Chapter 18

Qualification and Evaluation of Performances

18.1. Introduction

In order to design and commercialize smart video-surveillance systems, we need to properly understand the domain of function of the system in question. This involves evaluating the performances of the video analysis algorithms implemented, to check that the system's performance meets the customer's expectations, measure how advanced the system is, compare it with other commercial systems and take account of the legal constraints and/or norms. In addition, there is also a need at the scientific level to qualify and quantify the progress made in the field of computer vision.

Numerous initiatives have emerged with a view to comparing systems on the basis of common functional requirements with shared evaluation protocols and data, the main characteristics of which will be presented in section 18.2. However, the typical approach to evaluation, which is to select a video data set and annotate it by associating with a ground truth – in other words, manually creating the data that we wish the system to output – presents a number of problems. In particular, the process of evaluation depends closely on the choice of the test video sequences, metrics and ground truths, and it is often impossible to predict the performance of an algorithm if one of these parameters changes.

Chapter written by Bernard BOULAY, Jean-François GOUDOU and François BREMOND.

Launched as part of the collective program Techno-Vision supported by the French government and bringing researchers and industrial players together, the ETISEO competition [NGH 07b] has enabled significant progress to be made, proposing – in addition to annotated video sequences – metrics dedicated to a specific task, and tools to make evaluation easier. These metrics and tools are presented in section 18.3, along with the consequences and the avenues for improvement identified in this program.

Although it has been looked at in ETISEO, objective qualification of an algorithmic solution in terms of quantifiable factors (such as the contrast of an object) remains an open-ended and under-examined problem even today. In section 18.4, we describe an approach that could offer progress in this direction.

Finally, we will briefly present (in section 18.5) the research program QUASPER R&D (Qualification et certification des systèmes de perception, www.systematic-paris-region.org/fr/projets/quasper-rd, 2010–2012), launched recently, whose aim is to define the scientific and technical knowledge required to set up a platform by which to qualify and certify video perception systems.

18.2. State-of-the-art

There are many initiatives in place to evaluate performances in this area. These are evaluation programs, such as Context Aware Vision using Image-based Active Recognition (CAVIAR, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/), or workshops, such as PETS [PETS 10], which provide video databases to help compare systems, evaluating the performances of the same functionality applied to the same data set. Research programs such as Video Analysis and Content Extraction (VACE, www.informedia.cs.cmu.edu/arda/vaceI.html), Call for Real-Time Event Detection Solutions (CREDS for enhanced security and safety in public transportation) and Classification of Events, Activities and Relationships (CLEAR): evaluation campaign and workshop (www.clear-evaluation.org/), in addition to the videos, offer a set of metrics to evaluate the performances of various algorithms. In the following section, we will define what exactly a smart video analysis system is, and will focus on the evaluation of such systems.

18.2.1. *Applications*

Today, there are many so-called video intelligent systems (VISs). These systems extract different kinds of information from a video stream for various applications, ranging from activity recognition or object detection to detection of specific events or complex behaviors, to statistical analysis. Of the main applications, we can cite

Comment [ISTETrans1]: The author insists this is Video Intelligen***t*** System. However, my research suggests it is Video Intelligen***ce*** System. I've changed it back to "intelligent" throughout, in accordance with his wishes, but this doesn't seem to make sense to me – can you watch out for this?

motion detection, detection of degradation or malfunction of cameras, road traffic monitoring, people counting, crowd monitoring, person flow analysis, face recognition, intrusion detection, object tracking and detection of unattended bags. All these applications are based on a set of functions that perform a collection of tasks at different semantic levels: detection is the act of finding an entity (an object or an event); classification is the act of matching the detected entities with a category (e.g. person, vehicle and luggage); and tracking is the act of maintaining the same identity for the same entity over a period of time. Characterization is the act of extracting representative features of the entities (e.g. color, shape and duration); this act is used in the previous three tasks.

The most common approach is to arrange these functions into a "bottom-up" sequence, from the pixel level to the event level, as shown diagrammatically in Figure 18.1.

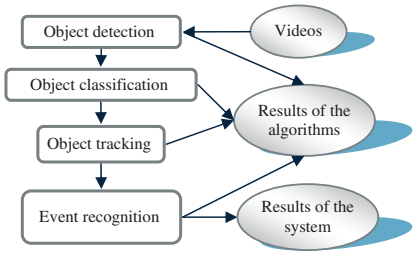

Figure 18.1. *Simplified diagram of the different stages in a video intelligent system*

18.2.2. *Process*

The conventional process for evaluating a VIS is shown in Figure 18.2. It relies on the use of databases of videos, ground truths and evaluation criteria.

