



## Robot de telepresència: imatge i so

Joan Serra Ortega  
Guillem Alenyà

Juliol, 2013



# Robot de telepresència: imatge i so

El robot Helena (Fig. 1) està dissenyat per ser un robot de telepresència que ha de servir per comunicar diferents persones d'un edifici sense necessitat de moure's.



Imatge 1: Robot Helena amb la càmera usada al capdamunt.

## 1. Motivacions

Per complir l'objectiu de disseny del robot, entre d'altres funcions, és necessari veure en temps real el que està "veient" el robot (captant imatges a través d'una càmera web) a través d'un ordinador amb connexió a la xarxa i un navegador web. És per aquest motiu que s'ha optat per treballar en aquest apartat tal com recull aquest informe.

## 2. Objectius

- Instal·lar una càmera web al robot Helena
- Emetre el que capta la càmera web per la xarxa.
- Visualitzar en temps real el que "observa" el robot.

### 3. Desenvolupament

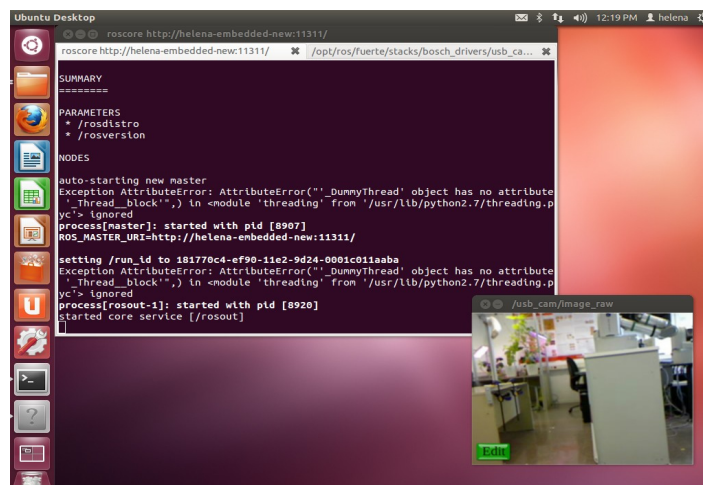
Aquest robot funciona amb el sistema operatiu *ROS*<sup>1</sup> (*Robot Operating System*), el qual permet dotar al robot d'aplicacions especials adaptades per funcionar en robots. En aquest cas, l'aplicació que es buscava era una que fos capaç de captar la imatge de la càmera web, publicar-la en un *ROS topic* poder-la emetre per visualitzar-la en el navegador web d'un altre ordinador. Com que no existeix una sola aplicació que faci totes aquestes funcions, gràcies al funcionament del *ROS* basat en nodes és possible fer-ho mitjançant el *roscore* (el node que permet la comunicació entre altres nodes), el node que conté el driver de la càmera (permet la visualització de la imatge a l'ordinador on està connectada) i el que emet la imatge de la càmera per la xarxa. Per tant, per fer possibles els objectius hem de tenir en funcionament 3 nodes: el *roscore* (que és el principal) i els altres 2 nodes.

#### 3.1 Funcionament de la càmera

Per fer funcionar la càmera web s'ha utilitzat un driver comú que és compatible pràcticament amb tots els models de càmeres web. És un paquet del *ROS* amb el nom *usb\_cam*<sup>2</sup>. El driver permet visualitzar la imatge de la càmera web i publica-la en un *ROS topic*, en aquest cas publica la imatge en el *ROS topic* anomenat */usb\_cam/image\_raw*.

##### 3.1.1 El servidor emissor de la imatge

Per emetre la imatge amb el *ROS topic* */usb\_cam/image\_raw*, s'utilitza el paquet de *ROS* *mjpeg\_server*<sup>3</sup>. Aquest programa el que fa és publicar tots els *ROS topic* que s'estan emetent a una pàgina web. Per tant, si s'entra en un navegador i s'escriu l'adreça [http://<ip\\_del\\_robot>:8080/stream?topic=/usb\\_cam/image\\_raw](http://<ip_del_robot>:8080/stream?topic=/usb_cam/image_raw) quan el driver de la càmera està engegat i l'*mjpeg\_server* està emetent, es pot visualitzar la imatge de la càmera web del robot. Des del navegador s'escull el *ROS topic* que es vol visualitzar, podria ser útil si el robot tingués dues càmeres ja que només s'hauria de canviar el nom del *ROS topic* de la *url* abans esmentada.

