4

# Video Surveillance of Today: Compressed Domain Object Detection, ONVIF Web Services Based System Component Communication and Standardized Data Storage and Export using VSAF – a Walkthrough

Houari Sabirin[1] and Gero Bäse[2]
*[1]Bandung Institute of Technology,*
*[2]Siemens AG - Corporate Technology,*
*[1]Indonesia*
*[2]Germany*

## 1. Introduction

The growing trend for surveillance systems performing multiple functions (video surveillance, access control, response co-ordination) driven by technology integration and interoperability is emerging because organizations increasingly viewing integrated security systems as inevitable and necessity. In this paper we focus on the interoperability aspect for video surveillance systems.

An open standard has been developed by the Open Network Video Interface Forum (ONVIF) to facilitate the interoperability between networked surveillance video components. The ONVIF specification defines the network layer of IP security devices including automatic device discovery, video streaming and intelligent metadata. It is IP-based, makes use of web services and provides a formal conformance process. By employing the specification, interoperability is assured between products regardless of brand. End-users are enabled to compose the most suitable combination of products for their specific needs regardless of vendor and integration costs are reduced significantly. Conformance of products to the ONVIF specification is based on vendor self-declaration and applying test tools available for ONVIF members.

With the latest advance in compression technology such as MPEG-4 AVC|H.264, a joint standard from ISO/IEC (ISO/IEC 14496-10:2009) and ITU-T (ITU-T Recommendation H.264 (03/10)), the surveillance system provides better video quality and larger video resolution, while not stressing the network bandwidth further. Moreover, a novel export format fostering interoperability not only between different installations but also close cooperation with legal authorities “MPEG-A Part 10 Video surveillance application format” has been published by the Moving Picture Experts Group (MPEG). It specifies the use of MPEG-4 AVC|H.264 along with content description metadata e.g. object identification and other particularly surveillance related information neatly arranged.

The mission for this book chapter is to raise awareness for the specifications provided by standardization that ensure the interoperability of surveillance system as well as to introduce the surveillance system that combines low computational cost for video content analysis with cost-effective and flexible communication protocols for networked video. This chapter provides a walkthrough to the ONVIF video surveillance system, from the camera through analysis and storage right up to display and export. In addition and serving as key example for modern analytics a novel algorithm for object detection in compressed video data will be presented. It illustrates the general trend towards processing increasing amounts of data in real-time automatically by avoiding completely the task of decoding the video prior analysis.

## 2. Camera and web services

A camera is called NVT (Network Video Transmitter) in the ONVIF eco-system. It is left open how many (different) video streams a camera may generate. Therefore, also implementations combining video encoding for several analog cameras in one box are supported as NVT maintaining backward compatibility to legacy systems.

As can be seen from the available configuration entities listed below a NVT is much more than just a camera. Edge devices in a network are being equipped with ever increasing processing power. Thus, complexity of algorithms being executable on board of the device is growing also. ONVIF supports for example analytic algorithms at the camera as well as parallel processing of video data of different source areas of a megapixel-camera.

A wide range of Pan, Tilt and Zoom (PTZ) features are supported. Differentiation between absolute, relative and continuous move operations is also enabled as is a range of coordinate systems and settings for home positions and presets.

Picture data compression

For the encoding of video JPEG is the only required standardized format to be supported by every NVT. The ONVIF group did acknowledge the growing acceptance of MPEG-4 AVC|H.264 in the market by supporting it as video compression standard but has chosen JPEG as minimal conformance requirement.

Configuration of a NVT is condensed in a structure called “Media Profile”. A Media Profile can be regarded as a box of building bricks, combining the input of one configuration entity with the output of another, e.g. a video encoder configuration and the appropriate video source configuration.

Every NVT has to provide at least one Media Profile. The number of supported Media Profiles is limited by available on-board resources only.

Configuration entities may be generated once and reused in different profiles. Signaling is in place if changing attempts for a particular configuration entity may affect other profiles as well. Configuration entities are only present, if the respective device supports this capability. E.g. it is not required for each and every NVT to support audio decoding capabilities.

Depending on available computational and memory resources configurations may be dynamically made available for usage in media profiles by a NVT. Particularly the number of supported video encoder instances may change dynamically depending on the requested frame rate, resolution and compression format of profiles already in use.

In order to support users in the creation process of a Media Profile NVT’s provide information about compatible configurations to existing profiles, available options for configuration entities but may refuse adding an output configuration entity to a Media Profile if no appropriate source configuration entity has been added previously.