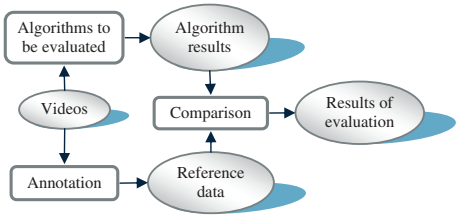

Figure 18.2. *Process for evaluating the algorithms of a video intelligent system*

18.2.2.1. *Video databases*

Video analytics systems must be tested on different video databases; the constitution of these databases is a determining factor in the quality of the evaluation. They must also be as representative as possible of realistic situations for the system being tested. If the context is variable and complex (e.g. if the system has to work both indoors and outdoors), we need to have a number of test sequences representing these variations. In video analytics, these variations may be due to the weather conditions (snow, rain, wind, sun, etc.), the changes in lighting (high noon, morning, nighttime, etc.) or indeed the movements of the camera. Unfortunately, the amount of evaluation data is often limited, because the tasks of acquiring and annotating these data require a great deal of time and human effort. In addition, this data set usually has to be able to be split into two subsets, both independent and complete, one to configure the algorithms (particularly if we are using algorithms based on automatic learning) and the other for evaluation.

There are many video databases in existence to evaluate a system in terms of its functionality: gesture recognition, crowd behavior, activity recognition, group tracking, people tracking, etc. These databases are shared, particularly between performance evaluation workshops such as PETS [PETS 10] (recognition of crowd behavior, see Figure 18.3) and Advanced Video and Signal-based Surveillance (AVSS, www.eecs.qmul.ac.uk/~andrea/avss2007_d.html) (abandoned baggage scenario, face detection, etc.). The CAVIAR project (http://homepages.inf.ed.ac.uk/rbf/CAVIAR/) offers a set of video sequences also containing annotations for the tracking and behavior of the individuals.

Figure 18.3. *Different shots from the PETS database for recognition of crowd behavior*

18.2.2.2. *Annotation tools*

Annotation, or ground truth, describes the true properties of a video sequence. This annotation may be extremely detailed and therefore the process may be very long. The annotation may precisely describe the form of the objects of interest (to the nearest pixel), or place a simple bounding box around them. It may also describe the properties associated with the object: its velocity, its color, its pose and many

other aspects. It may also contain the detailed description of the events taking place in the sequence, to fraction-of-a-second precision. There are many ways to acquire this ground truth:

– Manual acquisition (by far the most widely used method): A human observer annotates the videos, denoting the position of the object by a point, a contour or a bounding box. This approach is painstaking and introduces a certain bias, because two people annotating the same sequence will not produce exactly the same data [KAS 09]. Annotation rules are necessary in order to limit this bias.

– Semiautomatic acquisition: Standard video analysis algorithms are used to produce an initial annotated data set. The results are then corrected by a human operator, and the ground truth is constructed based on these corrections. The ground truth thus create is specific to the output of the algorithm being used, and can therefore be used only to evaluate systems that are compatible with this output.

– Use of auxiliary sensors: In acquiring data, sources other than video cameras are used, e.g. position sensors for the objects of interest, distance sensors, infrared barriers or indeed presence sensors. This approach means that the acquisition system is complex (synchronization and spatial referencing) and it is impossible to automate the annotation of all the properties.

– Synthesis of video data: Test videos are generated using image synthesis or augmented reality technology. The ground truth is then generated automatically at the same time as the images, and corresponds very precisely to the objects being handled. The main drawback to this approach is that to date the data obtained still lack realism, in spite of the very great advances made in the past few years.

It should be noted that some evaluation metrics exist that do not need a ground truth. For instance, [ERD 04] proposes to add noise to the data and verify that the result remains the same. Thus, this approach does not give a genuine qualitative evaluation of the system's performances, but rather a measurement of its stability.