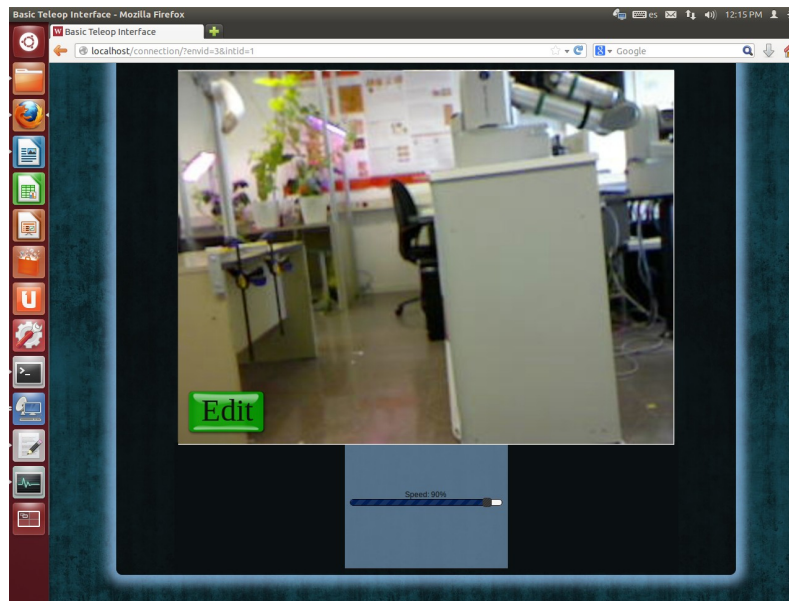


Imatge 2: terminal de Linux amb el roscore i imatge de la càmera web (en local).

##### 3.1.2 Detalls de la imatge

La imatge que es visualitza en el navegador té una resolució de 320x240 píxels. Degut a que el

robot s'ha de comunicar mitjançant el *wifi* amb la xarxa no és recomanable enviar una imatge de més resolució ja que la connexió podria quedar saturada i es podria aturar la visualització en directe de la imatge. També s'ha de tenir en compte que no només són les úniques dades que es transmetran per aquesta connexió.



Imatge 3: captura de pantalla del client connectat al robot.

## 3.2 Càmera web del Robot Helena

### 3.2.1 Requeriments

- ROS<sup>1</sup>: <http://www.ros.org/wiki/ROS/Installation>
- Driver de la càmera USB<sup>2</sup>: `sudo apt-get install ros-<versió>-usb-cam`
- Servidor MJPEG<sup>3</sup>: `sudo apt-get install ros-<versió>-mjpeg-server`

S'han d'instal·lar els aquests tres programes. S'ha provat el seu funcionament amb les versions de *ROS Fuerte* i *Groovy*. Funciona en les dues versions, però si s'ha d'instal·lar el *ROS* és recomanable escollir la versió *Groovy* perquè el funcionament del *driver* de la càmera és més estable.

### 3.2.2 Procediment

Un cop tot instal·lat, s'ha de crear l'arxiu “usb\_cam.launch” per tal de poder executar el *driver* de la càmera i veure'n la imatge. Aquest arxiu s'ha de crear dins del *rospackage* “usb\_cam”, corresponent al directori “/opt/ros/<versió>/share/usb\_cam”. El codi de l'arxiu “usb\_cam.launch” és el següent:

```
<launch>
  <node name="usb_cam" pkg="usb_cam" type="usb_cam_node" output="screen" >
    <param name="video_device" value="/dev/video0" />
    <param name="image_width" value="640" />
    <param name="image_height" value="480" />
    <param name="pixel_format" value="mjpeg" />
    <param name="camera_frame_id" value="usb_cam" />
  </node>
</launch>
```

```

    <param name="io_method" value="mmap"/>
  </node>
  <node name="image_view" pkg="image_view" type="image_view" respawn="false"
output="screen">
    <remap from="image" to="/usb_cam/image_raw"/>
    <param name="autosize" value="true" />
  </node>
</launch>

```

La línia 3 de codi permet configurar l'entrada d'imatge, si no es rep imatge en el moment d'executar el driver canviar el valor del “video\_device”, però en les proves fetes amb aquest valor ha funcionat sempre.

Un cop creat l'arxiu, cal crear la pàgina web per tal de poder-ho veure a través d'un navegador. Per aconseguir això es pot fer a través del servidor RMS<sup>4</sup> (*Robot Management System*) que permet controlar a través de la web el robot o crear una pàgina web concreta per veure la imatge. A la bibliografia trobareu l'enllaç del RMS hi ha tota la informació necessària per crear aquest servidor i fer-lo funcionar.