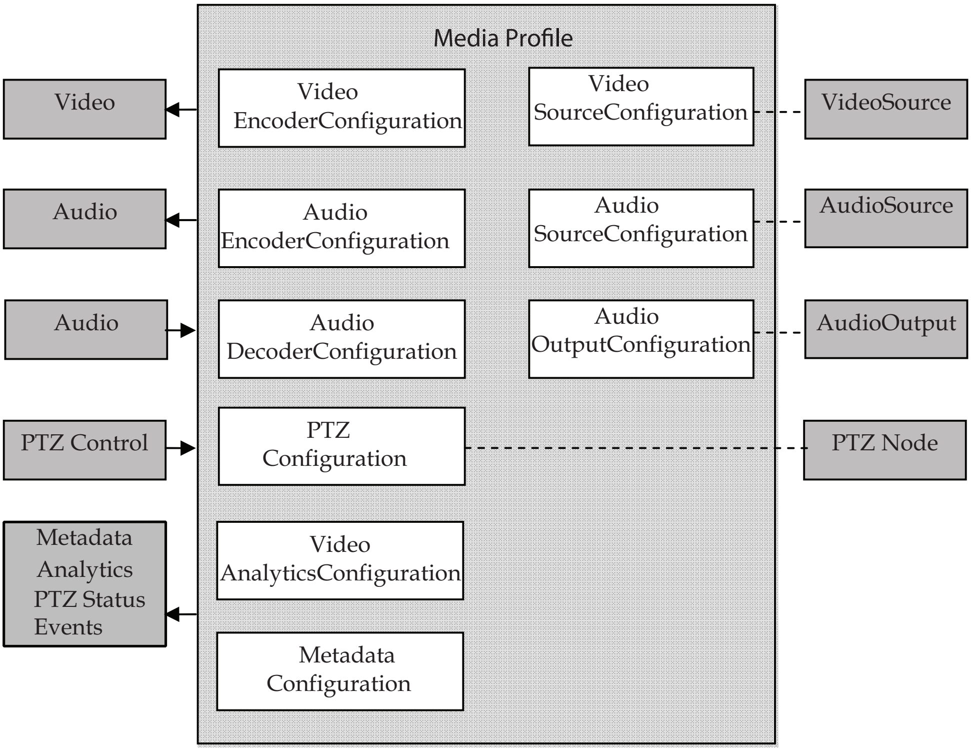

Media Profile
Video
Video EncoderConfiguration
Video SourceConfiguration
VideoSource
Audio
Audio EncoderConfiguration
Audio SourceConfiguration
AudioSource
Audio
Audio DecoderConfiguration
Audio OutputConfiguration
AudioOutput
PTZ Control
PTZ Configuration
PTZ Node
Metadata
Analytics
PTZ Status
Events
Video AnalyticsConfiguration
Metadata Configuration

Fig. 1. Media Profile (source ONVIF)

Each and every Media Profile provides a URI where media data processed according to the settings in the profile can be requested as live media stream.

All configuration services defined in the ONVIF standards are expressed as web services operations, which in turn are based on the Organization for the Advancement of Structured Information Standards (OASIS) web service standards framework, representing a basic design principle of ONVIF standards: to make use of already existing standards wherever possible. Web services allow for fast integration in different platforms.

Web services

Web services are providing the main components for client server operation: finding, binding and data exchange. Web services are intended for automated data exchange with or function calling on remote machines. The usage of web services provides for open and distributed architectures independent of platforms, programming languages and protocols. All devices in ONVIF are taking on the role of a web service provider. On top of HTTP (Hypertext Transfer Protocol) as transport mechanism, keeping firewall traversal issues as small as possible, SOAP (Simple Object Access Protocol) message exchange protocol is used for the communication between web service requester and provider. Integration on client side (web service requester) is simplified by providing WSDL (Web Services Description Language) files, which are normative in ONVIF. There are WSDL compiler tools available to generate platform specific code on a variety of platforms.

ONVIF relies on WS-Discovery standard techniques as common way to discover service providers. An exception here is the discovery proxy definition. ONVIF provides an alternative definition to allow for remote discovery in network architectures not fully supported by WS-Discovery. A discovery proxy is particularly necessary if service requester and provider do not reside in the same administrative domain of a network.

Security is of major concern for any installation using web services. ONVIF makes use of WS-Security framework for message level security. In order to allow for mutually authenticated transport session as well as preserving the confidentiality and the integrity at transport level usage of Transport Layer Security (TLS) protocols is recommended in addition.

All web services defined in ONVIF follow the Web Services Interoperability Organization (WS-I) basic profile 2.0 recommendations to assure best practice.

Every device in the ONVIF eco-system has to implement the *Device (web-) Service*. It allows a service requester to get an URL entry point of the NVT where all the necessary product specific WSDL and schema definitions can be retrieved. Furthermore, device specific parameter can be retrieved and set.

Embedded storage of a NVT may be utilized in an ONVIF conformant way by operating it under *Recording Service* (see paragraph 4) control. Here, the media profile is used as data source description. In case of network failure this mechanism may be used as temporary storage.

## 3. Data analysis

Thoroughly inspection of the video sometimes requires more processing power than available for analysis in a NVT. Additionally, analysis across video data originating in different NVT's, potentially at different instances of time, has to be performed server-based, also known as central or host-based analytics.

