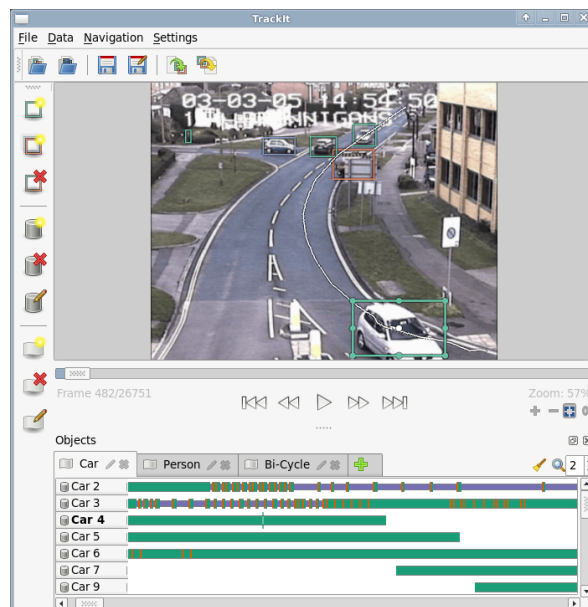


Softwarepraktikum

TrackIt

–

Video Ground Truth Editor



Programming and Documentation: Sven Klingel
(klingsn@studi.informatik.uni-stuttgart.de)

Programming and Documentation: Hanspeter Haegele
(haegelhr@studi.informatik.uni-stuttgart.de)

Supervision: Kuno Kurzhals:
(kuno.kurzhals@vis.uni-stuttgart.de)

Contents

1 The Program

1.1 Introduction

TrackIt is a program to enable users to easily apply bounding boxes (henceforth abbreviated with *bboxes*) on video files to create ground truth data for those files. Videos are opened with the use of the OpenCV framework, painting Bboxes is done via the QT Framework and OpenGL. Keyboard shortcuts assist in efficient working and data manipulation. All used file-formats are well-documented. We hope we created an easy-to-use, reliable and robust program.

2 Using TrackIt

The main purpose of TrackIt is to assist the user in manually generating bounding box data for videos to supply tracking information for other programs, create ground truth for visual studies or similar purposes. Here comes a step-by-step introduction and overview, showing all steps necessary to create tracking data by hand. You can move and rearrange all widgets freely, not limited to inside the main windows, which is especially useful in dual-screen setups.

2.1 Starting TrackIt

Under Linux we recommend starting TrackIt from a terminal, as the underlying OpenCV framework and video codecs might dump useful information to standard output.

Under Windows, just doubleclick the executable TrackIt.exe.

2.2 Opening Video Files

Most likely you'll start with a video file containing some features you want to track manually: Press „File → Open Video“ and a file dialog window will appear. Navigate to your video file and press open. You will be provided with a standard category and object, so you can just start creating Bboxes.

Tip (Linux): FFMpeg, the underlying video decoder used by TrackIt might in some cases dump information about wrongly encoded video files to console. Some of them may result in your tracking results to be not in sync with the actual video data. Hence we recommend starting TrackIt in a terminal.

2.3 Navigating in Video Files in the Video Window

The controls in the lower middle are for navigating. Pressing the *Play/Pause*-button will make the program play continuously at the framerate of the video, the "*previous frame*" and "*next frame*" buttons will always just play one frame forwards or backwards. Contrary to this, the "*Next Keyframe*"-button will jump to the next keyframe of the currently selected object (yet

