Working Document Evaluation and Metrics for Video Understanding

Date: 11/10/2010

Version: **1.0**
Author: **INRIA**

EVALUATION AND METRICS FOR VIDEO UNDERSTANDING

The objective of this document is to introduce the video sequence selection criteria and the metrics expected to be used at INRIA for video understanding projects.

1. DATA TERMINOLOGY

This section enumerates and defines all the vocabulary describing data used in a video understanding process.

Image: array of pixel generated at a time step by a video camera (e.g., composite, CCD, CMOS, PTZ, omni directional). An image is characterized by a timestamp (year, month, day, hour, minute, second, millisecond) and can correspond to a frame interleaved or not. An image can be of the following type: colour, black and white, infrared and with different compression levels.

Video sequence: temporal sequence of images which are generated by a video camera. A video sequence can be represented as a live stream (e.g., composite signal, MJPEG stream), as a file (e.g., a MPEG4 encoded file) or as a sequence of files (e.g., a sequence of JPEG files).

Video clip: a part of a video sequence, which corresponds to a particular situation to be evaluated.

Scene: the physical space where a real world event occurs and which can be observed by one or several video cameras. A scene without any physical object of interest is called an **empty scene**.

Blob: 2D image region that has been segmented based on regions (e.g., homogeneous in motion, colour, energy or texture information) or contours (e.g., using a shape model). This region can be defined as a set of pixels (not necessarily connected) or as a polygon delimiting its contour. It can be characterized by 2D features such as a colour histogram, a density, a 2D width and height.

Moving region: a blob that has been created following a motion criteria (e.g., either optical flow or reference image subtraction).

Physical object: a real world object in the scene. There are two types of physical objects: **physical object of interest** and **contextual object**.

Physical object of interest: a physical object evolving in the scene whose class (e.g., person, group, vehicle) has been predefined as interesting by end-users and whose motion cannot be foreseen using a priori information. It is usually characterized by a semantic class label, 2D or 3D features (e.g., 3D location, width and height, a posture, a trajectory, a direction, a speed), a list of blobs, an initial tracking time, a camera number for the camera which is the best seeing the object (in a multi camera configuration), an identifier. An identifier can either be defined locally to the current image, globally on the video sequence or globally on a scene (in a multi camera configuration).

Contextual object: a physical object attached to the scene. The contextual object is usually static and whenever in motion, its motion can be foreseen using a priori information. For instance, it can be in motion such as a door, an elevator, a fountain, a tree or displaceable (by a human being) such as a chair, a luggage.

Event (activity): generic term to describe any event, action or activity happening in the scene and visually observable by cameras. Events of interest can either be predefined by end users or learned by the system. Events are characterized by the involved objects of interest (including contextual objects and zones of interest), their starting and ending time and by the cameras observing the events. Examples of events are “intrusion in a forbidden zone”, “detection of an abandoned bag”, “detection of a fighting situation”, ”a meeting between two people”...

2. VIDEO UNDERSTANDING FUNCTIONALITIES

Video understanding is a process recognising user events in a scene observed by video cameras. The whole processing chain goes from pixel analysis up to alarm generation and is composed of four main video processing tasks:

- Task 1: detection of physical objects of interest: decomposition of the image into blobs (2D regions) corresponding to potential physical objects of interest. A typical approach consists in separating first moving pixels from non-moving pixels and then clustering moving pixels into blobs. A moving pixel is a pixel whose intensity is sufficiently different from the corresponding pixel in a reference image (e.g., background or previous image). Advanced functionalities consist in being able to distinguish interesting moving pixels generated by human activities from those corresponding to noise generated by contextual objects (e.g., moving trees), shadows or reflections. These advanced functionalities may require the use of contextual information (e.g., 3D geometry of the empty scene), a sophisticated reference image or chromatic information about pixels. The output of task 1 is a grey level image (0 = background, n = identifier of the physical object of interest).