### 3.1 Analytics device and notification

In ONVIF the *Analytics Device Service* provides for the configuration and set-up of such dedicated devices and the *Analytics Service* is being used to configure analytic algorithms. Furthermore, compilation of rules by provisioning of a rule description language is supported. Such a device is being referred to as NVA (Network Video Analytics). If the analytic part of the NVA is composed of different vendor modules it can be encapsulate with the *Analytics Device Service*.

Input for a NVA is mainly video data in different configurations. Also, audio and metadata may be used for analytic purposes and can be input to a NVA if algorithms handling such kind of data have been installed. Provisioning exists for additional informative data to be conveyed to and used by analytic algorithms. On the other hand, different sets of parameter controlling the analytic algorithms can be used to enable e.g. day-night switching.

Composition of analytic algorithm modules into analytic engines is enabled. Single analytic engines may be composed to an analytics application. In a control instance for the *Analytics Device Service* all necessary information for such an application is condensed.

In addition, the state of a particular control instance can be inquired. The expandable structure allows for also conveying state information for substructures like analytic engines or even single analytic algorithm modules.

Generally speaking, analysis in ONVIF provides two different kinds of results. Events may be sent out addressing subscribers and a more comprehensive scene description may be generated. A XML schema has been defined covering basic scene elements and providing for easy extension of almost every single element to enable vendor specific data provisioning.

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="...www.onvif.org/...">
 <tt:VideoAnalytics>
  <tt:Frame UtcTime="2008-10-10T12:24:57.321">
   ...
  </tt:Frame>
  <tt:Frame UtcTime="2008-10-10T12:24:57.621">
   ...
  </tt:Frame>
 </tt:VideoAnalytics>
</tt:MetaDataStream>

<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="...www.onvif.org/...">
 <tt:Event>
  <wsnt:NotficationMessage>
   <wsnt:Message>
    <tt:Message UtcTime= "2008-10-10T12:24:57.628">
    ...
    </tt:Message>
   </wsnt:Message>
  </wsnt:NotficationMessage>
 </tt:Event>
</tt:MetaDataStream>
```

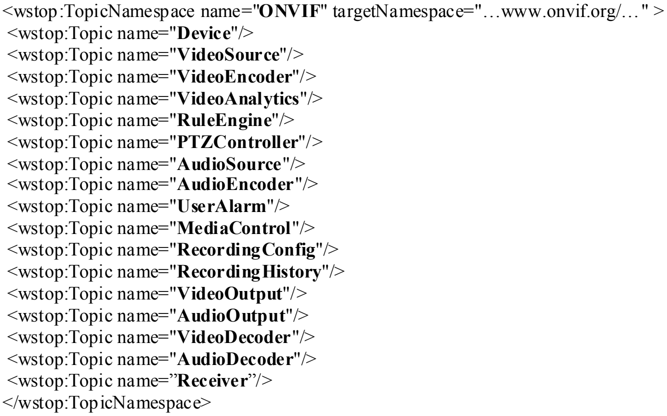
Fig. 2. Scene description and event XML stream structure example (source ONVIF)

In order to receive an event a requester has to subscribe to it on the interface of the device generating that particular event. The interface is provided by the *Event Service* which is mandatory to be supported by all devices. It is based on OASIS WS-BaseNotification and WS-Topics specifications. The device acts as notification producer and topic expressions are being used as filter. Topic expressions may be created as expression tree. By subscribing to a particular topic expression automatically all notifications for leaves expressions further down the tree are being subscribed to. Events may be streamed over RTP or transmitted as message in the web service environment. Here, also a real-time pull point interface is specified providing for a firewall friendly solution.

The notification messaging framework defined in WS-BaseNotification is being exploited. The message element of a NotificationMessage as defined in the ONVIF schema is composed of "Source", "Key" and "Data". A unique identification of the device's component detecting the event should be indicated by "Source". "Data" should contain detailed information about the detected event.

A categorization of events is provided by the usage of topics. A collection of root topics is being provided in the ONVIF namespace. It is easily expandable for event specific topics according to vendor needs.

If the topic refers to a property "Key" is used in order to make the property unique, i.e. distinguishing different qualities of objects in the scenery. Properties can be "initialized", "changed" and "deleted", in this way keeping track of what has been observed over the lifetime of a property.



