$=$ QuidEst $=$
Real-Time Monocular Depth Map to Audio Signal Conversion Algorithm

---

User's Guide

HTTPS://GITHUB.COM/CANESSAE/QUIDEST

About

The idea behind 'QuidEst' is to develop an algorithm that associates *'spatial sounds'* to specific areas within a video frame in order to locate in real-time the presence of (moving) obstacles from the surroundings (e.g., while walking or cycling) to then send an alarm according to its proximity.

This association is done through the analysis of (the nearest white parts of) the 255 gray scale depth map –divided in depth labels or `--dm` number of layers, which is extracted from a single USB webcam or cellular phone camera connected to a PC Linux.
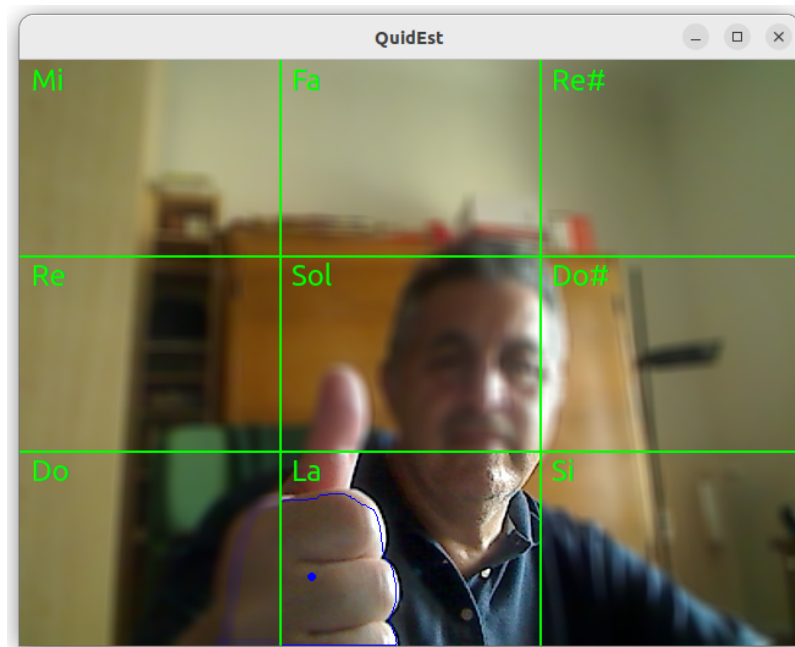

Fig.1: `QuidEst` allows to associate in real-time 9 musical notes to 9 specific areas in a video frame in order to send stereo acustic warnings due to the presence of (front or rear) moving obstacles in the surroundings.

The generation of spatial sounds is related to nine visible regions within a video frame (*i.e.*: top, bottom, left, right, center, bottom-left, bottom-right, top-left, top-right) as shown in Figure 1. Specific musical notes such as Do, Re, Mi, $\cdots$ will ring up when the nearest objects or solid Quids[1] are approaching from the (front or rear) surroundings. The 3 different notes on the left of the screen will ring on the left channel of the stereo speakers or headphones. The 3 notes on the center will ring as mono, and the 3 notes on the right of the screen will ring on the right channel.

The associated volume of a sound increases in relation to the closest area being identified. This is, in theory, proportional to at how near is the object from the

---

[1] *from Latin: 'quid = what is?'*

webcam (or from the person wearing the webcam or from the cellular phone attached to its body). On the screen, the blue areas around a centroid in the depth proximity correspond to the nearest objects, whereas the farthest objects appears blurred and do not deliver any sound as shown in Figure 1.

This novel class of depth map to sonar-like conversion system may be useful as a single webcam-based guidance system for additional vision assistance. The implementation of `QuidEst` may help to find better live outdoor paths for blind and visually impaired people through a safer forward walk, or it may be useful to have a simple system for rear vision awareness for all.

In brief, `QuidEst` is an extended C-based application with the aim of minimizing image processing time to approach real-time practical applications for personal safety. It runs under Linux O.S. It uses the MiDaS 2.1 deep neural network (DNN) of the OpenCV algorithm for a robust monocular depth map estimation in real-time[2], to then associate the nearest depth map areas framed around the PC screen into a specific musical note each. This latter construction is the main difference with respect to most *virtual Theremin* music instruments or acusting scene systems using webcams as available on the web.

