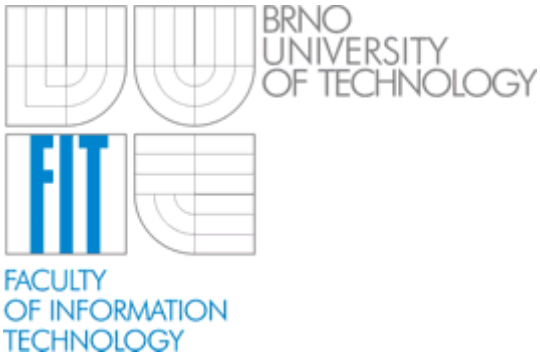


TETA

Tracking Evaluation Tool

version 1.0

Brno University of Technology

Faculty of Information Technology

Department of Computer Graphic and Multimedia

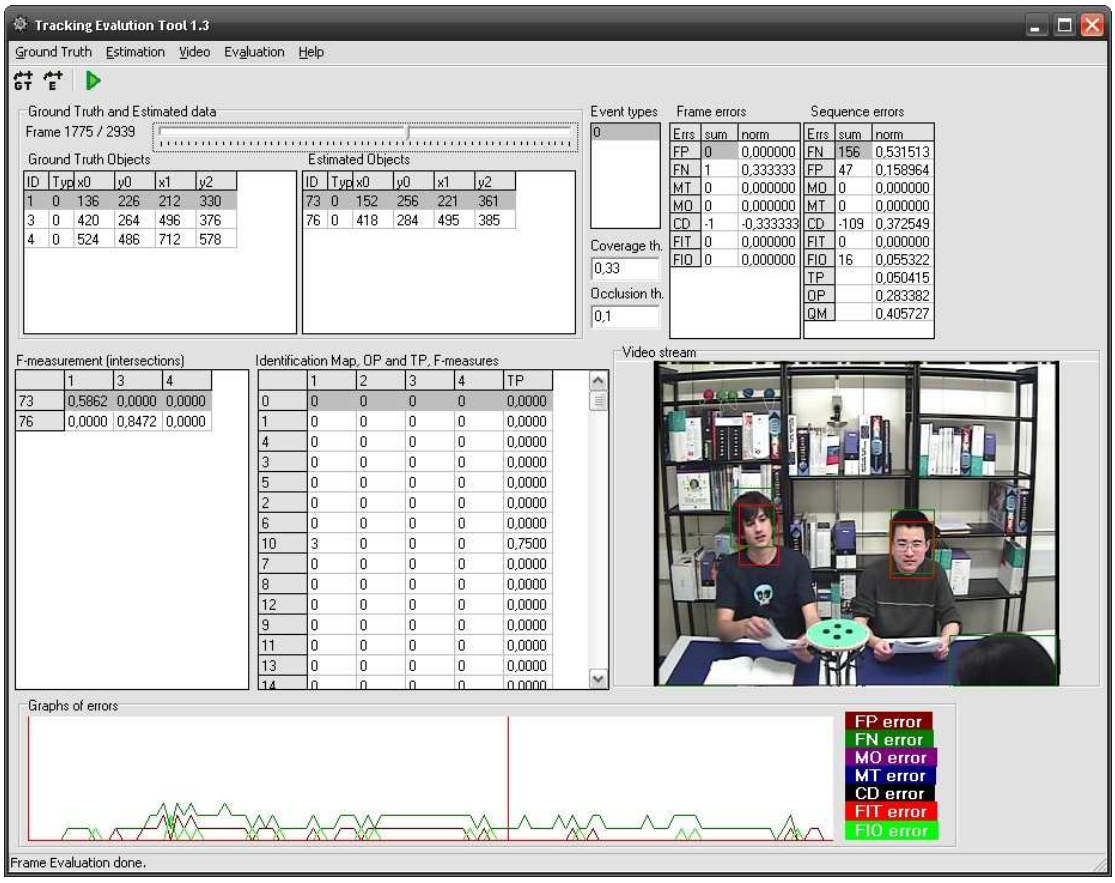
Vítězslav Beran, 14 November 2007

Content

Introduction

The TETA tool was designed according to published evaluation scheme [1] for evaluating multi-object tracking methods. The tool provides graphical user interface to browse through the processed video, to display both ground truth and tracked objects, to set some evaluation parameters or to work with graphical evaluation outputs in form of error graphs. Results are reported in both textual and graphical way.