```
<wstop:TopicNamespace name="ONVIF" targetNamespace="…www.onvif.org/…" >
 <wstop:Topic name="Device"/>
 <wstop:Topic name="VideoSource"/>
 <wstop:Topic name="VideoEncoder"/>
 <wstop:Topic name="VideoAnalytics"/>
 <wstop:Topic name="RuleEngine"/>
 <wstop:Topic name="PTZController"/>
 <wstop:Topic name="AudioSource"/>
 <wstop:Topic name="AudioEncoder"/>
 <wstop:Topic name="UserAlarm"/>
 <wstop:Topic name="MediaControl"/>
 <wstop:Topic name="RecordingConfig"/>
 <wstop:Topic name="RecordingHistory"/>
 <wstop:Topic name="VideoOutput"/>
 <wstop:Topic name="AudioOutput"/>
 <wstop:Topic name="VideoDecoder"/>
 <wstop:Topic name="AudioDecoder"/>
 <wstop:Topic name=”Receiver”/>
</wstop:TopicNamespace>
```

Fig. 3. ONVIF root topics (source ONVIF)

It should be noted, the synchronization of subscribed properties between device and client by requesting a SynchronizationPoint (see paragraph 4) corresponds to tuning in to an existing session. All properties are set to “Initialized” starting their lifetime for that particular client from beginning.

“Source”, “Key” and “Data” each consists of a set of simple name value pairs or structured information. A device can describe the items contained separately for each topic using the Message Content Description Language provided by ONVIF.

Each and every device is required to provide URI locations to schemata being used in the description as well as URI locations to topic expression dialects and message content filter dialects supported.

Support of the Concrete Topic Expressions defined in the [WS-Topics] specification and its ONVIF extension referred to as ConcreteSet is mandatory. The extension allows e.g. for usage of “OR” operations providing for general subscriptions to topics generated by different sources at once.

Also required is support for a subset of XPath 1.0 syntax allowing for more detailed message content filtering for client subscriptions.

Figure 4 provides an overview of all ONVIF defined interfaces and services required to be implemented for a NVA. The NVA device can be managed using the *Device Service* and functionally configured using the *Analytics Device Service*. In addition the *Analytics Service* is used for the configuration of single analytics algorithms (not shown). The *Receiver Service* provides necessary information in order to fetch media data using the data streaming interface. Notification is provided using the *Event Service* and metadata of a scene description may be streamed out.

### 3.2 Compressed domain video analysis

The video data stream arriving at the NVA is compressed in e.g. MPEG-4 AVC|H.264 format. In order to analyze the picture data the stream needs to be decompressed first. If compressed domain analysis is being performed the decompression stage can be avoided, thus reducing required computationally complexity.

We propose a method to perform object detection in MPEG-4 AVC|H.264 bitstream by taking into account motion and residue information. It is performed in two steps: First, a

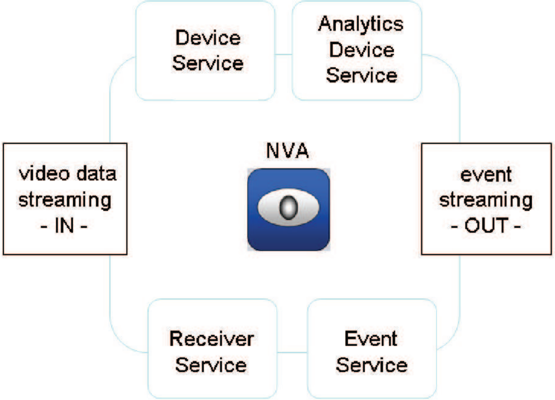


Fig. 4. Network Video Analytics device required services/interfaces

simple segmentation process using temporal filtering is performed over each motion vectors and coefficients to segment the moving object from background. The filtering process is performed by observing the smoothness of a frame, denoted by the number of macroblock having nonzero motion vectors and nonzero dequantized coefficients (a "qualified" macroblock to represent moving objects); second, an automatic clustering is performed to classify groups of block partition into clusters representing the candidate of object. In this process we define searching windows with respect to camera position to group adjacent blocks into the same cluster without determining the number of cluster and initial cluster center. Using appropriate searching window scheme we can ensure that even small objects can be correctly recognize by the algorithm.

The algorithm first parses the MPEG-4 AVC|H.264 bitstream and defines the data into "block partition data" and "residue data". Block partition data is the data that are uniform within a 4x4 block partition, that is, all 16 pixels in that block has the same value. Residue data is the data that is unique for each pixel, in which a 4x4 block partition may have different values in its 16 pixels. Therefore, in one macroblock there are 16 partitions of block partition data and 256 partitions of residue data. The reason to categorize the data into these categories is to simplify and increase the accuracy filtering frame from noise. Fig. 3 shows the block diagram of the algorithm.

