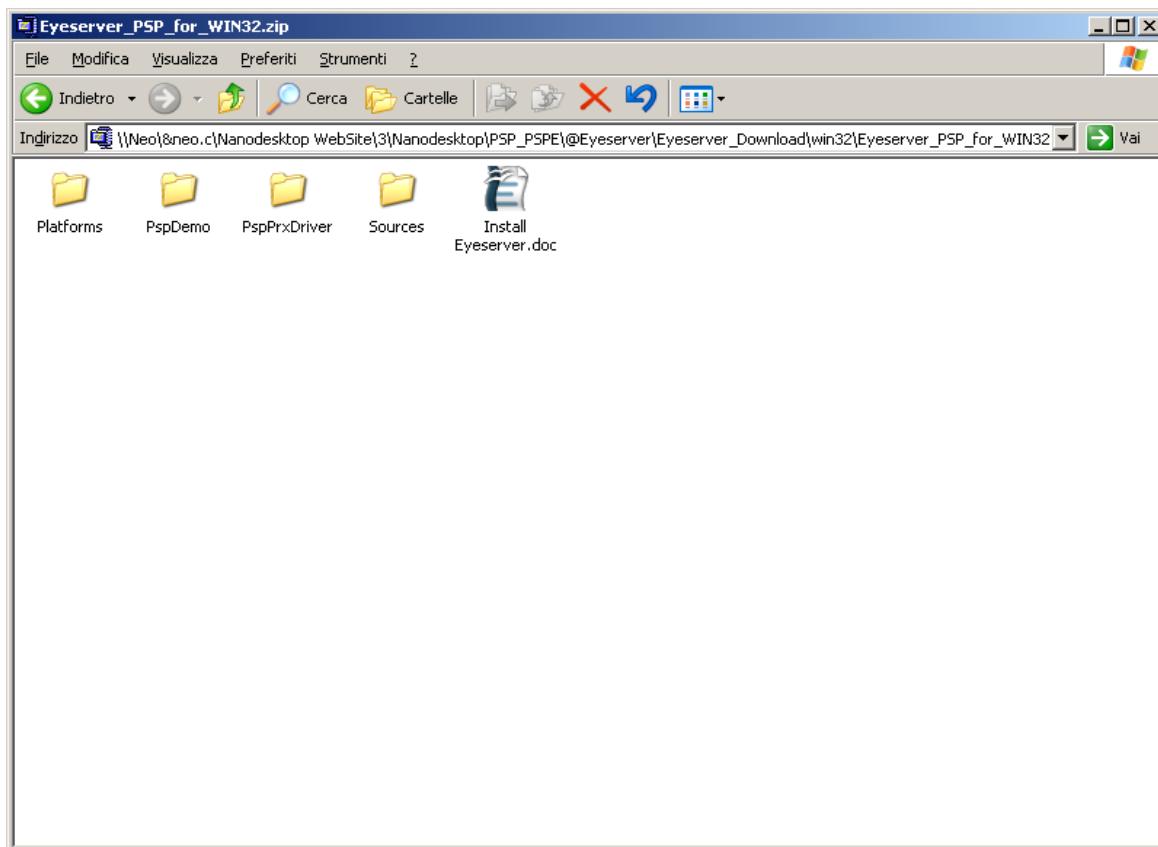
Eyeserver can be used if you want to see how Blind Assistant works but you haven't a Sony GoCam at home.

## Install Eyeserver

The first thing to do, in this case, is to download Eyeserver package from Visilab website, at this address:

[http://visilab.unime.it/~filippo/Nanodesktop/PSP\\_PSPE/@Eyeserver](http://visilab.unime.it/~filippo/Nanodesktop/PSP_PSPE/@Eyeserver)

In the package *Eyeserver\_PSP\_win32.zip* (or the equivalent if you are using Eyeserver on other devices as Fox), you'll find a document like this:



Follow the instructions that are contained in that document: you must prepare Eyeserver and the test, done with the homebrew that is included in Eyeserver package, has to be completed successfully. Now, *connect your PC (with Eyeserver running) with your PSP with an USB cable*.

**Start Blind Assistant Loader:** it will load the Blind Assistant core and, after, it will begin the process of looking for a camera.

Your Windows OS will detect a USB attach, followed by an USB detach. **Ignore them.** In fact, this happens because Nanodesktop first tries to connect to a Sony GoCam, and only in a second time it tries to connect to a PC for Eyeserver.

After the sequence USB attach/USB detach (for checking the presence of GoCam), the PSP will do a second USB attach operation and it will start the Eyeserver connection.

You'll see that Nanodesktop will attempt for 4 times to connect to EyeServer.



Be careful: it is possible that the connection fails also if you have correctly connected the USB cable between the PC (with Eyeserver running normally on it) and the PSP.

In this case Blind Assistant will signal that the camera isn't connected, also if it seems that is all ok.

If this happens, you must do two things:

- a) ensure yourself that there is no trouble with *ndUsbDriverCF.Prx*. This is simple to check, because in this case the icon USB in the lower right angle of the screen of the PSP will become *red*.
- b) perhaps there is the usual *Eyeserver connection trouble (ECT)*. This trouble is known. The reason of the trouble is in an incorrect software interaction between Eyeserver and the USB library (the problem is in Windows API, not in Eyeserver). The solution, fortunately, is very simple: try to detach and reattach manually the USB cable that connects the PSP device to the PC. After, says to Blind Assistant to retry the connection.

When you have created a valid connection between EyeServer and Blind Assistant, you can proceed in the same way seen in chapter 1.

**Note: When you use Eyeserver, you could see ndHighGUI errors that say that it is impossible to change CCD Hw Resolution. It is perfectly normal and you can ignore them.**

