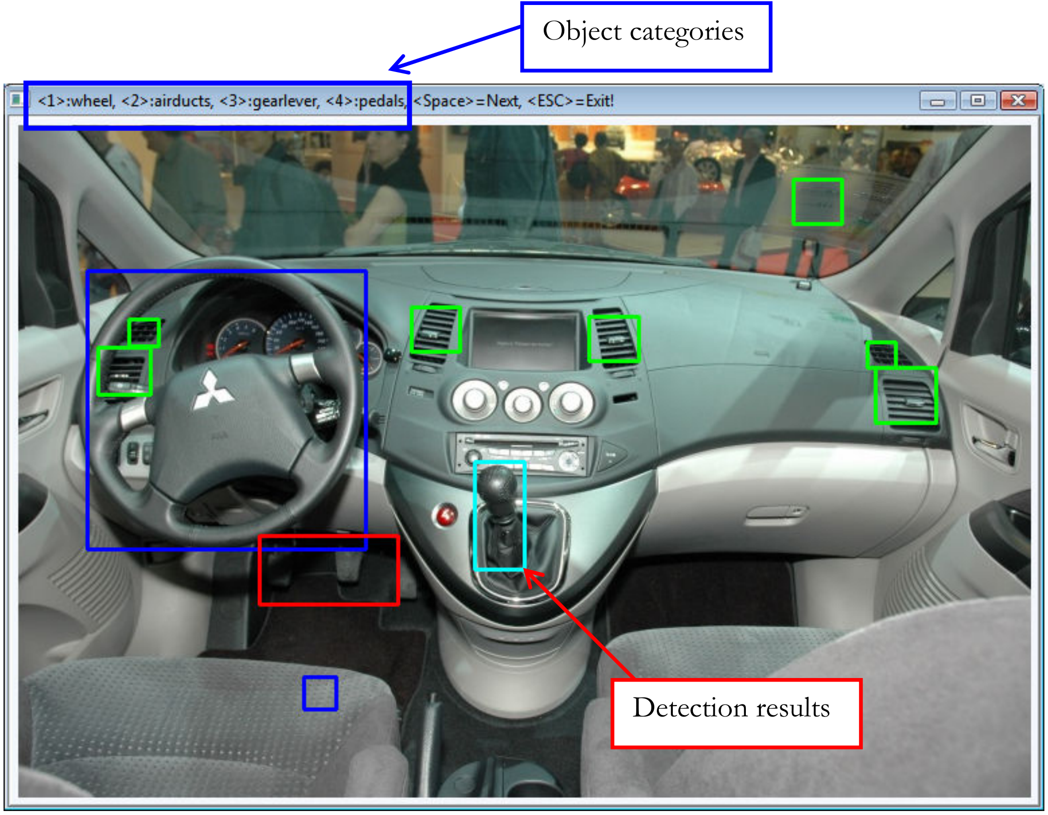
Annotator

A semi-Automatic Image Annotator

1. Program Description

Annotator is an image annotation tool that supports semi-automatic annotation. It was created using the C++ programming language and the OpenCV library [OpenCV]. OpenCV was used for opening and displaying images, object detection and user input handling. Annotator supports both manual and semi-automatic annotation. In order to work in semi-automatic mode, user has to provide object detectors created with OpenCV. These object detectors are based on the Viola & Jones detection method [Viola].



<1>:wheel, <2>:airducts, <3>:gearlever, <4>:pedals, <Space>=Next, <ESC>=Exit!

Fig. 1: User interface.

2. System Requirements

System must be running Microsoft Windows and have at least 16MB of RAM available for the program. The program uses the OpenCV library v1.0. All necessary library files are given in the download pack.