As yet, there are no entirely automated annotation tools. However, many tools have been put forward to facilitate and accelerate the process of annotating video sequences. Viper (http://homepages.inf.ed.ac.uk/rbf/) is a tool that is most widely used among the computer vision research community. It is able to annotate the objects of interest with bounding boxes. The user can define the attributes to be annotated for each object. An interpolation function is available to speed up the drawing of the bounding boxes (the operator draws the boxes at two times: t and $t + n$); however, this interpolation is simply performed on the basis of the size of the box, rather than using the information available in the image. The ground truth is exported in XML format. The tool has not been updated since 2004. Furthermore, although it is possible to annotate events, the tool is not optimized for this task. Anvil (www.anvil-software.de/) is a video annotation tool primarily used by

Comment [A2]: AQ. Please provide a URL which is directly linking to « Viper ».

researchers in social sciences. It enables us to annotate events using a timeline. Users are able to add their own plugins in order to annotate their own objects of interest. For instance, a plugin was developed to annotate people, using calibration of the camera. In [JAY 02], the authors propose an Open Development for Video Surveillance (ODVIS) system. This system offers an application programming interface (API) to evaluate tracking algorithms. The CAVIAR project (http://homepages.inf.ed.ac.uk/rbf/CAVIAR/) proposes an annotation tool written in Java, the source code for which is available on the project's Web site.

To conclude our discussion of annotation, it should be noted that there are no standards defining the content and format of a ground truth. To use ground truths produced by different tools, a data conversion is, therefore, needed.

18.2.2.3. *Measures of performance*

Mostly, the ground truth is made up of objects and/or events that vary over time. Evaluation consists of measuring the resemblance between the ground truth and the information obtained by the video analysis algorithms, using evaluation criteria in order to do so. Most qualitative results of evaluation are expressed in terms of the following basic measurements:

– The rate of true positives (TP) represents the number of detections (objects or events) that corresponds to entities in the ground truth.

– The rate of false negatives (FN) represents the number of entities in the ground truth that does not correspond to a detection.

– The rate of false positives (FP) represents the number of detections that does not correspond to an entity present in the ground truth.

– The rate of true negatives (TN) represents the number of non-detections that corresponds to non-annotations. This number cannot always be calculated.

In order to be able to interpret the result of the evaluation, metrics are calculated based on the basic measurements: precision (P) and sensitivity (S) (also called recall rate), for which the calculation formulae are given in [18.1]. The F-score (F) represents the harmonic mean between precision and recall:

$$P = \frac{TP}{TP + FP}; \quad S = \frac{TP}{TP + FN}; \quad F = 2\frac{P * S}{P + S} \qquad [18.1]$$

Metrics that are more specific to the application being evaluated can be calculated. In [ZIL 05], the authors propose a measurement called CREDS to evaluate video analytics algorithms in a competition organized by the RATP

(Parisian Metro operator). The CREDS metric uses the basic measures (TP, FP and FN) to judge the quality of detection, but attributes *bonuses* and *maluses* that are defined according to the application and the users' needs.

The basic measurements are not always applicable – particularly when the evaluation relates to a physical measurement, e.g. the number of people or the speed of a vehicle. For a quantitative evaluation, the notion of bias is used. The bias represents the difference between the expected value (α_i) and the measured value (d_i). The mean bias (*MB*) and bias dispersion (*BD*) are usually calculated by the formulae [18.2]:

$$MB = \frac{1}{n}\sum_{i=1}^{n}\left(d_i - a_i\right); \quad BD = \sqrt{\frac{\sum_{i=1}^{n}\left(\left(d_i - a_i\right) - MB\right)^2}{n-1}} \qquad [18.2]$$

18.3. An evaluation program: ETISEO

Unlike the evaluation programs mentioned above (VACE, www.informedia.cs.cmu.edu/arda/vaceI.html, CREDS [ZIL 05], CLEAR, www.clear-evaluation.org/) that focus only on the users' point of view, the research program ETISEO [NGH 07b] is aimed at helping algorithm developers identify weaknesses by highlighting the dependency between the algorithms and their conditions of use. The main idea of ETISEO is to evaluate video processing algorithms by focusing on one task in the process (such as detection or tracking of objects), depending on the type of sequence (e.g. road traffic scenes) and a general obstacle (e.g. the presence of shadows).

18.3.1. *Methodology*

The methodology of ETISEO is based on the following principles:

– *Typology of the tasks*: ETISEO identifies four tasks in algorithmic processing, which correspond to the main stages of video analytics systems shown in Figure 18.1: object detection (task 1), object tracking (task 2), object classification (task 3) and event recognition (task 4).

– *Typology of the problems*: ETISEO separately addresses different problems duly defined and classified. For instance, the problem of shadows can be divided into several subproblems: (1) shadow with different levels of intensity (slightly or heavily contrasted); (2) shadow with the same level of intensity but with a different

background in terms of colors or textures; and (3) shadow with different light sources in terms of position or wavelength.