The evaluation method compares ground truth data and estimated data using normalized coverage measure. The tool supports both textual and XML data formats with mutual conversion ability.



Installation

The TETA tool does not need any special installation. To install it, just make a copy of all files and directories to a local destination.

User manual

The TETA is a Windows XP application for evaluation of tracked video data. Two data sets are necessary at least. One, the annotated data referred to as the Ground Truth (GT) objects, and second, the output of an image-based tracking system referred to as the Estimations (E). An output of the evaluation is error values (defined and described in previous document part). The configuration errors can be evaluated for particular frame and the identification errors for entire sequence. Error values are displayed on interactive graph, which allows finding trouble frames effectively.

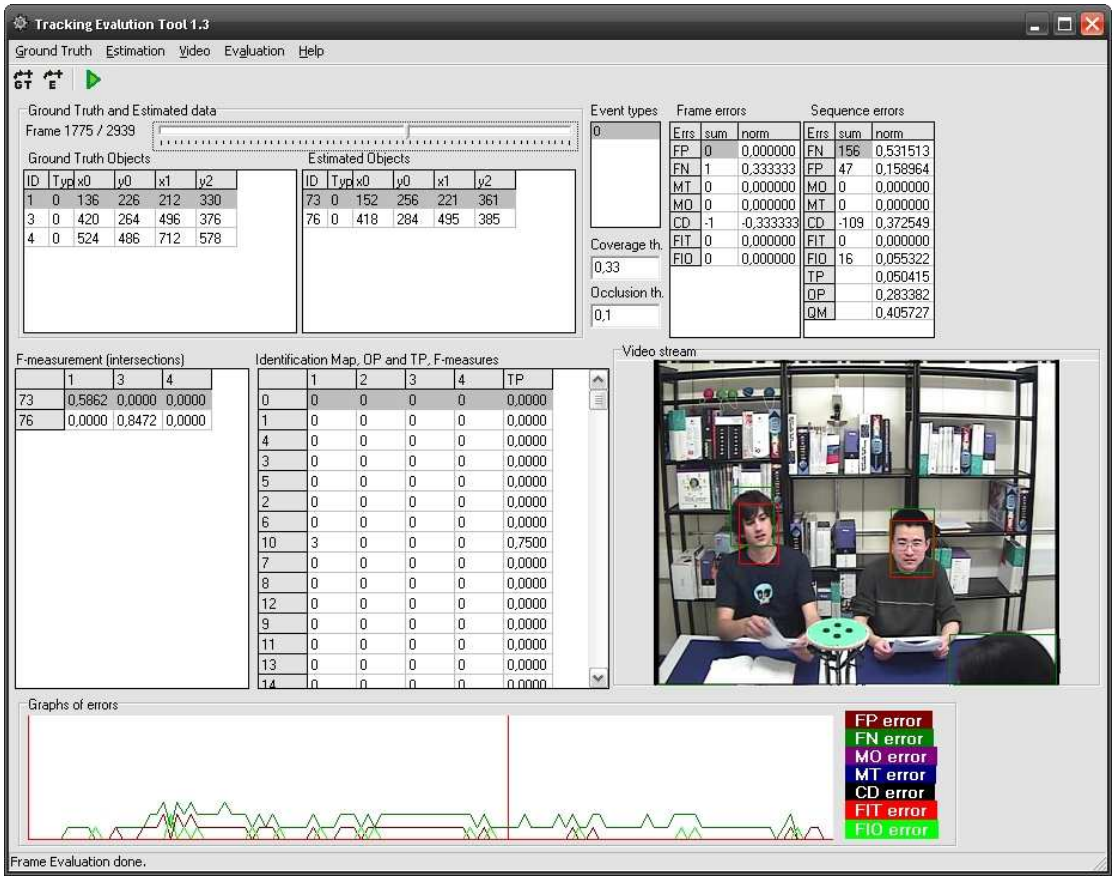


Figure 1 AMI Tracking Evaluator GUI.

The user interface provides all necessary information and functionalities just in one main frame that is composed from modules such as per-frame information, evaluation setting forms, statistical information or graphical error bars.

Data manipulation

The most important and controlling data set is the Ground Truth set. According to this set, the numbers and amount of frames in the sequence are initialized and also consequent browsing is allowed only through frames occurring in GT set.