- Task 2: classification of physical objects of interest: classification of blobs into labels corresponding to classes of physical objects of interest, with respect to a predefined semantic: person, vehicle, group of persons, etc. Advanced functionalities consist in refining object classes (e.g., motorcycle, cycle, car, truck, airplane, for the vehicle class), in splitting objects (e.g., two separate persons are better than a group), in computing a posture and an orientation for objects, in computing their 3D parameters while taking into account static occlusions by contextual objects. The output of task 2 is a list of physical objects with their properties.

- Task 3: tracking of physical objects of interest: process which consists in matching objects detected at image time t-1 with those detected at image time t and maintaining a unique identifier for each object over the whole video clip. Advanced functionalities consist in tracking separately rather than globally physical objects in case of dynamic occlusion, in tracking accurately objects even in case of static occlusion, in tracking objects in a network of cameras with overlapping or distant field of view. The output of task 3 is a list of physical objects with their properties (the camera having the best viewing point to observe the physical objects, trajectory, kinematics, time-filtered properties) and their links to previous objects.

- Task 4: event recognition: the goal of this step is to recognize any event predefined by the user (abandoned bag, forbidden zone access, attack) from descriptors given by preceding tasks (e.g., shape, speed, position and trajectory). An event is characterized by involved objects, the event recognition initial time and the camera having the best viewing point (in a multi camera configuration).

There are many ways to implement these tasks. Some systems only address task 1, some systems combine the first three tasks into a single task while others merely skip task 2. Video understanding systems may address globally these four main tasks. However, when it is possible, we propose to evaluate them at the end of each task. The output of task 4 is a list of events together with their involved physical objects.

3. VIDEO SEQUENCE DATABASE

From a general point of view, reference data should correspond to views of the real world thus they should represent real difficulties encountered during software and application development. They should illustrate video surveillance applications from indoor and outdoor scenes including persons and vehicles, etc. The configuration can be mono or multi cameras and sensors can be black and white, colour or infrared ones. The calibration information is available.

The database should contain instances of increasing levels of difficulties in several categories. For instance, if the studied problem is the management of crossings between people, the database should contain sequences ranging from crossings with 2 persons to crossings implying at least 5 persons. The current video sequence categorization is as follows:

V1) Acquisition information:

V1.1) Camera configuration: mono or multi cameras,

V1.2) Camera type: CCD, CMOS, large field of view, thermic cameras (infrared),

V1.3) Compression ratio: no compression up to high compression,

V1.4) Camera motion: static, oscillations (e.g., camera on a pillar agitated by the wind), relative motion (e.g., camera looking outside a train), vibrations (e.g., camera looking inside a train),

V1.5) Camera position: top view, side view, close view, far view,

V1.6) Camera frame rate: from 25 down to 1 frame per second,

V1.7) Image resolution: from low to high resolution,

V2) Scene information:

V2.1) Classes of physical objects of interest: people, vehicles, crowd, mix of people and vehicles,

V2.2) Scene type: indoor, outdoor or both,

V2.3) Scene location: parking, tarmac of airport, office, road, bus, a park,

V2.4) Weather conditions: night, sun, clouds, rain (falling and settled), fog, snow, sunset, sunrise,

V2.5) Clutter: empty scenes up to scenes containing many contextual objects (e.g., desk, chair),

V2.6) Illumination conditions: artificial versus natural light, both artificial and natural light,

V2.7) Illumination strength: from dark to bright scenes,

V3) Technical issues:

V3.1) Illumination changes: none, slow or fast variations,

V3.2) Reflections: reflections due to windows, reflections in pools of standing water, reflections due to bright floors,

V3.3) Shadows: scenes containing weak shadows up to scenes containing contrasted shadows (with textured or coloured background),

V3.4) Moving Contextual objects: displacement of a chair, escalator management, oscillation of trees and bushes, curtains, V3.5) Static occlusion: no occlusion up to partial and full occlusion due to static contextual objects,