– *Compilation of a video database that is representative of the problems*: Each video sequence is specific to a problem. For example, for the problem of shadows, videos have been selected for different intensities of shadows (low and high). The database thus compiled of over 40 scenes with one or more cameras contains 85 video sequences. These videos include scenes from the subway, the street, airport aprons and building entrances or corridors (see Figure 18.4).

– *Annotation of the videos*: Three kinds of data are collected for each sequence: (1) the ground truth, including the annotations necessary for the four identified tasks (bounding boxes on the objects, type of objects, events, etc.) and produced using the Viper tool presented in section 18.2; (2) the particular difficulties of the video (e.g. the presence of a slight shadow) and the acquisition conditions (for instance, the weather conditions); and (3) the camera calibration parameters and the topology of the scene (e.g. the zones of interest that make up the scene).

– *Evaluation metrics*: ETISEO has defined different metrics to evaluate each of the tasks identified. These metrics are detailed in section 18.3.2.

– *Analysis of the sensitivity of the algorithms*: ETISEO offers a tool to automatically evaluate and analyze an algorithm's behavior in the face of the problems under consideration. The tool is still available, but a new version – Visualization and evaluation tool (ViSEvAl) – is available under AGPL license (see section 18.3.3.3 for details).

Comment [A3]: AQ. Please provide the expanded form of "AGPL" at first occurrence.

Figure 18.4. *Different environments constituting ETISEO's video database: airport apron, building entrance, subway and street*

18.3.2. *Metrics*

In ETISEO, most of the metrics require us to match the detections with the ground truths. The matching may be spatial (bounding boxes associated with the objects) or temporal (time interval associated with the events). In order to qualify the match between a detected object and an object from the ground truth (reference object), four measures of matching (similarity or dissimilarity) have been defined (formulae [18.3]): the Dice coefficient (D_1), the overlapping measure (D_2), the Bertozzi coefficient (D_3) and the maximum deviation measure (D_4):

$$\begin{aligned} D_1 &= 2*\frac{\#(RD\cap C)}{\#(RD)+\#(C)}; \quad D_2 = \frac{\#(RD\cap C)}{\#(RD)}; \\ D_3 &= \frac{\left(\#(RD\cap C)\right)^2}{\#(RD)*\#(C)}; \quad D_4 = \max\left\{\frac{\#(C\setminus RD)}{\#(C)}, \frac{\#(RD\setminus C)}{\#(RD)}\right\} \end{aligned} \qquad [18.3]$$

where # expresses the surface (number of pixels), *RD* (reference data) the annotated 2D box and *C* the detected 2D box.

The two objects are matched in the sense of a measure of matching if the value of that measure is greater (for D_1 to D_3) or lesser (for D_4) than a predetermined threshold. In addition, the result of the metrics is calculated by giving the measures of precision, sensitivity and the *F*-score.

18.3.2.1. *Metrics for object detection (task 1)*

To evaluate object detection, ETISEO proposes five metrics:

– The metric "number of objects" looks at the number of objects detected (called blobs) that corresponds to the reference objects by comparing their bounding boxes. The main advantage to this metric is that it does not prioritize large blobs as pixel-based metrics do, because it focuses only on the number of objects. However, because the matching computation uses a threshold, it is not possible to distinguish detected objects that overlap the reference objects by 120% from those that overlap them by 100%. The following metric was therefore introduced to differentiate these cases.

– The "area of the object metric" calculates the number of pixels in the reference data that has actually been detected.

– The "split metric" measures the fragmentation of the detected object. It calculates the number of objects detected per reference object using the overlapping measure (D_2).

– The "merge metric" measures the merging of the detected objects. It calculates the number of reference bounding boxes that corresponds to a detected object (in the sense of D_2).

– The "2D/3D distance metric" qualifies the location of the objects detected. It measures the mean of the 2D/3D distances between the centers of gravity of the objects detected and the corresponding annotated objects. Unlike the "area of the object metric", this metric is not biased by the size of the objects. There is a difficulty in calculated the 3D center of gravity of an object because there is no consensus.

18.3.2.2. *Metrics for object tracking (task 2)*

The task of object tracking is evaluated using one primary metric (tracking time) and two additional metrics (object ID persistence and object ID confusion: a single number associated with an object throughout its entire "lifetime" in the sequence).