1.1   Requirements

The minimum hardware needed to test `QuidEst` is any standard PC Computer *e.g.*, Intel Core i5, 64bit and at least 4G RAM, running a recent release of Linux O.S. (23.04LT or newer) having an internal webcam.

Optional hardware that can be used are an external USB webcam, audio stereo speakers, or headphones and a smartphone with the free DroidCam app installed (as shown below).

1.2   Copyright, Credits and Contacts

©Permission to use, copy, and distribute the `QuidEst` code and its documentation for educational purposes ONLY, and without any fee, is hereby granted provided that reference to https://github.com/canessae/Quidest appears in all distributed copies and in any other supporting documentation. This application is provided 'as is' without any express or implied warranty.

For further information, or to report Bugs, please visit:
https://github.com/canessae/Quidest

---

[2]https://github.com/isl-org/MiDaS

Install

The latest version of the Debian 'QuidEst-x.x.x-Linux.deb' package can be downloaded from https://github.com/canessae/Quidest/releases
For example, you can download the version QuidEst-0.1.0-Linux.deb (60 MB) typing

```
wget https://github.com/canessae/Quidest/releases/
                download/0.1.0/QuidEst-0.1.0-Linux.deb
```

2.1   Dependencies

It is necessary to install first some extra packages and their dependencies. The following packages (and their dependencies) are needed to be installed before hand:

> libopencv-videoio4.5d, libopencv-imgproc4.5d, libopencv-core4.5d, libopencv-imgcodecs4.5d, libopencv-contrib4.5d, libopencv-calib3d4.5d, libopencv-imgcodecs4.5d, libopencv-video4.5d, libopencv-flann4.5d, libopencv-dnn4.5d, libqt5widgets5, libqt5gui5, libqt5core5a, libhidapi-libusb0, libqt5multimedia5-plugins, libqt5multimedia5

The updated list of required packages can be found as

```
dpkg -I QuidEst-0.1.0-Linux.deb
```

To install the packages listed above, issue the command:
'sudo apt-get install <pkg1> <pkg2> ···' and so on. For example,

```
sudo apt-get install libopencv-videoio4.5d libopencv-imgproc4.5d
                ...    ...
```

2.2   Library Install

To install the QuidEst '(.deb')' package in the '/opt' directory type the command

```
sudo dpkg -i QuidEst-0.1.0-Linux.deb
```

In case some 'warning: files list file for package ···' may appear on the screen after having issued 'dpkg', then type

```
sudo apt-get install -f
```

which install what is missing to 'dpkg'.

NOTE: The `QuidEst` **working path is**

```
cd /opt/quidest/bin/
```

2.3   Uninstall

In order to remove the `QuidEst` ('.deb') package type

```
sudo dpkg -r quidest
```

QuidEst Commands

To start using the `QuidEst` line command, some important comments are necessary.

The MiDaS 2.1 '*small*' network (included in the directory '`/opt/quidest/models/`') cannot provide complete depth information on distant regions, however the generation of depth map starting from a single RGB image (or video frame) can still give reasonable results for real-time applications such as `QuidEst`.
To improve details on the computed depth map using the option '`-l`', one could use MIDaS 2.1 large model: '`model-f6b98070.onnx`'[3] (397 MB) and to add it to the directory '`models/`'.[4]. However the video responses will become excessively slow.
In general, the `QuidEst` line command can be used as follows:

```
cd /opt/quidest/bin
./quidest [options] videoDev
```

where

```
Options:
-h, --help          Displays help on command line options
--help-all          Displays help including Qt specific options
-n, --nmodel <Path to .onnx file>      AI network model file path
-l, --large         Use large MiDaS network
-f, --fading        Enable audio fading mode
--dm, --depthmap <Integer Nr levels>      Set depth levels (default 20)


Arguments:
videoDev          Video device number (integer)
```

3.1   Examples

- Input: Internal webcam (video device '`/dev/video0`'), MiDaS small and default Nr of depth levels

```
cd /opt/quidest/bin
./quidest -n ../models/model-small.onnx 0
```

Press '`q`' to quit. To invert the video image press '`i`'.

