POLITECHNIKA WROCL AWSKA
WYDZIAL ELEKTRONIKI

Kierunek: Automatyka i Robotyka (AIR)
Specjalność: Embedded robotics (AER)

PROJECT

Inteligent mirror with face recognition system

Inteligentne lustro z systemem rozpoznawania twarzy

Autor:
Mateusz Banach

Introduction:

Main task of the project was to create autonomous intelligent mirror and develop algorithm to control additional functionality by face recognition system. To archive this task OpenCV library was used on the software part, on the hardware the project was basing on Raspebrry Pi 2.

Theory:

Face recognition system is a computer application capable of identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a face database.
It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Recently, it has also become popular as a commercial identification and marketing tool.

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel's research center in Nizhny Novgorod (Russia), it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license.

Hardware:

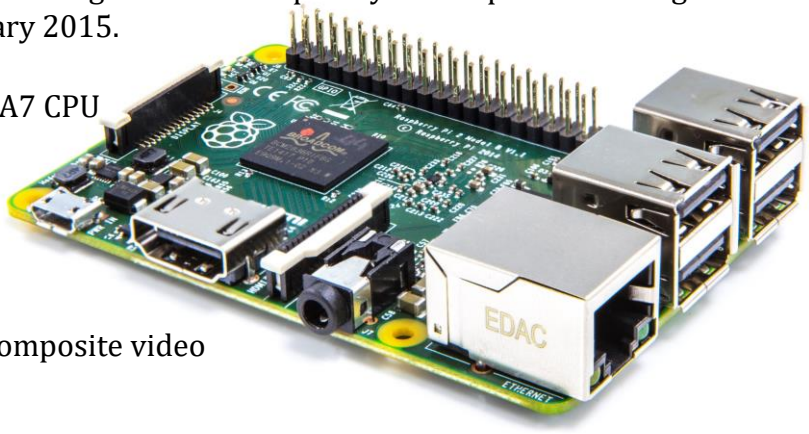
List of hardware using in project

- Raspberry Pi 2 Model B
- Speakers
- LCD monitor
- Two Way Mirror Film
- Camera RapiCAM IMX219
- Power supplies, hdmi cable, micro-usb cable

Raspberry Pi 2

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. Peripherals (including keyboards, mice and cases) are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles.

The Raspberry Pi 2 Model B is the second generation Raspberry Pi. It replaced the original Raspberry Pi 1 Model B+ in February 2015.



A 900MHz quad-core ARM Cortex-A7 CPU
1GB RAM
4 USB ports
40 GPIO pins
Full HDMI port
Ethernet port
Combined 3.5mm audio jack and composite video
Camera interface (CSI)
Display interface (DSI)
Micro SD card slot
VideoCore IV 3D graphics core

OpenCV & Face recognition

Instalation and configuration of Raspbian on Raspbbery Pi.

Raspbian is a Debian-based computer operating system for Raspberry Pi. The Raspbian was downloaded from official Raspbbery Pi website [https://www.raspberrypi.org/downloads/raspbian/].

Setting up and installation of the camera hardware.
The flex cable inserts into the connector situated between the Ethernet and HDMI ports.
To configure camera fter setting up was used following commands:

sudo apt-get update
sudo apt-get upgrade
sudo raspi-config

After that camera can be enable in options. To use camera we can use commands as:
raspivid is a command line application that allows you to capture video with the camera module, while the application *raspistill* allows you to capture images.

-o or *–output* specifies the output filename and *-t* or *–timeout* specifies the amount of time that the preview will be displayed in milliseconds.

