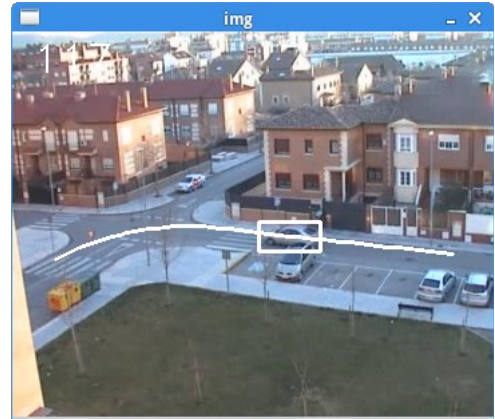


TrATVid

Introduction:

TrATVid (Trajectory Annotation Tool for Video) is a python3 application that allows easy annotation of object trajectories in video. Annotated trajectories can be composed of both Bounding Box rectangles or Points. While it allows definition of nodes for given key frames, the rest of nodes are interpolated using a novel bounding box interpolation method, based on 3D reconstruction and B-Splines interpolation.

TrATVid allows you to save your work on standard xml files. Furthermore, a useful tool included is the possibility to extract image blobs for all the frames of a given trajectory, that can be used, for instance, for classification training or test purposes.



For any doubt or suggestions about *TrATVid*, please contact me at: pedro.gil@uah.es

Installation guide:

Download *TrATVid.zip* file and extract it in your PC. This file includes the python3 files, a sample *settings.xml* file (for program settings) and a video sample (including one annotated trajectory). In order to run the program, the following software is required:

- Python 3 (already installed in most linux distributions).
- OpenCV library.
- SciPy library.

1. Linux:

For Ubuntu and Debian users, open a terminal and type:

```
sudo apt-get install python3-opencv python3-scipy
```

On other distributions, analogous packages should be available.

2. Windows:

Python installation: We recommend installing a trusted Python distribution (Anaconda, WinPython,...). To install WinPython (<http://winpython.sourceforge.net/>) download the appropriate software at

http://sourceforge.net/projects/winpython/files/WinPython_2.7/2.7.9.2/

After extracting the distribution, you have to follow instructions at

<http://winpython.sourceforge.net/> page to complete installation. Pay special attention to the distribution registry instructions.

In some cases, you may need to install some extra packages, as Python-enum. WinPython includes WinPython Package Manager in order to easily install or uninstall any package.

OpenCV installation: For a proper installation, check instructions in “*Installation by Using the Pre-built Libraries*” paragraph at

TrATVid

http://docs.opencv.org/doc/tutorials/introduction/windows_install/windows_install.html

Follow the procedure in “*Set the OpenCV environment variable and add it to the systems path*”. Afterwards, add the “<opencvpath>\source\3dparty\ffmpeg” directory to the environment variable PATH. This is required by the OpenCV library to work with videos properly.

OpenCV-Python installation: Follow instructions in paragraph “*Installing OpenCV from prebuilt binaries*”, items 7 and 8, at

http://docs.opencv.org/trunk/doc/py_tutorials/py_setup/py_setup_in_windows/py_setup_in_windows.html#install-opencv-python-in-windows

Upon completion of all the instructions given above, you should be able to run the program with the default settings. In this case, the sample video provided should be displayed, and the annotated sample trajectory shown as you move along the video (keys 'x' and 'z').



User's guide:

Settings:

The program can be configured through a xml file, which may be provided as the first argument of the program. If no argument is provided, the program will try to open the file *settings.xml* in the same directory as the program. You can use and edit the XML file provided (*settings.xml*). This file must include the following tags:

- **<video path="/path/to/video"/>**
 - Path where the video is located (including last '/'). Furthermore, it is the directory where the annotation files will be saved.
- **<video name="video.avi">**
 - Name of the video to be annotated. Please, use video formats and containers compatible with the OpenCV library (www.opencv.org). MP4 or MJPEG formats are recommended.
- **<file name="video.xml">**
 - Name of the file where the annotations will be saved. If the file does not exist, a new file is created. If the file exists, the annotations on this file will be loaded at program start, and overwritten at saving.
- **<backup time="300">**
 - To prevent data lost, the program performs a periodic back up, with a period equal to that indicated in this tag (in seconds). Data is written over a file with the same name on the same directory, but with the extension "bak" appended.
- **<interpolation type="GC">**
 - This tag indicates the default type of interpolation between key frames on new trajectories. Interpolation type can be changed once the trajectory is created (see below). Available interpolators are:
 - *NI*: No interpolation (node coordinates are copied between frames)
 - *LI*: Linear interpolation on 2D
 - *CS*: Cubic B-Spline interpolation on 2D
 - *GI*: Linear interpolation on 3D using Geometric 3D reconstruction
 - *GC*: Combined Cubic B-Spline interpolation and Geometric 3D reconstruction
- **<interpolation margin="5">**
 - Margin, in pixels, used in blob extractions (see below).

Program usage:

Once the file *settings.xml* has been correctly configured, you can run the program by opening a terminal and typing (Linux or Windows):

```
$ python3 TrATVid.pyc
```

On the other hand, an alternative settings file can be provided by including it in the program arguments:

```
$ python3 TrATVid.pyc sett.xml
```

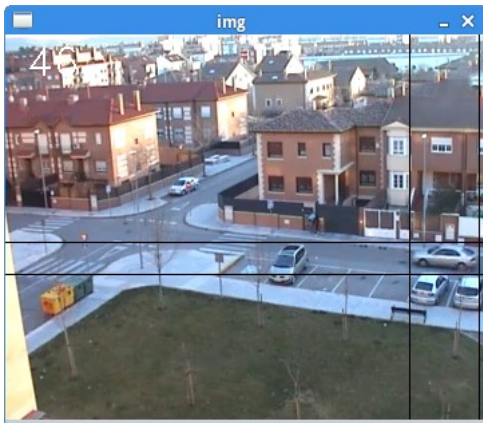
In any case, if the annotation file provided in the settings is new, you will be presented with the first frame of the video. Using 'x' and 'z' keys, you can move the video forwards/backwards, one frame at a time. Pressing 'q' key (*quit*) at any time will let you close the program.

Creating a trajectory:

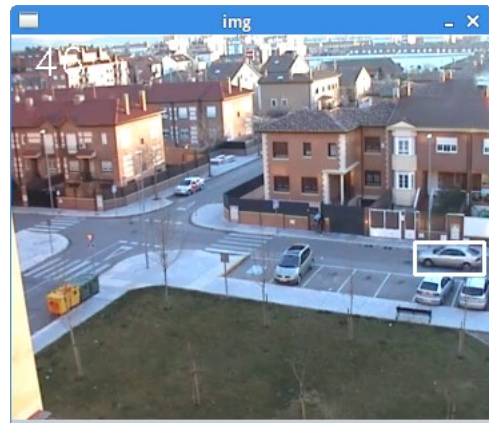
To add a trajectory, find, using 'x' and 'z' keys, the frame where you want the annotation to start.

TrATVid

Press the 'a' key (*add*), and the cursor will change to a cross, to help you locate the object. Clicking on the image will create the first node of a *point trajectory*. Dragging over the image, on the other hand, will create the first node of a *rectangle trajectory*. Once the node has been drawn, you can further edit it¹. Dragging inside the rectangle will move the whole rectangle, whereas dragging outside it will move the corresponding edge, and similarly for points. Note that while editing the node, you can not change the video frame. Pressing the 's' key (*save*) will save the node and create the new trajectory. At this point, you can press 's' again to save the new trajectory, or continue editing the trajectory. You can press *Esc* at any point to cancel the process.



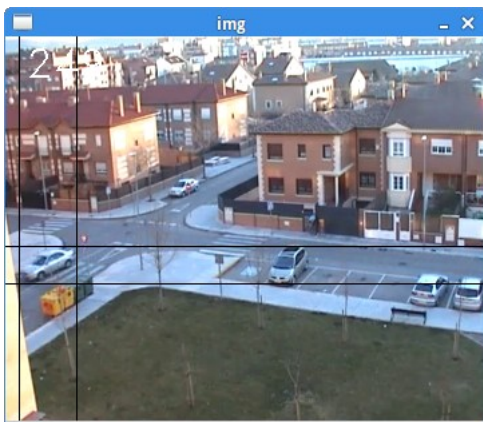
Creating a new rectangle node



Saving a new rectangle node

Editing a trajectory:

To add a second node to the trajectory, use 'x' and 'z' keys to place the video on the frame where you want to add it. Pressing 'a' again, will add a second node, that can be edited to place it in its proper location. Pressing 's' will add this node to the trajectory. Moving along the video, you can check how the intermediate positions of the trajectory are interpolated, using the corresponding interpolation method. Following the same procedure, you can add as many nodes as you need.



