Watson

Overview

Watson is a real-time visual feedback recognition library for interactive interfaces that recognizes head gaze, head gestures, eye gaze and eye gestures using the images of a monocular or stereo camera. Please see these sites for details of this component:

- WATSON: Real-time Head Tracking and Gesture Recognition
- Watson Head Tracker

Users

Launching

You can start Watson from the toolkit Launcher as part of the Rapport 1.0 system. By default, Watson for the Toolkit is configured to work with a USB camera using the VFW (Video for Windows) driver. If using a different camera or driver, change the camera type parameter accordingly. See the developer section below for details.

Developers

Configuration

Watson comes configured to work properly with a USB webcam. If you wish to change any of the parameters below, make modifications to this file: ParamWatsonUser.cfg found in bin\Watson\Sequences\USB.

Changing the camera type

By default, Watson for the Toolkit is configured to work with a USB camera using the VFW (Video for Windows) driver. If using a different camera or driver, change the camera type parameter accordingly.

SECTION_SEQUENCE

CAMERA_TYPE: USB_VFW

CAMERA_TYPE is USB_DIRECTX for DirectX driver or DRAGONFLY for Point Grey DragonFly camera. If using Stereo Camera, please see the section below 'Configuration for Stereo Camera'.

Configuration for Stereo Camera

To use Watson with the Toolkit using a Stereo Camera, look in bin\Watson\Sequences\SRI. However, the configuration file 'ParamWatsonUser.cfg' in this directory is not configured to work with the rest of the Toolkit by default. Add the section to set up ActiveMQ messaging to the ParamWatsonUser.cfg. This will enable Watson to send messages and work with the rest of the Toolkit. See the 'Setting up ActiveMQ messaging' section below for activating ActiveMQ messaging for the Stereo Camera.

Setting up ActiveMQ messaging

Note: If using the core\Watson\Sequences\USB configuration for VH toolkit, this section should already be set up in the configuration file. If using other Watson configuration (such as core\Watson\Sequences\SRI for Stereo Camera, add this section to the ParamWatsonUser.cfg file

To activate ActiveMQ the following section is added:

SECTION_NETWORK

CONNECT_ACTIVEMQ: TRUE

TYPE_INFO_SENT_ACTIVEMQ: INFO_HEAD_NODS INFO_FACING_OBJECT END

This activates ActiveMQ messaging and will start sending messages related to head nods and facing. The facing message tells which object (or character) you are facing.

Setting up Regions of Interest

The commands related to facing are defined so a region in 3D space is created and associated to a caption. To create the region of interest (ROI), imagine a plane parallel to the image plane of the camera. e.g., In case of a webcam in a laptop, the computer screen is already parallel with the camera. The camera is the referential(0,0,0). When looking at the camera, the X axis goes (positive) to the right. The Y axis goes (positive) up.

Regions of Interest on the screen can be associated by adding the appropriate caption and coordinates as shown in the example below. The example assumes that we are creating two Regions of Interest with captions "doctor" and "elder" and shows how to associate coordinates to these captions in order to create ROIs.

A "doctor" region can be associated to the right part of the screen by adding these two commands at the end of ParamWatsonUser.cfg (after the section statement):

```
OBJECT1_CAPTION: doctor
```

```
OBJECT1_ROI: 0 -1000 1000 1000
```

The Roi parameter are: offset_x offset_y width height. Everything is in millimeter.

Similarly an "elder" region can be set up on the left side of the screen with:

```
OBJECT2_CAPTION: elder
```

```
OBJECT2_ROI: -1000 -1000 1000 1000
```

To debug the object region of interest, the following options can be added to the TYPE_INFO_SENT_ACTIVEMQ parameter

```
TYPE_INFO_SENT_ACTIVEMQ: INFO_SCREEN_COORDS INFO_HEAD_NOD INFO_FACING_OBJECT END
```

This will send a new massage at every frame with the projections of the head gaze on the screen (in millimeter).

Message API

Sends

- vrVision

Receives

Known Issues

FAQ

FAQ