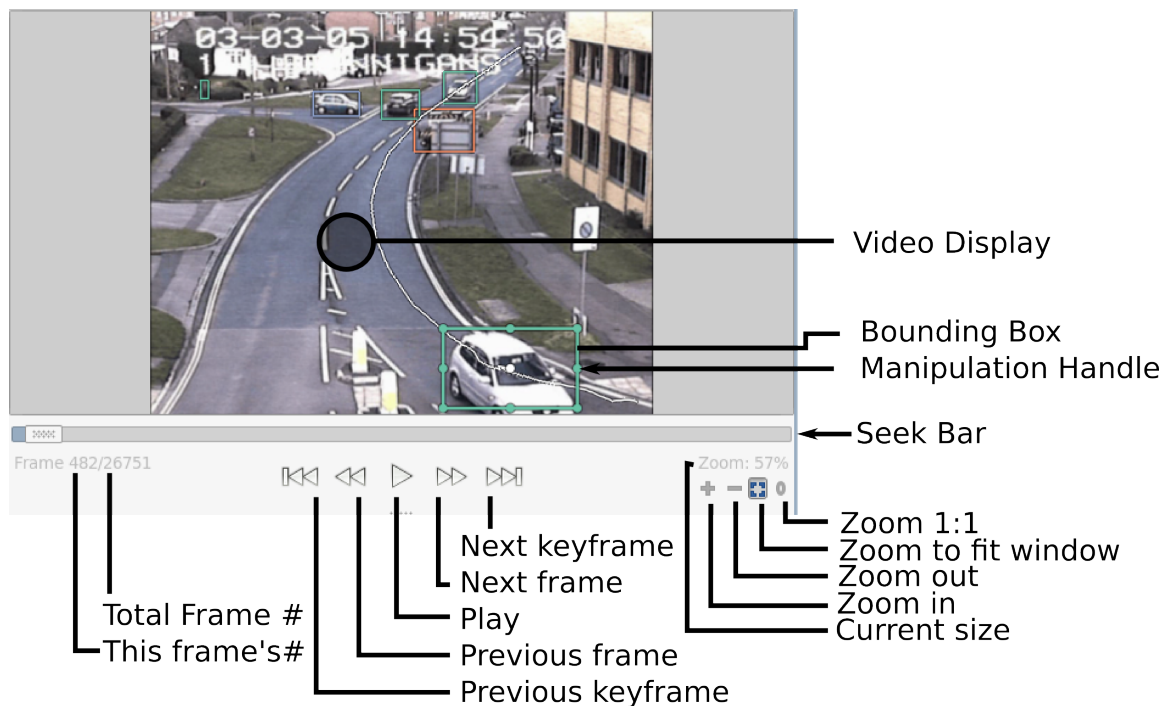


Figure 1: Video window

only if there are actually keyframes before or after).

You can directly jump to any part of the video via clicking on any point of the *seek bar*, or grabbing the slider and dragging it to the desired position. The Framecount on the lower left meanwhile indicates the framenummer and the total number of frames.

The lower right section of the Video window contains the zoom buttons to control the video's size. Enlarge the video size with the "+" button, shrink it with the - button. The other two buttons control fitting to window dimension size and 1:1, meaning that one pixel in the video is one pixel on the screen.

2.4 Opening & importing Tracking Data Files

TrackIt supports various types of Tracking data, ViPER being one of the most commonly used ones. To open a Data file, click on "File → Open" and choose your .btd file.

If you want to open files of other file types, such as .bb or a ViPER .xml file, you must use "File → Import". After opening a data file, TrackIt will automatically look for a video file with the same file name or a video file name specified in the meta data of a ViPER file and try to open it.

2.5 Data window

The *data window* contains a view of all the data that has already been created or loaded. At first, let's focus on the general handling of this window. As shown in figure

On the top there are the *Category Tabs*, which allows you to organize your objects. Each tab contains (from left to right):

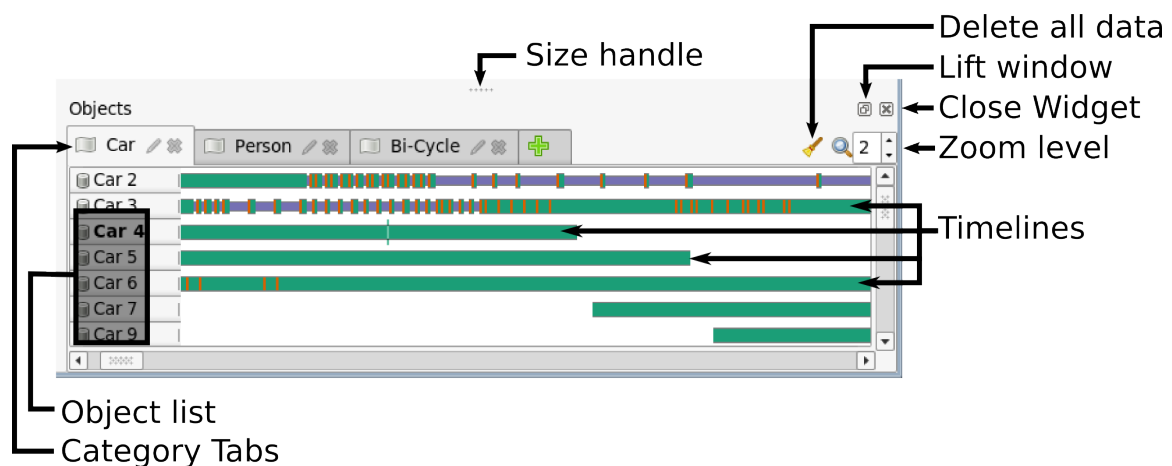


Figure 2: Data window

Flag-symbol Indicates it's a group.