Data in GT or E sets are managed using load, reset or save functions provided by menu or buttons.

- *Load to set* – firstly **clear** the particular set, then show 'Open file' dialog and **load** new events from opened file into existed set.

- *Add to set* - show ‘Open file’ dialog and **add** new events from opened file into existed set.
- *Reset* - **clear** the set.
- *Save as* - **store** data set into file selected in ‘Save as’ dialog and in XML format (not supported yet) or TXT format 1 (described later)

The XML event source contains information about event type (if the tracked object is the hand, head, face, etc.). Such information is not provided by TXT source format, so it must be set manually (see Figure 2).

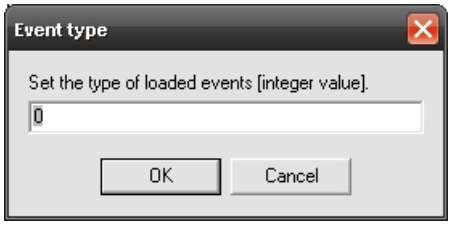


Figure 2 Dialog for default event type setting.

When loading events from the source file, it is possible to re-index event's IDs and also process the interpolation of event object parameters. For particular sequence, such functionality provides possibility to compose several annotations of different people which are annotated in different frames.

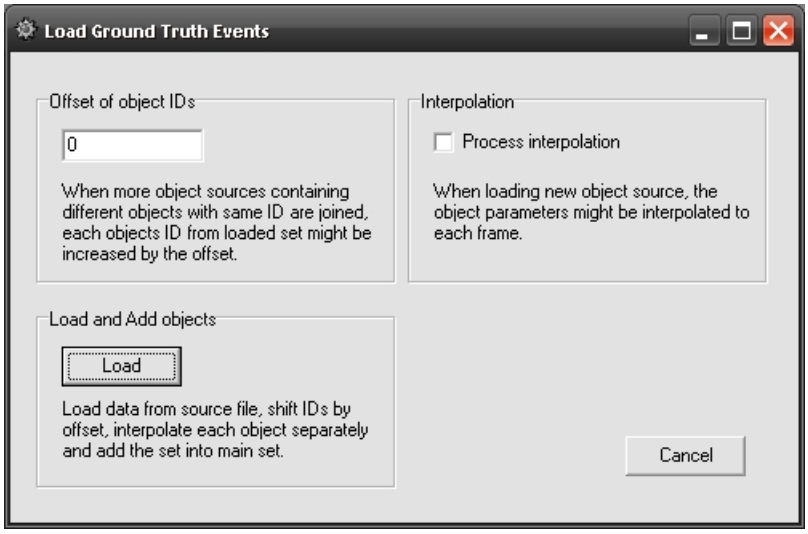


Figure 3 Dialog for settings and loading objects.

Evaluation and setting

The evaluation process run through all frames and all objects and computes several measurements (see ‘Multi-object tracking evaluation’ section). The error measurement is influenced by two thresholds – **Coverage** and **Occlusion**.

The coverage parameter controls, how much must be GT and E objects overlapped to be mapped together and occlusion parameter controls, how much must be two GT objects overlapped to be flagged as occluded.

Figure 4 Dialog for default event type setting.

Due to several types of possible events in some data sets, the event types that are processed can be selected in '**Event types**' table.

Browsing and per-frame information

When both GT and E sets are loaded, the evaluation process must take place first and then, the browsing over all frames with per-frame information is available.

Browsing through entire sequence is provided by tracking bar with **Frame** numbers. The frame amount and frame numbers is taken from GT set, not from E set! All events of particular frame are displayed in two tables (**GT Objects** and **E Objects**) where *Type* is an event type and *x0, y0* and *x1*, *y1* are the object area corners. The detected object parameters than can be manually compared.

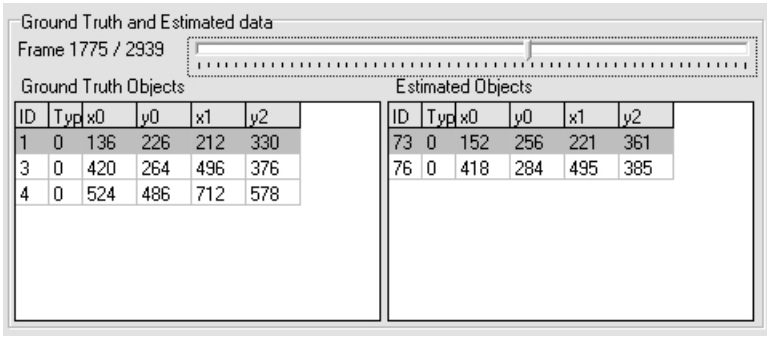


Figure 5 Browsing track-bar and actual frame objects listing.