### 3.2.1 Data definitions

In MPEG-4 AVC|H.264 luminance component of a macroblock can have various combinations of block partitions due to tree structured motion compensation that enables a macroblock partitioned into 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 partition types (ISO/IEC 14496-10:2009). Therefore, a macroblock may have motion vectors only in several block partitions, leave some other partitions have no motion vectors and can be assumed as partition that is not belong to moving object. By adjusting the partition of macroblock we can further remove 4x4 block partitions that are actually not belong to moving object by observing the distribution of residue data in that macroblock. Thus we may keep our process only for the partitions with motion vectors. For block partition data, if the original macroblock partition is larger than 4x4, we adjust the block partition to have the same motion vectors for each of its 4x4-partition member.

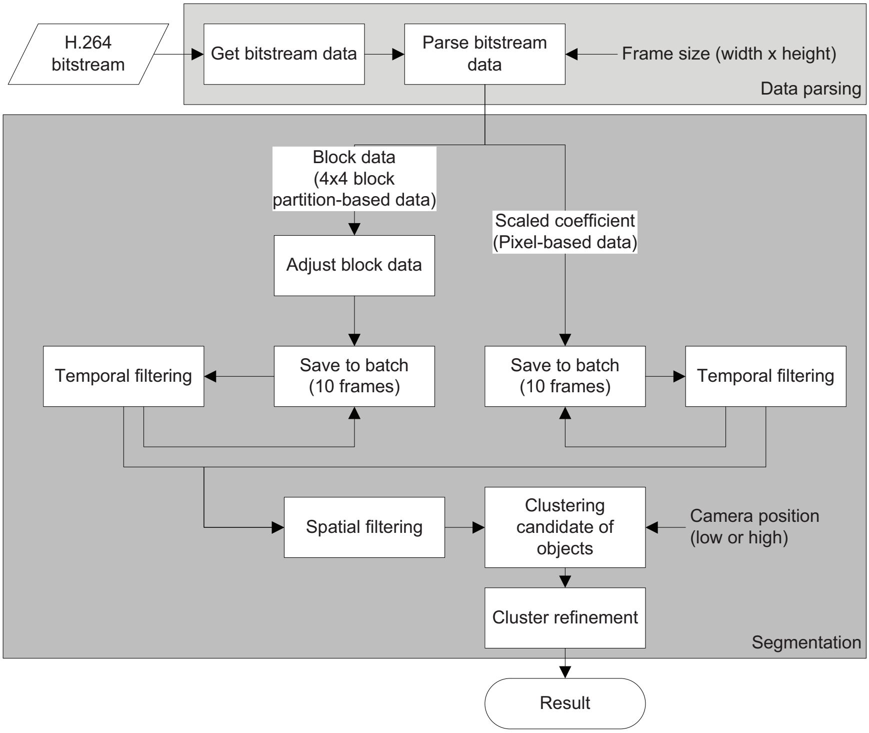

H.264 bitstream
Get bitstream data
Parse bitstream data
Frame size (width x height)
Data parsing
Block data (4x4 block partition-based data)
Scaled coefficient (Pixel-based data)
Adjust block data
Save to batch (10 frames)
Temporal filtering
Save to batch (10 frames)
Temporal filtering
Spatial filtering
Clustering candidate of objects
Camera position (low or high)
Cluster refinement
Segmentation
Result

Fig. 5. Block diagram of object detection algorithm

After adjusting the distribution of values in block partition data, the segmentation process continues by first removing noise that may lead to incorrect object detection using temporal filtering. Here noise is defined as data that appears inconsistently along the series of consecutive frames. In the temporal filtering, series of ten frames as one batch are set. Within the batch, the consistency of a macroblock containing block partition data and residue data appears are observed. For this purpose we must take the subset of block partition data and residue data in a frame from one macroblock.

First, define block partition data $B_{ij}$ as a 4x4 block partition data in location *{i,j}* and $R_{mn}$ is residual data of a pixel in location *{m,n}* in a frame. The index *{i,j}* is in 4x4 block-unit and *{m,n}* is in pixel unit. Therefore, in a CIF frame, for example, there are 25,344 block partitions structured into 88 rows and 72 columns of block partitions and up to 352x288 residue data.

Next, define $M_k$ as an arbitrary *k*-th macroblock in frame *f* composed of non-null block partition $B_{ij}$ and nonzero residue data $R_{mn}$, respectively. Macroblocks with non-null block partition and nonzero residue data is defined as "qualified macroblocks".

The segmentation process continues by removing noise that may lead to incorrect object detection using temporal filtering. Here we define noise as data that appears inconsistently along the series of consecutive frames. In our temporal filtering, we set series of ten frames as one batch. Within the batch, we observe how consistent a macroblock containing block partition data and residue data appears. For this purpose we must take the subset of block partition data and residue data in a frame from one macroblock.

### 3.2.2 Block filtering

Threshold values are set to determine the degree of inconsistency in terms of number of frames in a batch to filter the noise macroblock. These values are adjusted based on how much motion vectors and residual data are produced in the encoded bitstreams. In the experiments the consistency of qualified macroblocks along the sequence are observed by adjusting the number of frames in which the macroblocks that are inconsistent are allowed to be kept.

Camera position is also used to determine the threshold value. In video taken from camera located in high position such as outdoor surveillance where camera is located in a pole at the side of a street, it is expected to have small moving object, therefore an object may be represented by few number of block partitions. On the other hand, if the camera is located in low position such as indoor surveillance where camera is located in the ceiling of a room, we expect to have large moving object which represented by many block partitions.

By adjusting the threshold value for several sequences with value from 2 (i.e. remove a macroblock if it inconsistently appear in less than 2 frames within the batch) to 8, our observations yield the optimum threshold value for block partition data $\rho_B$=5 for high camera position and $\rho_B$ =4 for low camera position. Similarly we determine threshold value for residue data $\rho_R$ =4 for both camera positions.

Finally, let $h$ be the number of frame in a series of consecutive 10 frames, the data is filtered for every macroblock $M_k$ within the 10 frames by keeping the blocks $B_{ij}$ if $h$>$\rho_B$ and keeping residue data $R_{mn}$ if $h$>$\rho_R$, otherwise the blocks and residue data are removed.

Up to this process, we already remove the noise and segment the frame into foreground: the block partition data and residue data that expected to be part of moving object; and background: the data that are not part of moving object. To keep the relation of block partition data and residue data in a single representation, we define both data types into single unit of 4x4 block partition size.