Adding a second node to the trajectory



Saving the second node of the trajectory

Furthermore, if you need to edit a node already added, just use 'a' key again, that is, 'a' is used both for adding and editing a node. To delete a node, first place the video on the frame where it is the

¹ Some issues related with OpenCV configuration can result in unexpected behavior on Windows systems. If that happens, just clicking the right mouse button before releasing left button will make the rectangle.

TrATVid

node you want to delete. Press 'a' to edit the node, and then press 'd' (*delete*) to delete it. A confirmation message will appear on the terminal. Press 'y' (*yes*) to confirm, or any other key to cancel. Note that if you pressed 'd' before editing the node by pressing 'a', the whole trajectory would be deleted instead.

After editing a trajectory, do not forget to press 's' again to save the trajectory, or *Esc* to cancel the whole edition.

Working with trajectories:

To select a trajectory, double-click inside the node (or near the point for *point trajectories*), and the trajectory will be selected for edition. After selecting a trajectory, the rest of visible trajectories (if any) disappear, and the path of the selected one will turn black. At this point, we can add/delete/edit trajectory nodes, as described before, or delete the trajectory, pressing the 'd' key. For this last step, you need to press 'y' to confirm, or any other key to cancel.

Save to a file:

Press the 'w' key (*write*) to save your work to the xml file set in the *settings.xml* file. If you are currently editing a trajectory, you will be prompted to save your work before write it to the file. If the file exists, it will be overwritten, and the next time you start the program, the saved annotations will be loaded. Trajectories are stored sorted by starting time.

Change interpolation type:

Although the “combined B-Spline interpolation and 3D reconstruction” method is recommended for interpolation between key frames, the rest of available interpolation methods can be used instead. To change the interpolation method for a given trajectory, first select it by double-click inside the node. After that, pressing keys '1' to '5' will change the interpolation method, according to:

- '1': No interpolation (node coordinates are copied between frames)
- '2': Linear interpolation on 2D
- '3': Cubic B-Spline interpolation on 2D
- '4': Linear interpolation on 3D using geometric 3D reconstruction
- '5': Combined Cubic B-Spline interpolation and geometric 3D reconstruction

After changing the interpolation type, you need to save your change (key 's').

Getting results:

- **Saving complete trajectories:** *TrATVid* allows you to annotate trajectories using a sparse set of key frames, that is, you do not need to annotate all the frames of the trajectory, but only a reduced set, while the rest of nodes are obtained by interpolation. However, if you need all the node coordinates for a trajectory, regardless they are actual or interpolated nodes, you can press the 'o' key (*output*). A file, with the same structure that the annotation file, but with the coordinates for all the nodes, not only the actual ones, is generated. The name of the file will be the same than the annotation one, with the suffix “co_” added to the file name.
- **Saving an enhanced video:** Once a video has been annotated, a copy of the video, with all the trajectories superimposed, can be generated, pressing 'v' key (*video*). The new video will be saved as a sequence of images in the directory */path/video/* where *path* is the project path indicated in the settings file.
- **Saving trajectory blobs:** *TrATVid* allows you to store an image blob for each of the nodes

TrATVid

of a given trajectory, both for actual and interpolated nodes. For rectangular nodes, the image is obtained from the rectangle of the node, enlarging the margin, in pixels, by the size indicated in the setting file, in the tag *interpolate/margin*. For point nodes, the image is a square of size *margin*, centered on the coordinates of the node. To get all the blobs of a given trajectory, select it by double-clicking inside the rectangle (or near the point), and press 'p' key (*print*). The program will save all the blobs of the trajectory in png format in the directory '/ID/', where ID is the ID automatically assigned to the trajectory (if the trajectory is new, ID will be 0, and will eventually overwrite previous trajectory blobs). If no trajectory is selected, the program will save all the trajectories. All the data will be saved on the video path indicated in the settings file.