3. Invoking Annotator

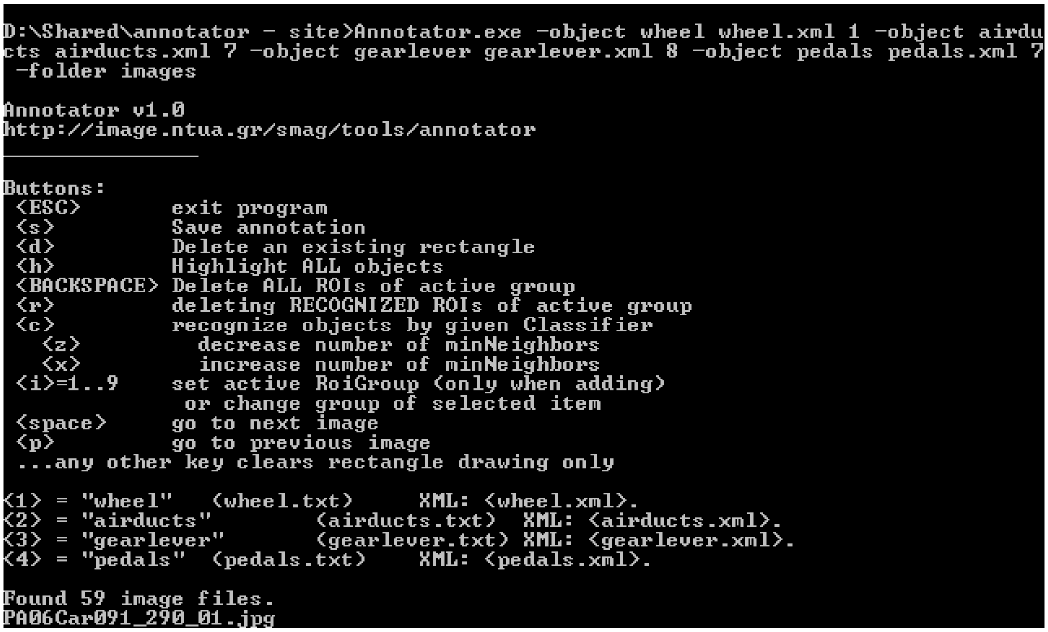
No installation is required to use Annotator. To call Annotator, the user must follow the syntax described bellow.

```
Annotator.exe
   -object object_category_name1 [detector1 [threshold1]]
   [-object object_category_name 2 [detector2 [threshold2]]] […]
   -folder folder_name
```

```
 -object object_category_name  Name of each object category. The
                               annotation is saved in a text file
                               with the category’s name.

                   [detector]  In order to work in semi-automatic
                               mode, the user has to provide the
                               OpenCV detector’s XML file.

                  [threshold]  Threshold value for filtering the
                               results of the detection process.
                               Default value is 3.
```



```
D:\Shared\annotator - site>Annotator.exe -object wheel wheel.xml 1 -object airdu
cts airducts.xml 7 -object gearlever gearlever.xml 8 -object pedals pedals.xml 7
 -folder images

Annotator v1.0
http://image.ntua.gr/smag/tools/annotator
_______________

Buttons:
 <ESC>       exit program
 <s>         Save annotation
 <d>         Delete an existing rectangle
 <h>         Highlight ALL objects
 <BACKSPACE> Delete ALL ROIs of active group
 <r>         deleting RECOGNIZED ROIs of active group
 <c>         recognize objects by given Classifier
   <z>         decrease number of minNeighbors
   <x>         increase number of minNeighbors
 <i>=1..9    set active RoiGroup (only when adding)
              or change group of selected item
 <space>     go to next image
 <p>         go to previous image
 ...any other key clears rectangle drawing only

<1> = "wheel"   (wheel.txt)     XML: <wheel.xml>.
<2> = "airducts"        (airducts.txt)  XML: <airducts.xml>.
<3> = "gearlever"       (gearlever.txt) XML: <gearlever.xml>.
<4> = "pedals"  (pedals.txt)    XML: <pedals.xml>.

Found 59 image files.
PA06Car091_290_01.jpg
```

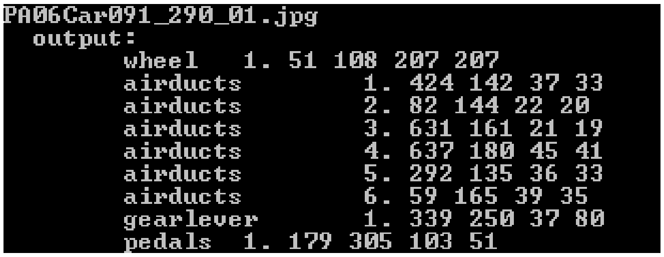
Fig. 2: Invoking Annotator.

4. Working with Annotator

Annotator opens all image files from the given folder, supporting many image formats. The tool starts by opening and displaying the first image in the folder. The user simply draws bounding boxes for every object depicted in the image. All bounding boxes can be moved, resized, or deleted. The object category to which a bounding box belongs, can change by simply selecting the bounding box and change the active object category by pressing the corresponding key. In semi-automatic mode, when an image is loaded for the first time, object detection algorithm creates bounding boxes for all detected objects in the image. The results of the detection can change by adapting the detection's threshold value.

The annotation is saved in text files, named after the corresponding category name. The user can save anytime, and when exiting the program, the user will be asked to save the annotation in the files. If the filenames already exist when starting the program, any already saved annotation will be loaded and the bounding boxes will be drawn in the images. That way the user can break the annotation task in smaller parts, and continue without changing the images or the annotation files.