V3.6) Dynamic occlusion: none up to a person occluded by a car, a person occluded by another person,

V3.7) Crossings of physical objects: none up to high frequency of crossings and high number of implied objects,

V3.8) Distance between the camera and physical objects of interest: close up to far,

V3.9) Speed of physical objects of interest: stopped, slow or fast objects,

V3.10) Posture/orientation of physical objects of interest: lying, crouching, sitting, standing,

V3.11) Calibration issues: little or large perspective distortion,

V4) Application type:

V4.1) Primitive events: enter/exit zone, change zone, walking, running, following someone, getting close,

V4.2) Suspicious behaviour detection: violence, fraud, tagging, loitering, vandalism, stealing,

V4.3) Intrusion detection: person in a sterile perimeter zone, car in no parking zones,

V4.4) Monitoring: traffic jam detection, counter flow detection, home surveillance, abandoned bag,

V4.5) Statistical estimation: people counting, car speed estimation,

Other video sequence characteristics can be investigated (such as PTZ camera, Omni directional cameras, stereo vision, and aerial view) however; only above characteristics are first priorities.

The video sequence database should be classified into three sets: work, test and evaluation set. The **work data set** is representative of the various sequences contained in the database. It is distributed to participants to allow them to run, modify and adjust their algorithms the way they want. In order to give participants the maximum amount of time, a non-exhaustive data sub set will be distributed at the beginning of the collecting phase of the first data set. The **test data set** is created and distributed at the beginning of the evaluation cycle validation. It is representative of the next coming evaluation set. It contains various sequences illustrating several cases predefined in the evaluation process. The analysis of the comparison results enables to qualify the pertinence of the chosen video sequences.

The **evaluation set** is intended to assess performances of participant algorithms. It contains the same variety of sequences as the test set but with more video clips in order to obtain sound statistical evaluation results.

Several characteristics of the tested algorithms cannot be automatically evaluated by a comparison with reference data. For instance, we can mention the processing time, the memory space usage, the amount of interactivity required, and the need of a learning phase. A questionnaire will be established during the first seminar. It will be distributed to participants along with data and they will send it back with their algorithm results. This information will enable a more detailed analysis. Finally, a live demonstration could be organized during the last workshop to run a participant algorithm on the data set and in presence of other participants and the organising committee. This opportunity will be investigated during the first seminar.

4. VIDEO UNDERSTANDING EVALUATION

This section enumerates and defines all the vocabulary used for the evaluation of a video understanding process as well as the considered process of automatic supervised evaluation.

Evaluation criterion: an evaluation criterion is an evaluation functionality to compare video understanding algorithm results with reference data. For instance, for the task “detection of physical objects of interest”, a criterion can evaluate the accuracy of the 2D or 3D location of objects, another one can evaluate the quality of the object shape. For the task “classification of physical objects of interest”, a criterion can evaluate the quality of the assigned class labels. In addition, these criteria can be detailed with regard to video clip categories. In the previous example, the assignment of class labels under static occlusion situations could be qualified, for instance.

Evaluation metric: a distance between video understanding algorithm results and reference data implementing an evaluation criterion. A way of displaying evaluation results is to use a ROC (Receiver Operating Characteristic) curve defined as a plot of the true positive rate against the false positive rate.

Ground truth data: data given by a human operator and which describe real world expected results (e.g., physical objects, events) at the output of a video understanding algorithm. These data are supposed to be unique and corresponding to end user requirements even if in many cases, this information can contains errors (annotation bias). These data can be written in a XML or MPEG7 format.

Annotation: information associated to a video clip including ground truth data plus other types of information about technical difficulties (e.g., shadows) and recording conditions (e.g., weather conditions) of the video clip under consideration. These annotations can provide several types for false or incorrect results (e.g., wrong classification, wrong detection).