The metric "tracking time" (equation [18.4]) measures the percentage of time for which a reference object RD_t is matched (in the sense of the matching measures discussed above) to a tracked object C_t by comparing their bounding boxes. $\#(RD_t \cap C_t)$ corresponds to the period of time for which the detected object corresponds to the annotated object and $\#(RD_t)$ corresponds to the lifetime of the annotated object. The main characteristic is that the evaluation can only be performed using previously detected objects. Otherwise, detection errors would influence the evaluation of the tracking performance:

$$T_{\text{tracking}} = \frac{1}{Nb_{RD}} \sum_{RD} \frac{\#\left(RD_t \cap C_t\right)}{\#\left(RD_t\right)} \qquad [18.4]$$

The additional metrics qualify the precision of the tracking. The metric "object ID persistence" (equation [18.5a]) examines the whole video sequence, looking for how many tracked objects are matched with a reference object $(Nb_{ObID_{RD}})$. However, this metric favors under-detection. For example, using this metric, an algorithm that tracks an object for a short period of time is given a higher evaluative score than another algorithm that tracks the same object for twice as long but with two different identifiers. Conversely, the metric "object ID confusion" (equation [18.5b]) calculates the number of annotated objects associated with a detected object (Nb_{ObID_C}). The disadvantage with this metric is that it favors over-detection. In particular, if an algorithm detects several objects for the same annotated object, it will obtain a high evaluative score because each object detected will correspond to at most one annotated object. Also, the fact that an algorithm obtains a good score with these latter two metrics does not necessarily prove the quality of the algorithm.

The three metrics must be used in conjunction in order to qualify the performances of a tracking algorithm.

$$\Pr ec = \frac{1}{Nb_{RD}} \sum_{RD} \frac{1}{Nb_{ObID_{RD}}} \quad [18.5a]$$

Comment [A4]: AQ. Please check « Pr *ec* » in equation [18.5a].

$$Conf = \frac{1}{Nb_{ObMatchRD}} \sum_{RD} \frac{1}{Nb_{ObID_C}} \quad [18.5b]$$

18.3.2.3. *Metric for object classification (task 3)*

The task of classification is evaluated using a metric that compares the type of the object detected to that of the corresponding annotated object, i.e. whose bounding boxes are matchable from the point of view of the matching measure chosen.

18.3.2.4. *Metric for event recognition (task 4)*

Event recognition is evaluated by a metric that compares the names of the annotated and detected events. The matching of the events is performed using a time distance, which calculates the time common to two time intervals. A threshold is thus defined to see whether the detected event corresponds in terms of time to an annotated event. Semantic comparison is performed by comparing the names of the detected and annotated events. If the comparison is positive then the event has been correctly recognized.

18.3.3. *Summary*

18.3.3.1. *Main consequences*

The ETISEO project (Video Understanding Evaluation, www-sop.inria.fr/orion/ETISEO/) enabled good practice rules to be put in place for evaluation, particularly in relation to the following points:

Comment [A5]: AQ. Please check the insertion of the « ETISEO project web address » here.

– ETISEO set up an extensive database and collection of metrics to evaluate video analytics algorithms. For each task to be evaluated, a primary metric is able to provide a global evaluation of the candidate algorithm, and addition metrics qualify its precision.

– ETISEO offers two ontologies to facilitate communication between the different actors: researchers, developers and end users. The first ontology describes the technical concepts used in a video understanding chain (e.g. a blob and a trajectory) and the concepts associated with the evaluation (such as the reference

data). The second ontology relates to the concepts of the domains of application of the videos (for instance, the event “opening a door”).

– ETISEO’s automatic evaluation tool makes it easy to compare the results of the algorithms and the reference data. For example, filters can be used to select and evaluate only a certain kind of data (e.g. stopped objects). Also, this tool allows us to visualize the results and the ground truth for a given video sequence.

– ETISEO’s evaluation took place in two phases. In the first phase, the participants were able to test their algorithms on a small data set, helping them to better understand and make better use of the data. This also enabled us to adjust the metrics and tools based on feedback from the participants. The second phase of evaluation is the final evaluation.

– ETISEO enabled us to evaluate algorithms in complex situations (e.g. a crowd scene) up to recognition of events of interest (detection of abandoned luggage, for instance).

18.3.3.2. *Avenues for improvement*

ETISEO also provided an opportunity to look again at certain problems, which can be highlighted during the “competition” phase of the program, which was necessary to solve in order to design an industrially viable evaluation methodology:

Comment [A6]: AQ. Please check the edited sentence [ETISEO also provided an opportunity to look again at certain problems …] for clarity.