Category name All objects in this Category will carry this Name, along with a number.

Pen Edit the Name of the category.

X Delete Category and all it's content (You will be asked to confirm this step).

Left hand side there are the labels of objects. Directly right of each label there's the timeline. To transfer an object to another category, right-click the object's label or timeline, select "Object.. → Move object" and select the category.

A timeline shows the bboxes of each object, sorted by frame number. Figure

2.5.1 Types of BBoxes

There is not only one kind of bounding Boxes, but several which serve different purposes.

Single BBox The most usual bboxes are „single-frame“ bboxes, valid for one frame, like the one you encountered in the example above, they are usually depicted in green.

Key BBox Key Bboxes are a concept to make it easier tracking an object's linear movement. It will be explained thoroughly in section

2.6 Creating a Key Bbox and interpolated „virtual“ frames

If an object moves with a constant direction and speed over several frames, you can save time by using linear interpolation between bounding boxes of the same object. So let's assume you have an object like a car passing through your video from left to right at a constant speed. Instead of generating a *bbox* for each frame you scroll to the end of the constant motion of this object. By clicking the orange rectangle in the *action bar*, a bounding box appears at the original place, and you can fit it to the position (and size) of the moving object. In the datawindow, you can then see the (now selected) orange

keybox, and left of it, the violet "*virtual bboxes*".

Play the scene again, and you will see that the last box will transform in position and size along a path from the last created *bbox* towards the created *keybbox*. You can not directly edit a virtual keybox, but by selecting it in the data (or, of course, video) window, and pressing the "*convert keybox*" (former *create keybox*) button, you can convert it to an editable type of bounding box while preserving linear interpolation before and after.

Tip: Object is not transforming and not moving, but still there? "Not moving" is a constant movement at zero speed, so you can just create a Key-BBox.

2.7 Objects and groups

Each object has exactly one *bbox* per frame. To track more than one video object in one frame you have to create a new data object by clicking on the "create object" button in the action bar, or by using the keyboard shortcut "Ctrl+F".

However, with many objects the overview might get cluttered, so grouping of similar objects is recommended, also group membership leads to more semantic information for other programs: For example, in a ball game, putting all players of team "blue" in one group and all "red" players in another.

To move an object into another group, right-click on the object, select "Object... → Move object". To switch categories use the "Q" and "E" keys.

2.8 Opening/Saving and importing/exporting data files

To save a file, press "File → Save data" or "File → Save data as...". Give a filename, and choose your data format from the selector and click okay. Now let's explain the very important distinction between opening/saving and importing/exporting: Opening and saving will always preserve all features TrackIt has. However, importing and exporting will always come along with a loss of meta-information, such as group membership or group name, yet no bounding box data will be lost. For more information see

2.9 Keyboard controls for efficiency

Using Keyboard controls can greatly improve your working efficiency. We'll just show some basic usage, a table with all available keys and secondary keyboard layouts can be found in tables of

A Appendix A

A.1 File Formats

TrackIt supports a the following file formats currently supported file Formats include: .bb, .btd, ViPER .xml However, not all file formats support all features of the other file formats or our program, loss of grouping and naming information may occur. however, there is never any loss of Bbox information. This table shows which file formats support which features, the features being listed as:

Categories One or more objects may be put together in a category tab, to simplyfy grouping objects.

Holes Allows objects to have frames with no bounding boxes between two frames with bounding boxes. If holes are not supported, objects will be split.

Named Categories Not only grouping information is saved, but also a label to each group, like „car“ and „persons“, or „animals“. If this is not possible, groups will be saved without a group name.

Object names Give Objects name instead of just a random number within their group.

Keyframes Supports saving of keybbboxes and linearly interpolating in between them. If not possible to save keybbboxes, these and their associated virtual Bboxes will be converted to single Bboxes.

Run-length-encoding Run-length-encoding saves space by saving the bbox position,first appearance and the duration of its presence there. If RLE is not possible, BBoxes will be converted to single keyboxes.

Table 1: supported features by File format

	Categories	Holes	Named Cat.	Obj. Names	Keyframes	RLE
.btd	X	X	X		X	
ViPer XML	X	X	X			X
.bb						

A.1.1 .btd

BTD (binary tracking data) is our own data format and therefore capable of saving all TrackIt data without any loss. The Data is saved in binary mode using QDataStream's serialization mechanisms in version 12 so it can be written and read on any platform where Qt is available.