Reference data: data supposed to be constant and unique, corresponding to a functionality of a video understanding task and used to evaluate the output of a video understanding algorithm at a given task level. Reference data include ground truth data, data given by a video expert and data computed from all annotation and contextual information. For instance, the 3D position of a person is a reference data computed from the bounding box given by a video expert and the calibration matrix. In addition, rules should be given to video experts in order to define as objectively as possible particular data. For instance, for a partially occluded person, one can choose to draw the bounding box for the visible part only or for the full object (including its hidden part).

Automatic supervised evaluation: process assessing algorithm performances in an automatic manner by comparing algorithm outputs with reference data. Automatic evaluation means that all the comparison is done without human interaction. In the automatic case, criteria and metrics are predefined and encoded into the evaluation system. Supervised evaluation means that video experts and human operators provide reference data used for the comparison.

The prerequisite to obtain an automatic evaluation based on a reference data comparison is to define reference data through the annotation process. Then, a comparator, which uses evaluation criteria and metrics, must be designed. Finally, this comparator (called evaluation tool) can be run on a video understanding algorithm to produce evaluation results quantifying the adequacy between algorithm outputs and reference data. This evaluation is run at each step (i.e., at each image for a mono camera processing) and its goal is to highlight algorithm capability to solve a set of current problems (e.g., shadow management, occlusions, object crossings) for each task of the video understanding chain. The evaluation process requires taking a decision concerning three topics: video database, annotation, evaluation criteria and metrics.

5. EVALUATION CRITERIA AND METRICS

This section describes criteria and metrics, which will be used by the evaluation tool, for each task.

There are 3 metric types which will be used by the ViSEvAL evaluation tool:

→ **Frame Metric:** this metric involves a set of criterion, distances between mobiles and a set of filters. It is computed for each frame.

Example: here is an example of a frame metric such defined in a configuration file.

MetricFrame *"Mono:M1.1:M1.1.1:DiceCoefficient:0.3:Identity"*
MetricFrame *"Mono:M1.1:M1.1.2:Overlapping:0.3:CloseTo:X:3:Y:5:Threshold:2"*
MetricFrame *"Mono:M1.1:M1.1.3:Overlapping:0.3:FarFrom:X:3:Y:5:Threshold:2"*

→ **Temporal Metric:** this metric involves a set of criterion, distances between and a set of filters. It is computed on a whole video sequence.

Example: here is an example of a temporal metric such defined in a configuration file.

MetricTemporal *"Mono:M3.2:M3.2.1:DiceCoefficient:0.3:Identity"*
MetricTemporal *"Fusion:M3.2:M3.2.1:3DOverlapping:0.5:Identity"*

In the previous examples for the configuration file:

- Mono is used when a metric is computed for object detected by one video camera (the bounding box describing the object is in 2D)
- Fusion is used when a metric is computed for object detected by several video cameras (the bounding box describing the object is in 3D)

→ **Event Metric:** this metric involves a set of criterion

Example: here is an example of an event metric such defined in a configuration file.

MetricEvent *"M4.1:M4.1.1"*
MetricEvent *"M4.2:M4.2.1:Duration:10"*
MetricEvent *"M4.3:M4.3.1:Frame:10"*

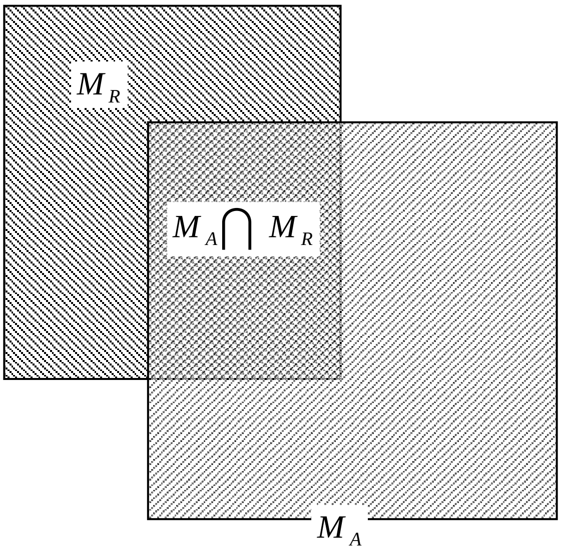
5.1 Distance between two mobiles

There are 2 types of distance: 2D distance for one video camera (Mono) and 3D distance for fusion between several video cameras (Fusion).
The distances are between bounding box of the detected objects and the bounding box of the reference data.