We define $C_{ij}$ as the $i$-th block partition data in frame $f$ called “candidate of cluster” composed of $B_{ij}$ and $R_{ij}$. $R_{ij}$ is composed of 16 residue data of $R_{mn}$ where *{m,n}* is the location of 16 pixel inside block *{i,j}*. $C_{ij}$ is constrained by having non-null block partition data and at least one nonzero residue data exist in a block partition. Therefore any block partition composed by either only $B_{ij}$ or only $R_{ij}$ will be omitted in the next process.

In temporal filtering, we removed any data that are inconsistent during series of frame. However, we may also find consistent data (according to the threshold parameters) which are actually noise. Such data occurred as isolated data, i.e. the data comprises of only few number of block partition (usually less than four blocks), or block partitions that “dangled” from group of blocks. The spatial filtering process aims to remove such noise.

Based on the aforementioned problem, the spatial filtering is quite simple: keep $C_{ij}$ if it has more than four adjacent neighbors, otherwise, it is removed. The adjacent neighbor is defined as 8-connectivity neighbor is the direct adjacent block of $C_{ij}$ from top-left, top, top-right, right, left, bottom-left, bottom and bottom-right order, respectively.

After the noises are removed, the blocks are then grouped into clusters where each cluster represents the blocks of actual detected moving object.

### 3.2.3 Automatic clustering

Clustering aims for collecting groups of block partitions, the candidate of cluster, into single representation of object to be tracked. A cluster is defined as group of block partitions in a frame that adjacent to each other. A group may contain at most four block partitions and consistently appears in several frames of ten consecutive frames based on the threshold of temporal filtering and at least has four neighbors based on spatial filtering process.

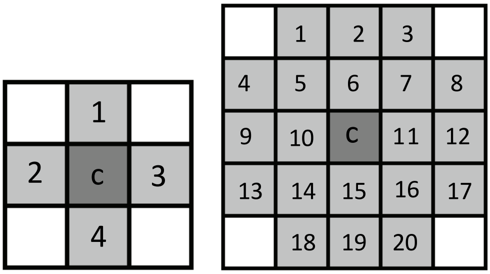
Clustering process can be seen as mapping block partition into clusters in subjective and non-injective case, that is, one block partition shall be mapped to exactly one cluster while one cluster may be mapped from one or more block partition, and there is no null cluster (a cluster without any block partitions mapped to it).

Here we perform automatic clustering of the block partitions because we don't know how many clusters can be generated from group of blocks and we cannot initiate random cluster center. It is a greedy algorithm that scans all block partitions in a frame in progressive order and search for its adjacent neighbor using specified searching window then recursively find for block partition and its adjacent neighbor inside the window.

Define a cluster $O_p$ consists of groups of candidate of cluster blocks where $p$ denotes the index of cluster. A cluster may consist of a finite number of block partitions, where one cluster may contain any arbitrary adjacent $C_{ij}$s. To lower the computational complexity in implementation the maximum number of clusters to be recognized is set to ten objects.

We use searching window to find adjacent blocks of a block partition. The searching window determines how far we must search for adjacent blocks to be grouped into the same cluster as the initial block partition from which the search is started. Two searching windows are defined, as shown in Fig. 6, with respect to the position of the camera used to take the video: 4-connectivity neighborhood window ("4w" window) for high camera and 20-connectivity neighborhood window ("20w" window) for low camera. We omit the adjacent blocks at the corner of the searching window to avoid the blocks that are actually not part of another object be clustered in the same cluster of an object.

In video taken with camera located in high position the searching window is limited to the four adjacent neighboring blocks. On the other hand, if the camera is located in low position, with the possibility that one object may have several group of block partitions, the searching area is extended to the adjacent twenty neighboring blocks. Since the position of camera usually unchanged during the use of the camera, implementation of searching window can be appropriately chosen once for one camera.



(a) (b)

Fig. 6. Searching window (a) 4-connectivity neighborhood window and (b) 20-connectivity neighborhood window; *c* is the block partition to be searched for neighbor

Due to incorrect motion estimation or very small differences of two frames, sometimes the moving object cannot be detected by looking at the availability of blocks with non-zero motion vectors or residual data. In the end, the clustering method will be failed to cluster any object. In this case, a temporal refinement is performed to restore missing clusters.