---

[3]ONNX: Open Neural Network Exchange is an open standard to represent machine learning models

[4]Use, *e.g.*, '`wget https://github.com/isl-org/MiDaS/releases/download/v2_1/model-f6b98070.onnx`'

- Input: external USB webcam (video device '`/dev/video1`'), MiDaS small and Nr depth levels=60

```
cd /opt/quidest/bin
./quidest -n ../models/model-small.onnx --dm 60 1
```

Press 'q' to quit. To invert the video image press 'i'.

- Input: external USB webcam (video device '`/dev/video2`'), MiDaS small, Nr depth levels=40 and Enable audio fading mode

```
cd /opt/quidest/bin
./quidest -n ../models/model-small.onnx --dm 40 -f 2
```

Press 'q' to quit. To invert the video image press 'i'.

3.2 Using Cellphone-as-Webcam

It is also possible to use a cellular phone camera connected to your PC Linux using DroidCam and the output will look like in Figure 2.
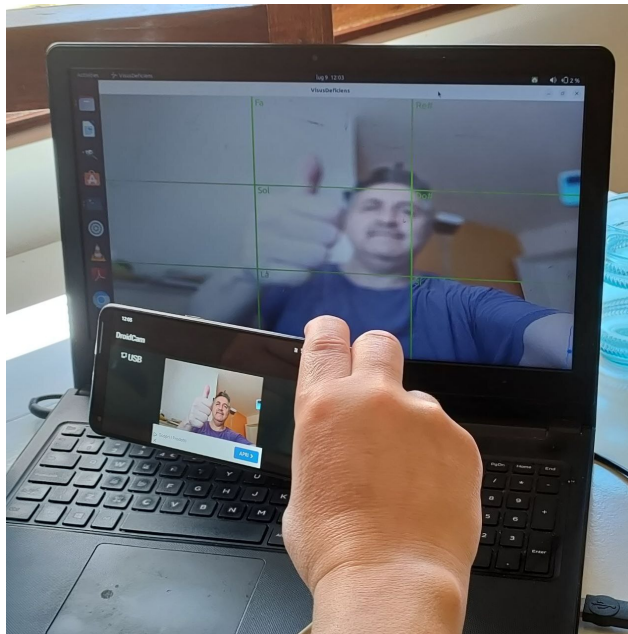

Fig.2: `QuidEst` in action using Cellphone-as-Webcam.

DroidCam turns a cellphone or tablet into a webcam for your PC. Since 2010 is one of the most popular original phone-as-webcam app. It is freely available from https://www.dev47apps.com.

Install first DroidCam app. in your smartphone and activate "USB debugging" by following instructions in the above website or search Google on how to set up on your particular phone brand. On the other hand, install DroidCam Client for linux O.S. in your PC, issue the command line: '`droidcam`' and then activate the connection with an USB cable to your phone webcam as in Figure 3.
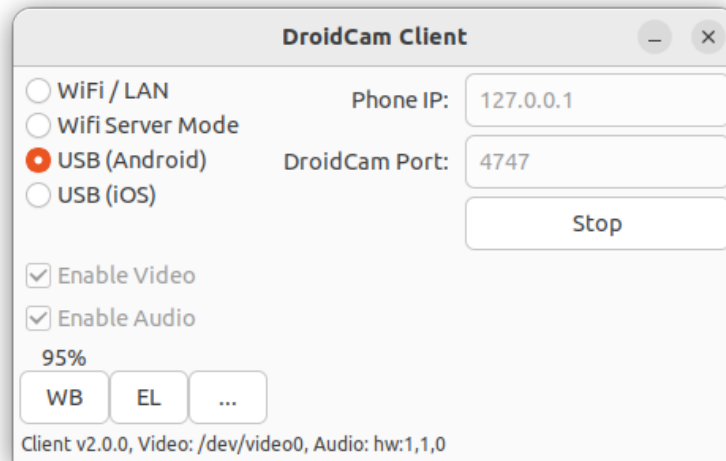
Fig.3: DroidCam Client for linux O.S.

It is then also possible to have as

- Input: Cellphone-as-Webcam, MiDaS small, default Nr of depth levels and audio fading mode enabled

```
cd /opt/quidest/bin
./quidest -n ../models/model-small.onnx -f 2
```

Check the video devices in your PC: 'ls /dev/video*' and identify the one used by DroidCam. In the example above video device selected is '/dev/video2'.
Press 'q' to quit. To invert the video image press 'i'.