The information about object identification map or configuration errors are also listed and can be useful e.g. for algorithm tuning. The F-measurement table contains corresponding objects within actual frame represented by objects IDs.

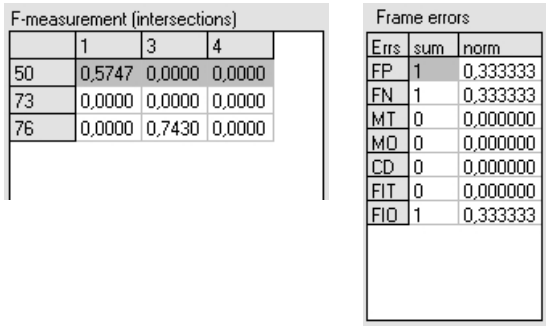
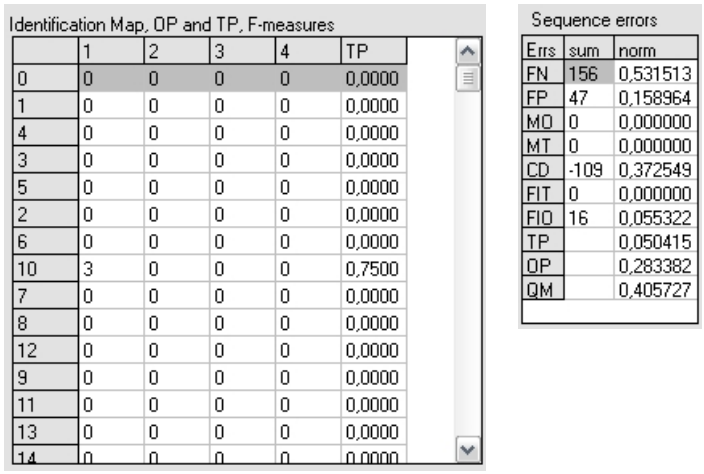


Figure 6 Information about objects measurements in one frame.

Configuration Map shows mapping of GT and E objects with flag of occlusion.

Results and graphs

When entire sequence evaluation is processed, the **Identification Map** is displayed and shows, how many times where GT and E objects mapped together over entire sequence.



Identification Map, OP and TP, F-measures

	1	2	3	4	TP
0	0	0	0	0	0,0000
1	0	0	0	0	0,0000
4	0	0	0	0	0,0000
3	0	0	0	0	0,0000
5	0	0	0	0	0,0000
2	0	0	0	0	0,0000
6	0	0	0	0	0,0000
10	3	0	0	0	0,7500
7	0	0	0	0	0,0000
8	0	0	0	0	0,0000
12	0	0	0	0	0,0000
9	0	0	0	0	0,0000
11	0	0	0	0	0,0000
13	0	0	0	0	0,0000
14	0	0	0	0	0,0000

Sequence errors

Errs	sum	norm
FN	156	0,531513
FP	47	0,158964
MO	0	0,000000
MT	0	0,000000
CD	-109	0,372549
FIT	0	0,000000
FIO	16	0,055322
TP		0,050415
OP		0,283382
QM		0,405727

Figure 7 Information about objects measurements in entire sequence.

Graphs of errors show the error values courses over entire sequence. Each graph can be turned off, so the observation of particular error graph is transparent.

Figure 8 Graphical representation of several error types.

Data formats

The TETA is able to import events from text and XML file and also to export data.