5.1.1 2D Distance:

2D distance includes **Dice Coefficient**, **overlapping** and **Bertozzi.**

Computation of the bounding boxes overlapping



Distance D1: **The Dice Coefficient**. The mobile detected with the algorithm M_A is matching with a mobile of the reference data M_R following this formula:

$$D_1(M_A, M_R) = \frac{2 * A(M_A \bigcap M_R)}{A(M_R) + A(M_A)}$$, where $A(.)$ is **the area**.

In the configuration file, the distance is used as:

*MetricFrame "Mono:M1.1:M1.1.1:**DiceCoefficient**:0.3:Identity"*

With Threshold parameter (equal 0.3 in this example)

Distance D2: The overlapping.

$$D_2(M_A, M_R) = \frac{A(M_A \bigcap M_R)}{A(M_R)}$$

In the configuration file, the distance is used as:

*MetricFrame "Mono:M1.1:M1.1.2:**Overlapping**:0.1:CloseTo:X:3:Y:5:Threshold:2"*

With Threshold parameter (equal 0.1 in this example)

Distance D3: Bertozzi.

$$D_2(M_A, M_R) = \frac{A(M_A \cap M_R)^2}{A(M_R) * A(M_A)}$$

In the configuration file, the distance is used as:

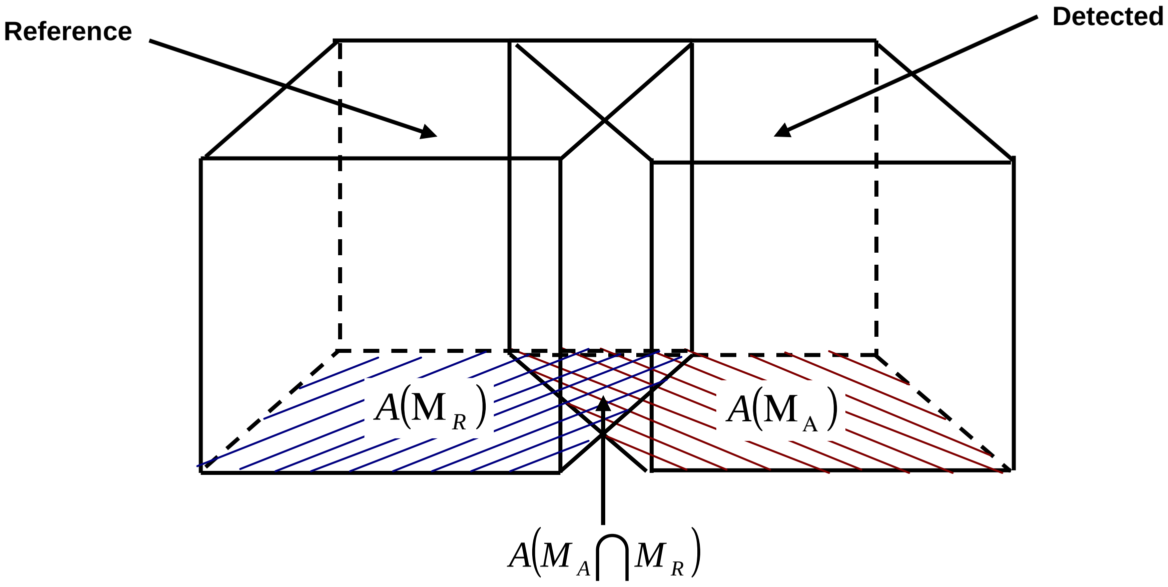
*MetricFrame "Fusion:M1.1:M1.1.6:**3DBertozzi**:0.2:Identity"*

With Threshold parameter (equal 0.1 in this example)

All the distances belong to [0, 1], 1 means the two mobiles have the same bounding boxes, 0 means the bounding boxes are disjoint.
These distances will provide values that may be interpreted "as it" but also permit, after thresholding, to validate the matching criteria.