The actual serialization of our internal data structure is specified in table

A.1.2 ViPER xml File

Open-standard xml document format introduced by the ViPER-toolkit.
(see <http://vipер-toolkit.sourceforge.net/>) This file format is more thoroughly described

Table 2: BTD file format

Type	Description
quint8	Part of magic number, must be 0x42 ('B')
quint8	Part of magic number, must be 0x54 ('T')
quint8	Part of magic number, must be 0x44 ('D')
quint8	BTB version (currently 1)
QString	Filename of the associated video file (can also contain a relative or absolute path)
quint32	Current ID of the ID counter (Last assigned ID + 1)
QList<Category>	Categories
<i>Category</i>	
QString	Name of the category
QList<Object>	Objects
<i>Object</i>	
quint32	ID of the object
QList<BBBox>	Bounding Boxes
<i>BBBox</i>	
quint8	Type (1=single, 2=key)
quint32	Framenumber
QRect	Position and size

in

<http://viper-toolkit.sourceforge.net/docs/file/>. .

A.1.3 BB Files

BB Files have a simple parsable file format. The first line contains the number of frames, followed by a list of all frames. Each of these frames consists of the numbers of objects that appear in this frame, followed by an object definition, or nothing in case the number of objects is "0". Each object definition contains its lifetime in frames (its first appearance is determined by the framenummer that holds its data), followed by a (possibly empty TODO: Möglich?) list of bounding boxes. Each bounding Box consists of four integers: distance to top border of video, then distance to left border of video, width and height of video. The EBNF definition puts bb files in a more formal format: Unfortunately, BB Files allows only saving of bboxes for consecutive frames for an object, so if there are "holes" without a bbox in between, this object will be split into as many new objects. as it takes to create only objects with consecutive frames.

Table 3: BB File definition

BBFILE	::=	NUMBER_OF_FRAMES „\n“ LIST_OF_ALL_FRAMES
LIST_OF_ALL_FRAMES	::=	NUMBER_OF_OBJECTS „\n“ (OBJECT_DEFINITION ϵ)
OBJECT_DEFINITION	::=	LIFETIME_IN_FRAMES „\n“ LIST_OF_BBOX
LIST_OF_BBOX	::=	BBOX „\n“ (LIST_OF_BBOX ϵ)
BBOX	::=	NUMBER " " NUMBER " " NUMBER " " NUMBER // first number: Distance in pixels to top border of video // second number: Distance in pixels to the left border. // third number: Width in pixels. // fourth number: Height in Pixels.
NUMBER_OF_FRAMES	::=	NUMBER
NUMBER_OF_OBJECTS	::=	NUMBER
LIFETIME_IN_FRAMES	::=	NUMBER
NUMBER	::=	NUMBER NUMBER „0“ „1“ „2“ „3“ „4“ „5“ „6“ „7“ „8“ „9“

Table 4: BB file format

Type	Description
Integer	Number of frames
Frame	First frame
Frame	Second frame
...	...
<i>Frame</i>	
Integer	Framenumber
Integer	Number of objects starting in this frame
Object	First object
Object	Second object
...	...
<i>Object</i>	
Integer	Lifetime in frames
BBox	First bounding box
BBox	Second bounding box
...	...
<i>BBox</i>	
Float;Float;Float;Float	top, left, width, height

B Keyboard Shortcuts

Table 5: File actions

File Actions	
Open Data File	Ctrl+O
Open Video File	Ctrl+Shift+O
Save Data File	Ctrl+S
Save Data as	Ctrl+Shift+S
Import Data	Ctrl+I
Export Data	Ctrl+E
Quit Program	Ctrl+Q

Table 6: Movie/Data Actions

Action	Keys	Alt. Keys	Alt. Key2
Play/Pause	Space		
Next Frame	Cursor Right	D	L
Previous Frame	Cursor Left	A	J
Next Keyframe	Ctrl+Cursor Right	Ctrl+D	Ctrl+L
Previous Keyframe	Ctrl+Cursor Left	Ctrl+A	Ctrl+J
Next Category	E	O	Page Up
Previous Category	Q	U	Page Down
Next Object	Cursor Down	S	K
Previous Object	Cursor Up	W	I
Create new Object	Ctrl+R		
Create new BBox	Insert	R	Z
Create new Key BBox	Shift+Insert	F	H
Delete Marked BBox	Del		
Zoom in	Ctrl+ "MWHEEL_UP"	Ctrl+ "+"	
Zoom out	Ctrl+ "MWHEEL_DOWN"	Ctrl+ "-"	
Reset Zoom	Ctrl+0		
Zoom to Fit	Ctrl+Comma		

List of Figures