In the temporal refinement, we project the clusters that inconsistently disappear in series of frames. A cluster may be projected when there is missing clusters in a frame, by taking the

average value of static parameters of clusters in a series of five frames so the interpolated cluster will have average value of position, direction and energy. From filtering and projection process we expect to have consistent clusters (the candidate of objects).
The examples of segmented clusters are shown in Fig. 7, where clusters are distinguished with different colors for visualization.

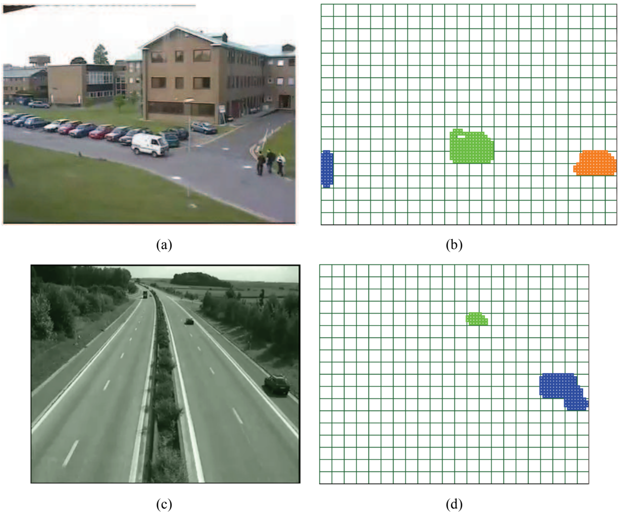

(a) (b) (c) (d)

Fig. 7. Block partitions already labeled into different clusters (shown with different colors) in (a),(c) original frame and (b),(d) clusters generated from the 428th frame of *PETS2001* sequence and the 51st frame of *Speedway* sequence, respectively.

The result of video analysis discussed in this section can be annotated in a structured description using XML metadata instantiation. The annotation may describe pertinent information such as the time and frame number when the objects are detected, the actual time information of the scene, the description of the location of detected object, the identity of the detected object and the information about the networked camera used to detect the object. These descriptions will be stored in a structured file format together with the actual video data through the network video storage device.

## 4. Storage device and data streaming

Media data provided by NVT's and associated metadata generated by NVA's have to be recorded in order to be used for later re-viewing and further forensic analysis. In ONVIF appropriate interfaces for a Network Video Storage (NVS) device have been defined providing for recording of streamed media and metadata as well as for structured access by clients.
The logical structure chosen is comprised of a container denoted as recording holding any number of tracks. Tracks can be of the type video, audio and metadata containing appropriate data at certain times. The minimum configuration, three tracks – one of each video, audio and metadata, might be extended by creating additional tracks for e.g. audio backchannel recording. The creation of new recording container and new tracks respectively is supported only if signalled as particular capability of the NVS.

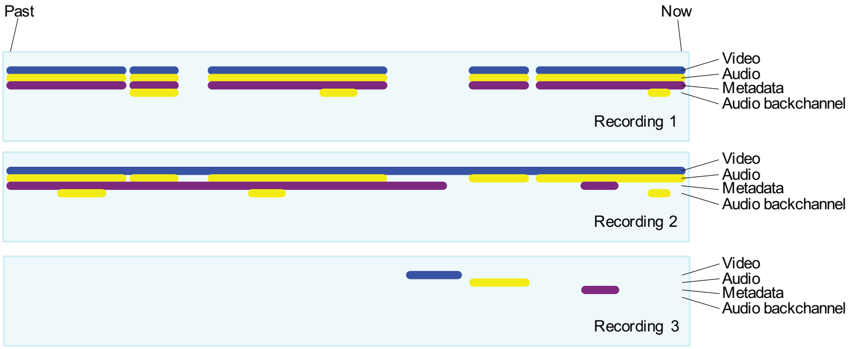


Fig. 8. Example of recordings and tracks structure (source ONVIF)

A parameter has been defined determining the maximum time data of a recording shall be stored. Retention time can be set to infinite but the device may automatically free up any required storage space for new recordings. A mechanism to lock ranges of data has not been defined by ONVIF.

To save data to a recording a RecordingJob as control instance is needed pulling data from one or more sources into the tracks of the recording. Required structures can be pre-configured providing means to implement e.g. alarm recording. Here, changing the mode of RecordingJob between idle and active is the only action required to start and stop recording.

The NVS relies on the *Receiver Service* in order to receive media data from other devices.

Receiving streamed data in ONVIF is unified and consolidated in the *Receiver (Configuration) Service*. Every device able to receive media streams implements this service. It is the client's responsibility configuring the receiver object to contain the operative stream URI, information how to setup the stream and the mode of the receiver. Modes are defined for the receiver attempting to maintain a persistent connection as well as to connect on demand. There is no actual media data transfer necessary because the receiver serves as RTSP client endpoint. Appropriate keep-alive methods may be used to obtain persistence. Tagging mechanisms are being used by the services calling the receiver to distinguish between tracks of the same type within a RTSP stream.

Receivers are to be used non-exclusively reducing the number of parallel connections a device may be required to maintain.

A particular point of failure has been identified in dangling RecordingJobs and receivers not properly connected to each other or not deleted completely. Therefore, a mechanism has been defined to automatically create a new receiver and attach it to a recording job. However this mechanism only works for receivers used with a single recording job. If the RecordingJob does not use the receiver any more it should be deleted automatically.

Data streaming

Transportation of media and metadata between endpoints in the ONVIF eco-system is being performed as streaming over IP-networks. Here, usage of the Real-time Transport Protocol (RTP) is required. It defines a packet based delivery of data providing support for detecting unordered arrival of packets. It has to be used together with the User Datagram Protocol (UDP).