The XML files must follow format defined for Event Editor [1] application:

```
<!ELEMENT AVEvents (EventGroups?, EventTypes?, File?, Editor?)>

<!ELEMENT EventGroups (Group+)>
<!ELEMENT EventTypes (Type+)>

<!ELEMENT Group (ID, Name, Meaning?, Enabled?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Meaning (#PCDATA)>
<!ELEMENT Enabled (#PCDATA)>

<!ELEMENT Type (ID, Name, Key?, Group?, GroupIndex?, Offset?,
Parameters?, Secondary*)>
<!ELEMENT Key (#PCDATA)>
<!ELEMENT Group (#PCDATA)>
<!ELEMENT GroupIndex (#PCDATA)>
<!ELEMENT Offset (#PCDATA)>
<!ELEMENT Parameters EMPTY>

<!ELEMENT Secondary (Key, Offset?, Parameters?)>

<!ELEMENT File (Source*, TimeFormat?, Event*, Title*)>
```

```
<!ELEMENT Source (#PCDATA)>
<!ELEMENT TimeFormat (#PCDATA)>

<!ELEMENT Event (ID, Time, Text?, Parameters?)>
<!ELEMENT Time (#PCDATA)>
<!ELEMENT Text (#PCDATA)>

<!ELEMENT Title (Time, Text)>

<!ELEMENT Editor ANY>
```

The TXT files can be one of followed formats: **format 1**

```
frame    frameID
object   objectID   BoxCenterX   BoxCenterY   Width/2   Height/2
```

where all spaces are tabs, or **format 2**

```
image[frameID].* objectID minXPos minYPos maxXPos maxYPos
```

or **format 3**

```
frameID objectID visibility minXPos minYPos maxXPos maxYPos
```

The results of evaluation might be reported in two ways - full and brief. The full version reports all types of errors, tables with F-measure values for each E and GT event combination, errors for each frame, etc. The brief report print out only main errors in one line using CSV format:

```
Sequence; F-Measure; FN; FP; MT; MO; CD;
 FNbar; FPbar; MTbar; MObar; CDbar;
 FIT; FIO; FITbar; FIObar; TPbar; OPbar
```

The brief output is appended to file so the final overview of evaluation in table is easy to make.

Console version

For scripting purposes, there exists also console application. The output of the Console Evaluator is stored to text file described later.

The Console Evaluator needs several parameters and all are necessary for proper algorithm's run. The program usage is as follows:

```
Ami_Evaluator_Console.exe [params]
   -gtf [filename]  -  name of the file with Ground Truth objects,
   -ef  [filename]  -  name of the file with Estimate objects,
   -out [filename]  -  name of the output file,
   -gtt [integer]   -  default Event Type for Ground Truth objects
                       when loaded from TXT source,
   -et  [integer]   -  default Event Type for Estimate objects when
                       loaded from TXT source,
   -tc  [float]     -  coverage threshold (0.33),
   -to  [float]     -  occlusion threshold (0.8),
   -b               -  brief output; will be appended at the end of
                       output file,
   -nt  [integer]   -  amount of inserted Types,
        [integer]   -  allowed Types, amount of inserted integers
                       must be same as set 'nt' params.
```

Example (data are in the same directory as the evaluator):

```
Ami_Evaluator_Console_10 -gtf GT.txt -ef E.txt -out
evaluation.txt -gtt 1 -et 1 -tc 0.4 -to 0.8 -nt 2 1 2
```

The format of store results contains more information than the application. Besides final Identification errors, there are also stored the Configuration errors for each evaluated frame in CSV format. The evaluation parameters are also stored in the output file.

Multi-object tracking evaluation

Fundamental concept for evaluation the performance of the different tracking algorithms is introduced. The quality of tracking result for a single object is based on shape shape-independent measures.

In following sections, the labeled tracking targets are denoted as ground truth objects *GT*, tracker outputs are referred to as estimates *E*. The output of a tracking approach is considered to be correct, if and only of one *GT* (resp. *E*) is tracking exactly one *E* (resp. *GT*).

Coverage test

Two measurements, recall represents the ration of the ground truth *GT* area, which is covered by the estimate *E*, and precision represents the ration of the estimate *E* area, which is covered by the ground truth *GT*:

$$\alpha_{i,j} = \frac{\left|E_i \cap GT_j\right|}{\left|GT_j\right|}; \qquad \beta_{i,j} = \frac{\left|E_i \cap GT_j\right|}{\left|E_i\right|} \qquad \textit{Eq. 1}$$

where $\alpha_{i,j}$ is the recall and $\beta_{i,j}$ is precision. Returning high value only when both recall and precision are high, F-measure is used:

$$F_{i,j} = \frac{2\alpha_{i,j}\beta_{i,j}}{\alpha_{i,j} + \beta_{i,j}} = \frac{2\left|E_i \cap GT_j\right|}{\left|E_i\right| + \left|GT_j\right|}. \qquad \textit{Eq. 2}$$

Configuration errors

In this context, configuration means the number, the location and the size of all objects in a frame of the scenario. To identify all types of errors, 5 configuration measures are introduced.