Per visualitzar només la imatge a través del navegador s'ha de crear una web amb el codi HTML següent:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<script type="text/javascript"
src="http://cdn.robotwebtools.org/EventEmitter2/current/eventemitter2.js">
</script>
<script type="text/javascript"
src="http://cdn.robotwebtools.org/mjpegcanvasjs/current/mjpegcanvas.min.js">
</script>

<script type="text/javascript" type="text/javascript">

function init() {
  // Create the main viewer.
  var viewer = new MJPEGCANVAS.MultiStreamViewer({
    divID : 'mjpeg',
    host : '<ip_del_robot>',
    width : 640,
    height : 480,
    topics : [ '/usb_cam/image_raw' ],
    labels : [ 'Robot' ]

```

```

    });
}

</script>
</head>

<body onload="init()">
  <h1>Robot Helena</h1>
  <div id="mjpeg"></div>
</body>
</html>

```

Aquest codi s'ha d'adequar a l'adreça *ip* del robot, només s'ha de posar la *ip* on s'indica. També es poden canviar altres valors, com la mida de la imatge. Cal copiar aquest codi HTML en un arxiu en format HTML i desar-lo en un directori qualsevol.

### 3.2.3 Execució

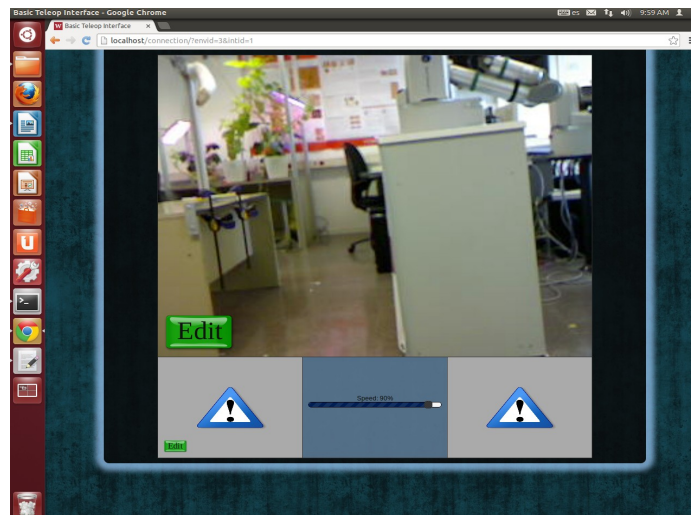
Si no es controla el robot directament (amb el seu propi teclat i pantalla), s'ha d'accedir al robot a través d'un túnel *ssh* (si no funciona l'exportador de pantalla, cal connectar un teclat al port *USB 2.0* del robot i iniciar la sessió).

Primer de tot, engegar un *roscore* obrint un terminal i executant aquesta ordre. Seguidament, en un nou terminal introduir l'ordre *roslaunch usb\_cam usb\_cam.launch* i executar-la. Si el ROS ha reconegut correctament el *rospackage* “*usb\_cam*” s'executarà, però si no l'ha reconegut s'ha d'executar l'arxiu “*usb\_cam.launch*” dins del directori on es troba. Si s'executa correctament el driver i es veu la imatge, executar l'ordre *roslaunch mjpeg\_server mjpeg\_server*. Amb això s'executarà el servidor que emetrà la imatge a la web.

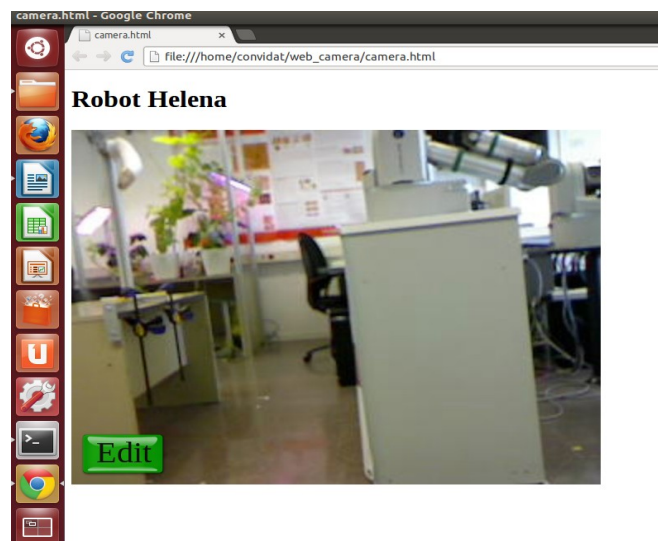
Per visualitzar la imatge a un navegador hi ha tres camins possibles: a través del servidor RMS, a través de la pàgina web creada per aquest motiu o directament veient la imatge emesa per l'*MJPEG\_server*.

1. Es pot visualitzar la imatge que emet directament el servidor *mjpeg\_server*. Per fer-ho, en un navegador introduir la web “[http://<ip\\_del\\_robot>:8080/stream?topic=/usb\\_cam/image\\_raw](http://<ip_del_robot>:8080/stream?topic=/usb_cam/image_raw)”. S'hauria de veure en directe la imatge captada per la càmera del robot.
2. Mitjançant el servidor RMS: en aquest cas, introduir l'adreça *ip* del servidor RMS en un navegador i iniciar sessió amb el nom d'usuari i contrasenya que s'han definit al crear-lo. La primera vegada en que s'obri demanarà diverses dades, omplir-les correctament i seguir. Després s'ha de configurar per tal de poder veure la imatge del robot, després d'iniciar sessió anar a *Admin Panel* -> *Manage Environments* i afegir o modificar un *Environment* amb la *ip* del robot. També cal crear un nou *Environment-interface pairing* amb la *ip* del robot i la interfície *Basic Teleop*. Afegir un nou *MJPEG Stream* amb l'*environment* del robot i el *topic* “*/usb\_cam/image\_raw?quality=95*”. Posant *quality=95* (correspon a un paràmetre de l'*MJPEG\_server*) reduïm una mica la qualitat d'imatge però s'alleugereix el trànsit de dades per la xarxa. Un cop fet això, és necessari afegir un *Keyboard Teleop* amb els paràmetres per defecte, encara que no s'utilitzi. Clicar a *Main Menu* i comprovar si el robot està *online* o *offline*. Si hauria d'estar *online* però no ho està, cal instal·lar i executar el *rosbridge\_server*. Si està *online*, clicar a *Basic Teleop* i s'hauria de veure la imatge captada pel robot. Si només es veu el primer fotograma, entrar al servidor a través d'un altre navegador compatible amb

*javascript*. S'hauria de veure aquesta web:



3. Mitjançant la web creada: només cal obrir la web en un navegador compatible amb *javascript* i es visualitzarà la imatge. Si només es veu el primer fotograma, obrir la web en un altre navegador. S'hauria de veure aquesta imatge:



### 3.2.4 Errors coneguts

#### 3.2.4.1 Error en el driver

Pot ser que el *driver* de la càmera tingui un error i provoqui que no es vegi imatge, que es vegi en negre:



```
/opt/ros/forte/stacks/bosch_drivers/usb_cam/usb_cam.launch http://localhost:11311
/opt/ros/forte/st...  roscore http://hele...  helena@helena-em...  /opt/ros/forte/st...
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
[mjpeg @ 0xb8eb60] No JPEG data found in image
Error while decoding frame.
```

Aquest error pot ser un problema del *driver* o del *hardware*. En aquest cas, reiniciar el *driver* de la càmera.

### 3.2.4.2 Error en el servidor MJPEG

Si el servidor d'imatge té moltes entrades i sortides, si no es desconnecta bé pot provocar que la següent vegada que es vulgui accedir l'stream el navegador es quedi carregant i no hi accedeixi. En aquest cas, tancar les connexions obertes i reiniciar l'*mjpeg\_server*.

## 4. Conclusions



Imatge 7: exemple d'us, on es pot veure l'usuari amb el robot i les imatges que el robot envia a un ordinador qualsevol.

S'ha aconseguit visualitzar el que capta la càmera web del robot a través de qualsevol altre ordinador amb connexió a internet i un navegador web, per tant s'han complert els objectius. Però poder fer un robot de telepresència complet, falta establir una via comunicació auditiva. El ROS ofereix un paquet anomenat *audio\_common*<sup>5</sup> que permet escoltar el micròfon connectat a un ordinador i publicar-ho en un ROS tòpic, però no hi ha cap servidor que emeti aquest tòpic per internet i tampoc hi ha un descodificador. Falta doncs, la part que comunica el ROS topic amb el navegador i que ha de tenir la capacitat de codificar i descodificar àudio.



## 5. Bibliografia

1. ROS: <http://www.ros.org/wiki/>
2. Driver *usb\_cam*: [http://www.ros.org/wiki/usb\\_cam](http://www.ros.org/wiki/usb_cam)
3. *mjpeg\_server*: [http://www.ros.org/wiki/mjpeg\\_server](http://www.ros.org/wiki/mjpeg_server)
4. servidor RMS: <http://www.ros.org/wiki/rms>
5. *audio\_common*: [http://ros.org/wiki/audio\\_common/](http://ros.org/wiki/audio_common/)