5.1.2 3D distance:

3D distances are the same than the 2D distances where $A(.)$ is **the area of the base.**



5.2 Filters:

The filters are used to select object of interest according to a property P. There are 3 filters:

5.2.1 Identity: the property P is identity, all the objects are kept

Example:

*MetricFrame "Mono:M2.1:M2.1.1:DiceCoefficient:0.3:**Identity**"*

5.2.2 Close to: the property P of this filter is a distance from an object (situated at (a, b)) to a point (X, Y). The object is kept if its satisfied: $\|(a,b)-(X,Y)\| \leq Threshold$

Example:

*MetricFrame "Fusion:M1.1:M1.1.2:3DBertozzi:0.3:**CloseTo**:X:3:Y:5:Threshold:2"*

This filter needs 3 parameters: X, Y, and Threshold

5.2.3 Far from: the property P of this filter is a distance from an object (situated at (a, b)) to a point (X, Y). The object is kept if its satisfied: $\|(a,b)-(X,Y)\| \geq Threshold$

Example:

*MetricFrame "Mono:M1.1:M1.1.3:Overlapping:0.3:**FarFrom**:X:3:Y:5:Threshold:2"*

This filter needs 3 parameters: X, Y, and Threshold

5.3 Criterion:

Results will be presented in two manners:

1. **Global:** to compare globally all algorithms and to obtain a meaningful measure of their performances. The results will be given by computing the classical rates TP, FN, FP, Precision and Sensitivity according to the following table.

	Reference Data (RD)	Noise (N)	
Detected	True Positive (TP)	False Positive (FP)	Precision (P) = TP/(TP+FP)
Not Detected	False Negative (FN)		
	Sensitivity (S) = TP/(TP+FN)		

2. **Detailed**: to obtain a clear and precise view on performances of the various algorithms at several points in the video interpretation chain. The results are given for each object of the reference data by computing previous defined rates (TP, FN, FP, P, S). With these methods the user can see exactly which object is difficult for a given criteria.

   We list in the next the different criterion available in the ViSEvAL too. In following, a detected object matches a reference data if a distance between this object is greater than a predefined threshold and if the criteria is verified.

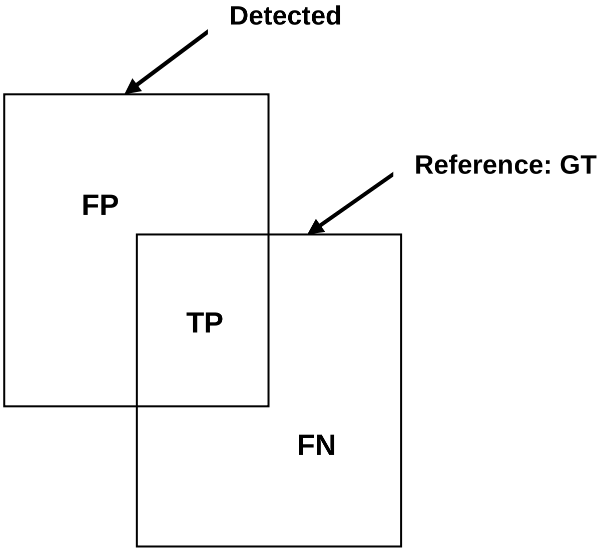
5.3.1 Frame metric:

M1.1: Number of detected objects matching to the reference data

<u>Example:</u>

*MetricFrame "Mono:**M1.1**:M1.1.3:Overlapping:0.3:FarFrom:X:3:Y:5:Threshold:2"*

M1.2: Number of common pixels between detected object and the reference data by analyzing their bounding boxes.