To stream video from the Raspberry Pi camera over a network we need to install the dependencies by running the following in a terminal:
sudo apt-get install mplayer netcat

Before working with OpenCV, was needed to buid and install repositories with source code for ARM side libraries [*from https://github.com/raspberrypi/userland]*
From repositories are using few useful components like aspiStill.c, camcv.c etc.
For firstly compiling and tests these components neeed appropriate permissions by *chmod* command and CMakeLists.txt file should be change for:

```
cmake_minimum_required(VERSION 2.8)
project( camcv )
SET(COMPILE_DEFINITIONS -Werror)
include_directories(/opt/vc/userland/host_applications/linux/libs/bcm_host/include)
include_directories(/opt/vc/userland/interface/vcos)
include_directories(/opt/vc/userland)
include_directories(/opt/vc/userland/interface/vcos/pthreads)
include_directories(/opt/vc/userland/interface/vmcs_host/linux)
add_executable(camcv RaspiCamControl.c RaspiCLI.c RaspiPreview.c camcv.c)
target_link_libraries(camcv /opt/vc/lib/libmmal_core.so /opt/vc/lib/libmmal_util.so /opt/vc/lib/libmmal_vc_client.so
/opt/vc/lib/libvcos.so /opt/vc/lib/libbcm_host.so )
```

The changes are almost in each component because of variety of versions and their addreses in RPI memory.

The main changes in source code (except some new #include statements and global variables) are in callback function *video_buffer_callback.*
Face detection (to help face recognition alghorithm) is made by *detectMultiScale* function.
For face recognition, grey pictures are required. Once is getting the I420 frame, it don't need to extract color information, are kept the "py" IplImages (in gray channel) and it is converted into a Mat object.

Interface and applications

The interface is a simply full-screen website. To develop this it was used Electron platform. Electron is a framework for creating native applications with web technologies like JavaScript, HTML, and CSS.
Because of mirror functionality it should not be completely filled with (useless) information. Only the outer corners should be used for content.
Libraries that been used:
Jquery (for DOM manipulation)
Moment.js (for timestamp manipulation)
FeedToJson (for convert RSS feeds into javascript usable JSON data)

Preparation:

To host the interface on webpage is needed an Apache webserver to be running on the Raspberry PI. To install apache is need command:
sudo apt-get install apache2 apache2-doc apache2-utils
PHP scripts is using on webserver:
sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache
Rotating the screen and hide Rainbow colored cube was made by edit */boot/config.txt*
To auto hiding mouse was installed *unclutter* program

For eneble VNC it needed to install x11vnc
To create an auto-start file we need to in ~/.config directory autostart folder in which was made x11vnc.desktop file which contains:
[Desktop Entry]
Encoding=UTF-8
Type=Application
Name=X11VNC
Comment=
Exec=x11vnc -forever -usepw -display :0 -ultrafilexfer
StartupNotify=false
Terminal=false
Hidden=false
For showing pages chromium is using in kiosk mode, to install it:
apt-get install chromium x11-xserver-utils
after installation some configuration need to be change. Disable the screensaver and autoboot in kioskmode by adding few lines in file */etc/xdg/lxsession/LXDE-pi/autostart* :
@xscreensaver -no-splash
@xset s off
@xset -dpms
@xset s noblank
@chromium --kiosk --incognito http://localhost

The interface is control by PM2. PM2 is a production process manager for Node.js applications with a built-in load balancer. It allows you to keep applications alive forever, to reload them without downtime and to facilitate common system admin tasks. In this case we will use it to keep a shell script running. It was installed by:
sudo npm install -g pm2
PM2 is starting with *pm2 startup* commad

For open personalized interface it should be prepared few scripts for execution different interfaces of intelligent mirror (for each person), the interfaces are in different folders because of their different configuration (mostly the same).