- False positive (*FP*) – there is an *E* indicating object, where no *GT* is.
- False negative (*FN*) – *GT* is not tracked by an *E*.
- Multiple trackers (*MT*) – more than one *E* is associated with only one *GT*.
- Multiple objects (*MO*) – more than one *GT* is associated with only one *E*.
- Configuration distance (*CD*) – normalized difference between amount of *E* and *GT* objects. Relevant only together with other errors.

Occlusion handling

The occlusion flag is defined, enlarging *GT* objects, if the ratio of *GTj* area, which is covered by *GTk* exceeds certain threshold t_O:

$$occ_j = \begin{cases} 1, & \exists GT_k : \dfrac{\left|GT_j \cap GT_k\right|}{\left|GT_j\right| + \left|GT_k\right|} > t_O \\ 0, & \text{otherwise} \end{cases}. \qquad \textit{Eq. 3}$$

In situation when occlusion flag is set, there is no evaluation of any error.

Evaluation procedure

The F-measure is used to evaluate all 4 types of errors and create the configuration map table with respect to occlusion flag of particular *GT* object.

$$FP = \sum_i \mathbf{I}\left[\sum_j \mathbf{I}\left(F_{i,j} > t_I\right) = 0\right] \quad FN = \sum_j \mathbf{I}\left[\sum_i \mathbf{I}\left(F_{i,j} > t_I\right) = 0\right] \qquad \textit{Eq. 4}$$

$$MT = \sum_i \mathbf{I}\left[\sum_j \mathbf{I}\left(F_{i,j} > t_I\right) > 1\right] \quad MO = \sum_j \mathbf{I}\left[\sum_i \mathbf{I}\left(F_{i,j} > t_I\right) > 1\right] \qquad \textit{Eq. 5}$$

$$CD = \frac{N_E - N_{GT}}{\max\left(N_{GT}, 1\right)} \qquad \textit{Eq. 6}$$

where t_I is the threshold of F-measure, when *E* maps *GT*, *GT* maps *E* resp., **I** function returns 1 when the expression is true, 0 otherwise, and i, resp. j is an index of E, resp. GT objects.

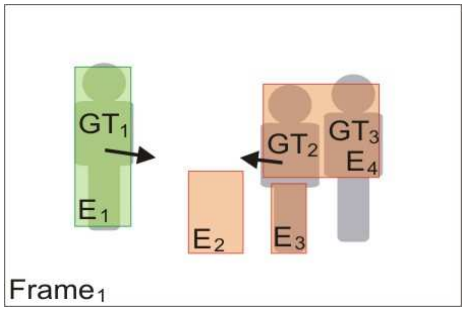


Figure 9. Frame configuration example.

		GT_j			$\sum_j \mathbf{I}(F_{i,j} > t_I)$
		1	2	3	
E_i	1	0,89	0,00	0,00	1
	2	0,00	0,00	0,00	0
	3	0,00	0,51	0,00	1
	4	0,00	0,47	0,47	2
$\sum_i \mathbf{I}(F_{i,j} > t_I)$		1	2	2	

Table 1. F-measure table with error evaluation.

E	*GT*	*occ*
1	1	0
3	2	0
4	2	0
4	3	0

Table 2. Configuration map.

Error type	value
FP	1
FN	0
MT	1
MO	1
CD	0,33

Table 3. Evaluated configuration errors.

For an easy comparison of tracking algorithm errors, are normalized over entire sequence using amount of *GT* objects in each:

$$\overline{X} = \frac{X}{n} \sum_{t=0}^{n} \frac{1}{\max\left(N_{GT}^{t}, 1\right)} \qquad \textit{Eq. 7}$$

where $X \in \{FP, FN, MO, MT\}$.

Identification errors

Identification means that particular *E* tracks exactly one *GT* over its entire lifetime (correctly identifies this *GT* object). 'Majority rule' was used to represents identification association. Two errors are defined and the degree of consistency.

- Falsely identified tracker (*FIT*) – *GT* is mapped by different *E* than frame before.
- Falsely identified object (*FIO*) – *GT* is mapped but frame before was not mapped.
- Object purity (*OP*) – ratio between amounts of frames when GT was correctly identified to the overall amount of frames.
- Track purity (*TP*) – same as *OP* but for *E*. Not interesting for our purpose.