– There are certain inconsistencies between the different partners, particularly as regards the definition of objects of interest and events. For instance, the partners use a stationary object differently. For some, if the object remains stationary for a certain length of time, then it is integrated into the model of the background and is no longer detected. For others, the object is maintained under surveillance until the end of the sequence, as was required by the competition. This makes it difficult to compare the results of the different participants’ algorithms. Therefore, a filter was defined so that stationary objects are not taken into account in the evaluation.

– ETISEO gives no time limit for the computation of the results to satisfy the real-time constraint. Also, no counter indication was given to the participants to retain the same parameters of the algorithms for all the video sequences or at least the same type of scene.

– It is difficult to compare different algorithms on the basis of the evaluative score. If the *F*-score of an algorithm differs from that of another by 0.1, is that really so significant? There is no absolute response to this question, because it depends on the application.

– ETISEO referred the estimation of problems presented by a video, but such an estimation is very crude. For instance, ETISEO uses the terms “normal” or “dark” to describe the level of light intensity in a video sequence, but the selection of these

terms is subjective and imprecise. Also, in order to predict the performance of an algorithm on a new sequence, this new sequence needs to be compared with those that make up ETISEO's database. We describe a method to automatically measure the level of difficulty in a video in section 18.4.

18.3.3.3. *Evolutions*

The tool developed during the ETISEO competition was later improved by the *Stars* team at INRIA. ViSEvAl (www-sop.inria.fr/teams/pulsar/EvaluationTool/ViSEvAl_Description.html) was developed so that users could contribute, e.g. by adding their own evaluation metrics. Written in C++, the tool implements a system of plugin interfaces to help users contribute simply. It enables us to view the detections (objects and events) and annotations on the images, the videos or in a 3D virtual world (see Figure 18.5). The tool also manages multi-camera captures. The tool is under AGPL license and can be downloaded for free.

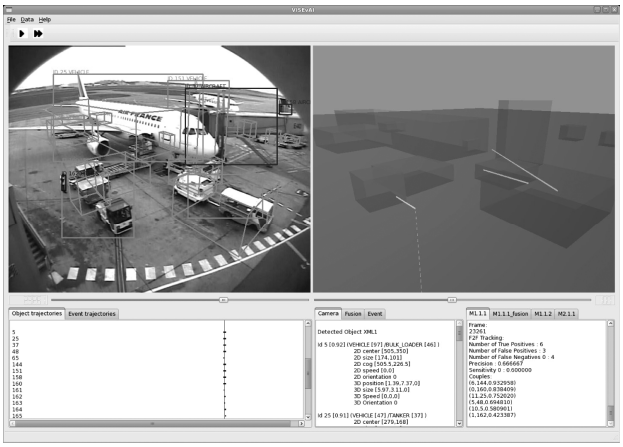

Figure 18.5. *The ViSEvAl tool allow users to view the result of the algorithms and the ground truth*

Comment [A7]: AQ. Please check the edited caption of Figure 18.5.

18.4. Toward a more generic evaluation

In [NGH 07a], the authors propose a methodology based on that of the ETISEO to evaluate video processing algorithms on new sequences. The aim is not to predict an algorithm's performance on the new sequence in question, but rather to estimate an upper limit of the algorithm's performance in terms of a specific factor (e.g. the

contrast) because an algorithm's performance also depends on other factors, such as the size of the objects and lighting changes. Thus, it is a question, for a given factor, of identifying the domain of variation in which we know that the algorithm's performances will be unsatisfactory. In other words, for a given algorithm, we cannot estimate the *sufficient* conditions for its success, but we can estimate the *necessary* conditions.

The implementation of this approach requires six elements:

– the results of different algorithms for the same task;

– the video processing problem to be characterized (e.g. poor contrast);

– the evaluation metrics;

– the measures of the input data that depend on the problem to be characterized;

– the reference data;

– the video sequences illustrating the problem.

The authors suppose that the results of the different algorithms are representative of their performances. In other words, they suppose that the users are able to parameterize their own algorithms to obtain representative results. Of the six elements listed above, there are only two that are not provided either by the writers of the algorithms or drawn from the video databases: the measures of the input data and the evaluation metrics.

In order to illustrate this methodology for evaluation, let us give an explicit example of the approach for two video processing problems: the handling of poorly contrasted objects and the handling of objects with shadows.

18.4.1. *Contrast*

The performance of video analytics algorithms is generally proportional to the level of contrast between the objects to be detected and the background of the image. The lower the level of contrast, the poorer the algorithms' performances will be. Therefore, we will determine the level of contrast beyond which a given algorithm can deliver an acceptable performance.