When the face recognition system is founding familiar face the right interface is *executed* by command:
pm2 start mateusz.sh
Script mateusz.sh contains folder of intelligent mirror interface and starting command:
cd ~/MagicMirror
DISPLAY=:0 npm start
this script need +x permissions

To control pm2 it can be used commands:
start; save (for restarts after rebooting and save current state*); restart; stop; logs* (to show logs*); show* (to show process information)

Espeak:
System when found property face is saying hello [name] to him by espeak application.
To install espeak it was used command:
sudo apt-get install espeak
that what espeak should say contains external text file. To execute espeak is using command: *system("espeak -f speak.txt -vfr+f5 -s130 2>/dev/null")*

Applications used:
Calendar (allows to share calendar using ICal format)
NOS News Feed (RSS with refreshing news)
Current Wheater
Wheater forecast
Clock
Caption (refreshing inscription)
Alert

Almost each app is personalized for another user. So program is respectively called for each user.
Sample view of program for 2 different persons:

Almost each app is personalized for another user. So program is respectively called for each user.
Sample view of program for 2 different persons:
The calendar is personalized for each person. Each person have another captions (caption are refreshing after 30s and are chosen randomly from prepare base)

Tests:

Face recognition:
with a 320×240 frame with approximately 10 FPS analyses face at each loop.
with a 640×480 frame with approximately 4 FPS with small 1s lag.
Interface was tested and working properly as shown on pictures.

Results

Results of face recognition are very good for a such affordable computer like Raspberry Pi, for real-time use like RC robot it is too slow (This call requires most of the cpu needed in a loop because of face detection)
Interface of Smart mirror is done and working correctly. In very easy way can be changed or improved.

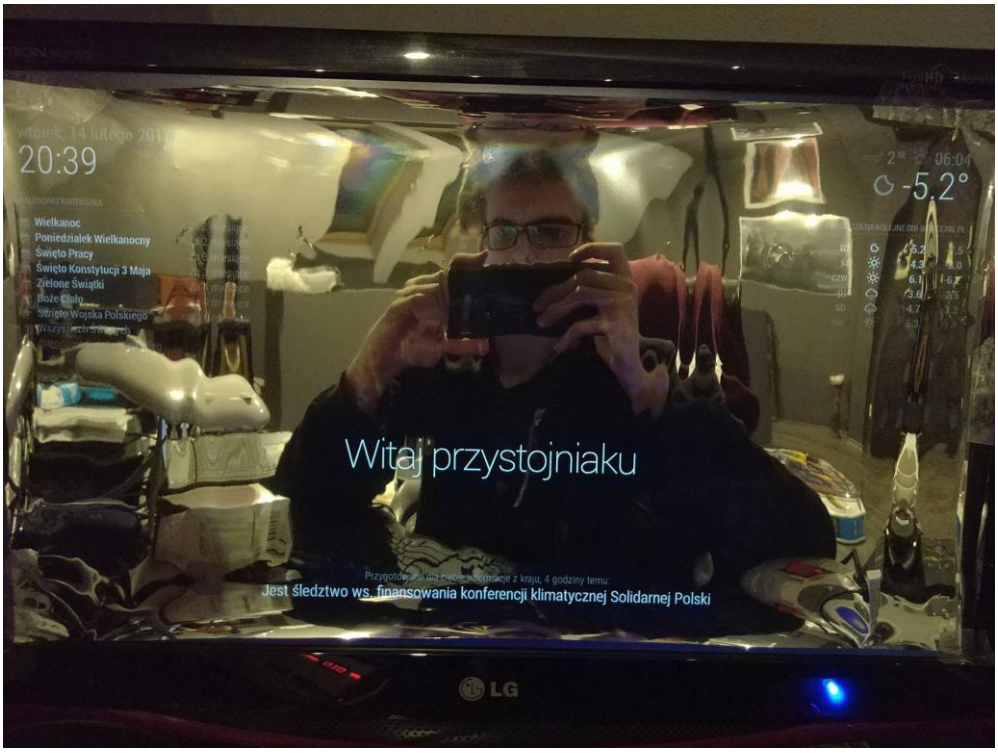


Photo don’t looks impressive because of defect main monitor, for the purposes of report the two way mirror film was peeled away from defected monitor, and attach provisionally to another monitor.