Evaluation procedure

The identification map is constructed during configuration errors are evaluated for each frame in the sequence. Configuration maps of each pair of frames serve to evaluate identification errors.

$$\overline{FIT} = \frac{1}{n}\sum_{t=1}^{n}\frac{\mathbf{I}\left(GT_j^t \to E_i^t \wedge GT_j^{t-1} \to E_k^{t-1}\right)}{\max\left(N_{GT}^t, 1\right)} \qquad \textit{Eq. 8}$$

$$\overline{FIO} = \frac{1}{n}\sum_{t=1}^{n}\frac{\mathbf{I}\left(GT_j^t \to E_i^t \wedge !\exists i : GT_j^{t-1} \to E_i^t\right)}{\max\left(N_{GT}^t, 1\right)} \qquad \textit{Eq. 9}$$

$$\overline{OP} = \frac{1}{j}\sum_{j}\frac{\max_i\left(IM_{i,j}\right)}{\sum_i IM_{i,j}} \qquad \textit{Eq. 10}$$

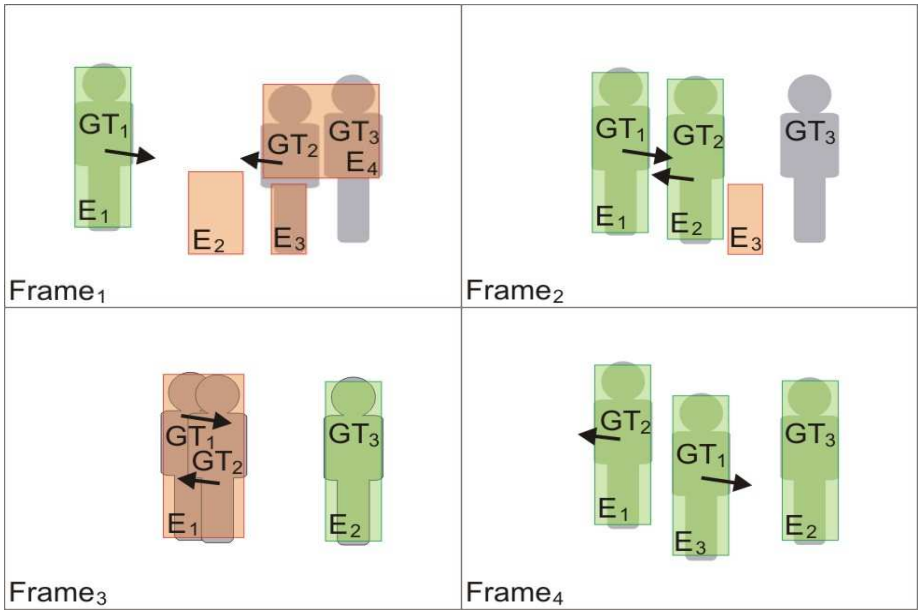


Figure 10. Example of sequence with GT and E objects.

Identification map for example sequence shows amounts of *GT* and *E* mapping and evaluation purity of *GT*, resp. *E*.

$IM_{i,j} = \sum_t^n \mathbf{I}\left(E_i \to GT_j\right)$		GT_j		
		1	2	3
E_i	1	2	1	1
	2	1	1	0
	3	0	1	1
	4	0	1	2

Table 4. Identification map.

Final error values for example sequence.

Configuration errors		Identification errors	
$\overline{FP}$	0,17	$\overline{FIT}$	0,17
$\overline{FN}$	0,08	$\overline{FIO}$	0,08
$\overline{MO}$	0,08	$\overline{OP}$	0,53
$\overline{MT}$	0,08	$\overline{TP}$	0,58
$\overline{CD}$	0,17	$\overline{ME}$	x

Table 5. Errors of example sequence.

References

[1] Schreiber, S., Gatica-Perez, D., Potúček, I., Thean, A., Wrigley, S. N.: AMI WP4 Tracking: Evaluation scheme (Draft), 2005.

[2] Sumec, S., Kadlec, J. Event Editor - The Multi-Modal Annotation Tool, Workshop on Multimodal Interaction and Related Machine, Learning Algorithms (MLMI), Edinburgh, GB, 2005.