For each image in the video database and for each pixel of that image, we use the notation (R_f, G_f, B_f) to represent the color of the foreground pixel in the RGB space and (R_b, G_b, B_b) to represent the color of the corresponding pixel in the associated estimated background image. In addition, the objects are supposed

to be segmented in the image. For each pixel, the contrast level *Cont* is determined by equation [18.6]:

$$Cont = \frac{\left|R_b - R_f\right| + \left|G_b - G_f\right| + \left|B_b - B_f\right|}{K_{\text{Dyn}} * 3} \qquad [18.6]$$

where K_{Dyn} is a normalization factor dependent on the dynamic of the image signal.

The approach in question consists of dividing each object of interest *O* into regions *R*, the shape of which depends on the type of object in question and the size depends on the size of the object. These regions are themselves divided into rectangular subregions *SR* of fixed dimensions. The contrast of each subregion is then defined by the quantified mean contrast of the pixels that makes it up, the contrast of each region is defined by the maximum contrast of the subregions making it up, and each object is characterized by ensemble of the contrasts of the regions that make it up (the duplicate values are removed).

18.4.1.1. *Application to the characterization of a detection algorithm*

For the task of object detection, the system's capacity to handle poorly contrasted objects is calculated by using the rate of detection errors, R_{de}, for each quantified level of contrast. For this calculation, the evaluation space has changed. Instead of considering the objects in their entirety, we consider all the homogeneous subregions, i.e. whose pixels present the same level of contrast. For a given level of contrast *c*, we use the notation *a*(*c*) to denote the total number of regions, and *x*(*c*) for the number of regions detected by an algorithm. The rate of detection errors $R_{de}(c)$ of the algorithm is then given by equation [18.7]:

$$R_{de}(c) = 1 - \frac{x(c)}{a(c)} \qquad [18.7]$$

and the detection algorithm's capacity to deal with poorly contrasted objects corresponds to the lowest level of contrast for which the rate of detection errors R_{de} is below a certain threshold.

18.4.1.2. *Application to the characterization of a tracking algorithm*

Object tracking algorithms can track an object if and only if, in most of the images, the system is capable of detecting all the regions that make that object up. The difficulty in tracking of an object is therefore characterized by the minimum level of contrast between the regions of which it is formed. Given that the

algorithm's performance is calculated by the tracking metrics described in section 18.3.2.2, the capacity to track poorly contrasted objects is defined as the lowest level of contrast for which the algorithm's performances are greater than a certain threshold.

18.4.2. *Shadows*

When a scene contains a high-intensity light source (the sun, a lamp, etc.), the objects are often detected with their shadows. The algorithms have difficulty distinguishing between the object and the shadow, because the contrast between the shadow and the background is often very high. In addition, the whole shadow or parts of it are mixed with the object itself. By the same approach as for the contrast of the object (but using specific geometries for the regions and subregions) the contrasts of the shadows are characterized for all significant object shadows in the video database. Then, the impact of the shadows is measured, for a candidate algorithm devoted to a given task (detection, tracking, etc.), by determining the minimum shadow contrast for which the algorithm's performances prove acceptable.

18.5. The Quasper project

The Quasper project relates to the evaluation of perception systems – particularly video analytics systems – but with the goal of standardizing a number of clearly defined applications: intrusion detection, pedestrian detection in onboard surveillance. The reflections in this project relate to the testing methodologies, the reference framework and the physical, hardware and software resources needed to evaluate entire perception systems (sensors, networking equipment, machines and software layers), to offer a test site both for the providers of perception systems and for their customers, who do not always have the means to compare the offerings of different solutions. Quasper brings together partners from academic and industrial spheres. The two main fields of application are security (video-surveillance system) and the automotive industry (onboard systems in vehicles).

The aim of Quasper is to offer a platform for evaluating the performances of perception systems. It is targeted specifically at multisensor perception systems, instead of those systems that are limited just to video cameras. The Quasper R&D project, which defined the methodologies to be used on the platform, also aims to put forward these methodologies as European standards for qualification of the performance of perception systems (in terms of ontology, metrics and tools). In particular, a great deal of work is being carried out as regards multicriteria analysis: how are we to combine the result of multiple metrics evaluating different aspects of a system's functions? The philosophy of the Quasper platform is the same as that

Comment [A8]: AQ. Please check the edited sentences [The aim of Quasper is to offer a platform … performance of perception systems (in terms of ontology, metrics and tools).] for intended meaning.

which motivated the car safety program European New Car Assessment Programme (EuroNCAP), to construct a series of relatively simple tests that are meticulously documented so as to be perfectly reproducible on different sites, corresponding to clearly defined functional needs, and based on a set of videos, sensors and stimulation systems (such as a weather effect simulation chamber) enabling us to test any kind of perception system offering the particular function.