When starting the program, the user can see the key definitions and the object categories that will be available for the annotation. The annotation results are displayed in the command window when moving to the next image. One example is shown in Fig. 3.



```
PA06Car091_290_01.jpg
  output:
        wheel   1. 51 108 207 207
        airducts        1. 424 142 37 33
        airducts        2. 82 144 22 20
        airducts        3. 631 161 21 19
        airducts        4. 637 180 45 41
        airducts        5. 292 135 36 33
        airducts        6. 59 165 39 35
        gearlever       1. 339 250 37 80
        pedals  1. 179 305 103 51
```

Fig. 3 : Command prompt output.

5. Keyboard functionality

The detailed keyboard functionality is explained below and shown in Fig. 4.

<ESC> Program exit. Before exiting user is asked to save all annotation files.

<s> Save annotation in output files.

<d> Delete selected object.

<h> Highlight all objects.

<Backspace> Delete ALL objects of the selected category. User is asked to confirm the selection.

<r> Delete all objects that were detected automatically. Objects that were detected automatically but were afterwards edited by the user in any way are not deleted.

<c> Detect objects for the selected object category.

<z> <x> Decrease/Increase the detection threshold. Detection results change accordingly.

<i>=1..9 Set the active object category. If an object is already selected, it changes category to the new active category.

<space> <p> Next/Previous image in folder.

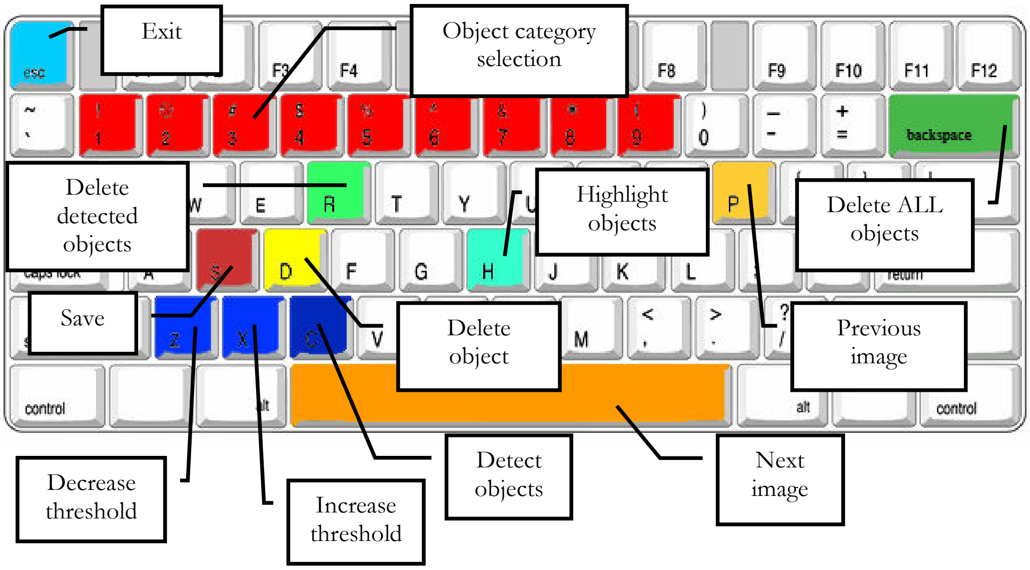


Fig. 4 : Keyboard functions.

6. Image annotation files

Annotation of each object category is saved in text files named by the category name. Each line of a text file contains all annotations for a particular image and object category. The annotation format is as follows:

```
filename  number_of_objects  x1  y1  width1  height1  x2  y2  width2
          height2 …
```

The first info is the filename of the image that contains the objects. The second info is the number of objects of the particular category that belong to the image. After that, there are groups of four numbers for every object. In these groups, the first number is the horizontal coordinate of the upper left point of the object, the second the vertical coordinate of the same point, the third the width of the object and the fourth number is the height of the object. Some examples of this annotation format are depicted in Fig. 5.

```
TestImages/test-0.pgm 1 28 54 93 29
TestImages/test-1.pgm 2 30 69 83 29 138 69 101 31
TestImages/test-10.pgm 2 11 52 72 26 113 51 75 26
TestImages/test-100.pgm 1 12 38 83 26
TestImages/test-101.pgm 1 19 26 84 24
TestImages/test-104.pgm 2 134 60 75 25 17 53 85 31
TestImages/test-105.pgm 2 145 72 81 28 45 70 82 25
TestImages/test-106.pgm 1 78 59 83 26
TestImages/test-107.pgm 1 28 50 86 24
...
```

Fig. 5 : Annotation format example.

7. Contact

Christos Varytimidis

http://image.ntua.gr/smag/people/Christos_Varytimidis
e-mail: chrisvar@image.ntua.gr
Tel: +30 210 7722491
Fax: +30 210 7722492
Address: Iroon Polytexneiou 9, 15780 Zografou, Greece