[3] Zemčík, P., Herout, A. DigILib. http://www.fit.vutbr.cz/research/groups/graph/index.php?page=digilib

Appendix

Application class diagram

In the schema, there are not displayed the functions for allocation and de-allocation of the structures. Each function has also one parameter for more, the pointer to particular structure, which is not stated in the UML diagram to make the diagram small.

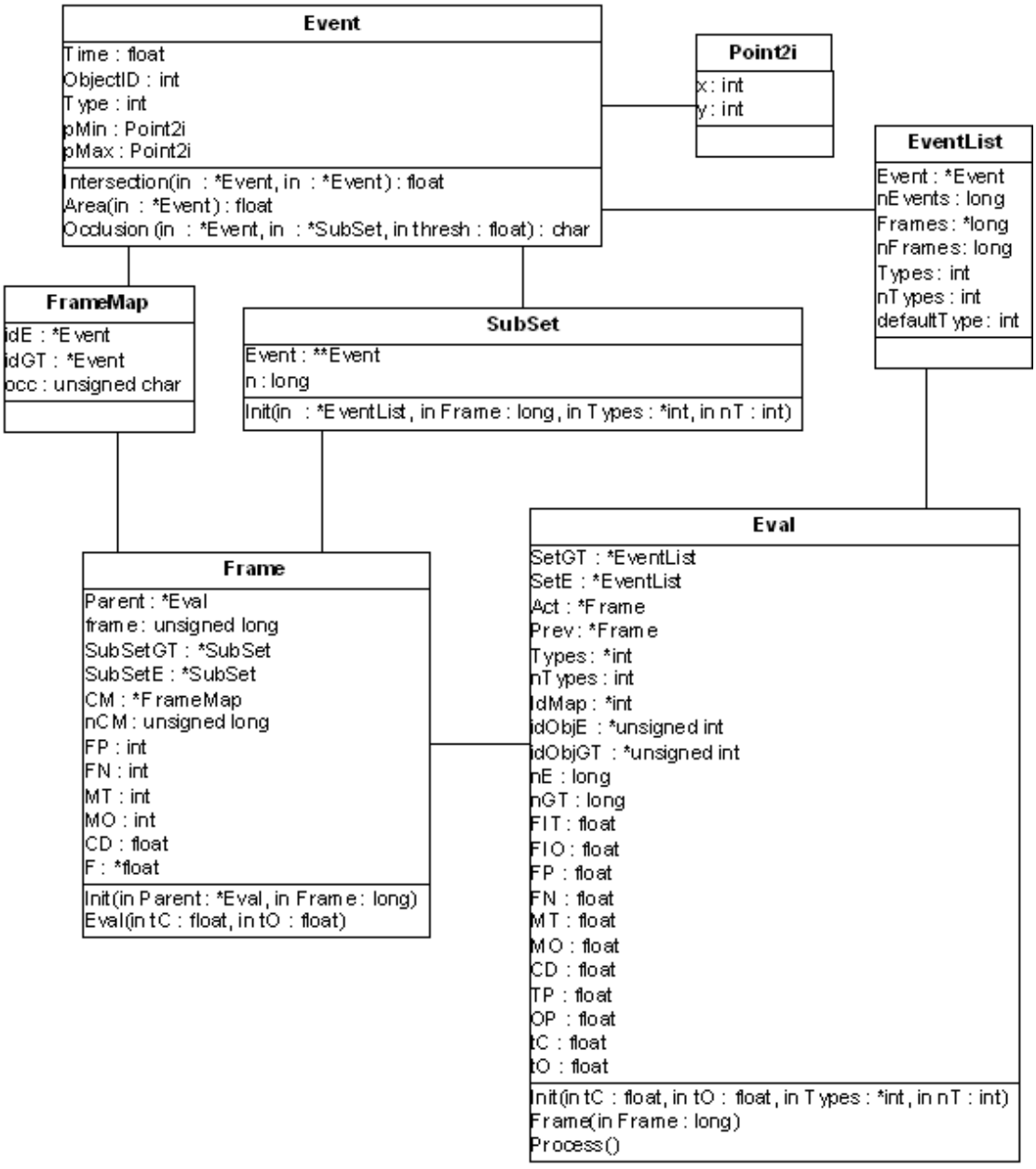


Figure 11 UML diagram of structures and functions for tracking evaluation.

The evaluation structures and functions were implemented using C language.

The GUI was designed and generated using Borland Builder C++ and DigILib [5].