Comment [ISTETrans9]: I've checked with the author, and this is not a typo: he's talking about systems which provide stimuli to the sensors.

18.6. Conclusion

In this chapter, we have presented a state-of-the-art on the methodologies used to evaluate the performances of a video-surveillance intelligent system. These methodologies are based on a common approach that consists of selecting a video database that is as representative as possible of the problem at hand, annotating that database by associating a ground truth with it and using evaluation metrics that measure a difference between the ground truth and the results obtained by the system being evaluated. This approach exhibits some limitations, because the evaluation process depends heavily on the choice of test video sequences, metrics and ground truths, and it is often impossible to predict an algorithm's performance when one of these parameters changes. In addition, the evaluation usually relates to the output of the system, without analyzing the contribution of the different analytical tasks performed to obtain this result.

In order to get around these limitations, the ETISEO program put in place a methodology and tools that enable users not only to evaluate the overall performances of the system being analyzed, but also to measure the performances of the different analytical tasks performed and characterize the domains of the system performance and its analytical tasks using, on the one hand, a typology of the scenes dealt with (e.g. a street scene and an airport arrivals hall) and, on the other hand, a graded typology of the problems dealt with (e.g. the presence of powerful shadows and high contrast).

ETISEO enabled significant advances to be made in the field of evaluation of the performance of video-surveillance intelligent systems. However, there is still some way to go before we see fully exploitable evaluation methods. To begin with, a standardization effort must be carried out, so that all the stakeholders (researchers, developers and users) actually work with the same concepts, defined by a common language. Then, the compilation and annotation of video databases are extremely costly tasks that are still contingent on the subjectivity of the operators. Hence, it is greatly desirable to move toward better automation of these tasks. Finally, it is desirable to be able to characterize the performances of a candidate algorithm for a given task, based on measurable factors (e.g. the level of contrast of the objects of

interest). In this chapter, we have briefly presented an approach that addresses this latter topic.

These avenues for improvement will be reexamined and supplemented in the project Quasper R&D, which aims to define the methodologies to be put in place in the context of a platform to evaluate the performances of perception systems.

18.7. Bibliography

[ERD 04] ERDEM C.E., SANKUR B., TEKALP A.M., "Performance measures for video object segmentation and tracking", *IEEE Transactions on Image Processing*, vol. 13, no. 7, 2004.

[JAY 02] JAYNES C., WEBB S., STEELE R.M., XIONG Q., "An open development environment for evaluation of video surveillance systems", *PETS*, Copenhagen, Denmark, 2002.

[KAS 09] KASTURI R., GOLDGOF P., SOUNDDARARAJAN V., MANOHAR J., GAROFOLO J., BOWERS M., BOONSTRA, KORZHOVA V., ZHANG J., "Framework for performance evaluation of face, text and vehicle detection and tracking in video: data, metrics, and protocol", *IEEE Transactions on PAMI*, vol. 31, no. 2, pp. 319–336, 2009.

[NGH 07a] NGHIEM A.T., BREMOND F., THONNAT M., MA R., "A new approach for video processing algorithms", *WMVC*, Austin, TX, 2007.

[NGH 07b] NGHIEM A.T., BREMOND F., THONNAT M., VALENTIN V., "ETISEO, performance evaluation for video surveillance systems", *AVSS*, London, UK, 2007.

[PETS 10] *IEEE 13th International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, available at: http://pets2010.net/

[ZIL 05] ZILIANIL F., VELASTIN S., PORIKLI F., MARCENARO L., KELLIHER T., CAVALLARO A., BRUNEAUT P., "Performance evaluation of event detection solutions: the CREDS experience", *AVSS*, Como, Italy, 2005.

Comment [A10]: AQ. The web addresses provided under the heading « other references » have been deleted from the lists and added (web addresses) to the text at their respective places. Please check.

Comment [A11]: AQ. Please provide page range for the referred journals in the reference [ERD 04].

Comment [A12]: AQ. Please provide complete conference title and date(month) of the proceedings/conferences in the references [JAY 02, NGH 07a, NGH 07b, ZIL 05].