<u>Example:</u>

*MetricFrame "Mono:**M1.2**:M1.2.1:DiceCoefficient:0.3:Identity"*

M2.1: Number of classified objects according to the type matching to the reference data.

<u>Example:</u>

*MetricFrame "Mono:**M2.1**:M2.1.1:DiceCoefficient:0.3:Identity"*

M2.2: Number of classified objects according to the type and the sub-type matching to the reference data.

<u>Example:</u>

*MetricFrame "Fusion:**M2.2**:M2.2.1:3DOverlapping:0.1:Identity"*

M3.1: Number of links between physical objects matching to reference data links.

<u>Example:</u>

*MetricFrame "Fusion:**M3.1**:M3.1.1:3DOverlapping:0.1:Identity"*

5.3.2 Temporal metric:

M3.2: Number of different ID that can take a reference data ($NumObjectID_{\mathrm{Re}\,fdata}$). The different detected objects ID are identified with their bounding box and the fact that their intersection intervals with the reference data are disjointed

$$Persistence = \frac{1}{\#(\mathrm{Re}\ fdata)} \sum_{\mathrm{Re}\ fdata} \frac{1}{NumObjectID_{\mathrm{Re}\ fdata}}$$

The higher the persistence is (best is 100%), the better the persistence of the ID is.

<u>Example:</u>

*MetricTemporal "Mono:**M3.2**:M3.2.1:DiceCoefficient:0.3:Identity"*

M3.4: Percentage of time during which a reference data is detected and tracked.

$$T_{tracked} = \frac{1}{\#(\mathrm{Re}\ fData)} \sum_{\mathrm{Re}\ fData} \frac{\#\left(RD \bigcap C\right)}{\#(RD)}$$

<u>Example:</u>

*MetricTemporal "Mono: **M3.4**:M3.4.1: DiceCoefficient:0.3:Identity"*

M3.5: Number of different ID that can take a detected object ($NumObjectID_{DetectedObject}$). The different reference data ID are identified with their bounding box and with the fact that their intersection intervals with the physical object are disjointed. To transform the result in a percentage, we will compute the inverse.

$$Confusion = \frac{1}{\#(DetectedOb\ jectMatch\ \mathrm{Re}\ fData)} \sum_{DetectedOb\ jectMatch\ \mathrm{Re}\ fData} \frac{1}{NumObjectI\ D_{DetectedOb\ ject}}$$

The more the confusion is close to 100%, the more the algorithm is robust to confusion along the time.

<u>Example:</u>

*MetricTemporal "Mono:**M3.5**:M3.5.1:DiceCoefficient:0.3:Identity"*

5.3.3 Event metric:

M4.1: Number of correctly recognized events matching to reference data, for each event type.

<u>Example:</u>

*MetricEvent "**M4.1**:M4.1.1"*

M4.2: Difference of the beginning and ending event time of the detected event matching to reference data event for each event type

<u>Example:</u>

*MetricEvent "**M4.2**:M4.2.1:Duration:10"*

M4.3: Common frames between the detected events matching to reference data events

<u>Example:</u>

*MetricEvent "**M4.3**:M4.3.1:Frame:10"*

M4.4: Common duration between the detected events matching to reference data events

<u>Example:</u>

*MetricEvent "**M4.4**:M4.4.1:Duration:10"*

These metrics can be refined by using the filter techniques of the previous criterion. Only events which occur in certain zones can be taken into account (e.g., zones close or far from the camera, detection in a dark or noisy zone). The characteristics of physical objects of interest can bee also taken into account (e.g., low speed, important size, numerous interactions, strong contrast, large density of people).