Additionally, support is required for media transfer using RTP/RTSP/HTTP/TCP in order to traverse firewalls. Here, conformance to QuickTime available form Apple Inc. has to be assured for the tunneling while also the Embedded [Interleaved] Binary Data specification for RTSP shall be obeyed. The latter requires for Base64 encoding of RTCP feedback as well as backchannel packets.

For the initiation of sessions as well as for playback control RTSP over TCP has to be used and the Session Description Protocol (SDP) for information provisioning about media types, formats and associated properties.

In order to keep RTSP sessions alive the client side is expected to send receiver reports or to call the RTSP server using any method. All devices are expected to support RTCP sender reports for media synchronization.

Error classification for RTP and HTTP respectively follows the standard status code definitions.

For metadata streams particular definitions have been made concerning several RTP header elements. For example, if the RTP marker bit is set to "1" it signals completion of the XML document transmitted.

Raising the level of efficiency a particular syntax has been chosen for the transmission of JPEG data over RTP. ONVIF uses RFC 2435 in which an RTP header extension is defined omitting JPEG frame header, JPEG scan header as well as quantization and Huffman coding table transmission in the payload bitstream.

Particular attention has to be paid to the backchannel connection handling. Here, functionality extensions to RTSP have been defined which can not be understood by regular server implementations. A Require tag is introduced to the RTSP header and clients use it in the DESCRIBE message to signal a bidirectional connection request. The enabled server includes an attribute in the SDP media description section indicating the direction media data will be send.

Not all in ONVIF required data streaming protocols guaranty for data delivery, e.g. packet loss may occur. In order to define a payload coding format independent mechanism to indicate a request for synchronization the Picture Loss Indication (PLI) message has been chosen. It is a RTSP feedback message providing for statistically more immediate feedback to the sender. If receiver implementations have access to the web service interface of the transmitter the synchronization point mechanism as defined in ONVIF may be applied instead. Through this mechanism the NVT is enforced to insert an INTRA coded picture in the video stream as soon as possible. Also, event properties or PTZ status information may be synchronized by this means.

The *Event Service* is being used to provide notification about creation, configuration changes and deletion of tracks and recordings. For those events particular Topics and message payload have been defined.

In general, it is left to the vendor of a NVS to provide information about data recorded, structures, timelines of recordings, etc. Assistance on how to convey this information in an ONVIF conformant way has been introduced with the concept of historical events represented also as notification messages. Here, events are being generated by the device itself and recorded. The *Recording Search Service* has been designed to have access to that type of events. Two events for the RecordingHistory root topic are required to be provided. That is, the state of a recording (signaling start/stop of a recording) and presence of data for a track (signaling substantial content).

In order to retrieve recordings and events associated with recordings a NVS hast to provide a search interface. It is session based (identified by a token) and realized as coupled find and fetch results operations. Results may be fetched all at once or in increments as defined by the requester. The search can be performed backwards and forward in time from a chosen starting point.

The search space can be limited with a set of parameters e.g. recording tokens and filters to be provided by the requester. Here, a restricted XPath dialect is being used to navigate around a tree representation of the XML data and selecting nodes based on search criteria. Also other metadata and PTZ positions may be addressed by the search interface.

In addition to providing access to historical events generated by the device itself or inserted by a client a NVS may be required to generate virtual events in order to communicate the original state of property events. If a requester indicates desire to be informed about the status of properties at the start point of the search such virtual events needs to be created on the fly.

## 5. Display device and data export

A Network Video Display (NVD) device provides processing for the decoding of media streams and configuration options on where and how to output it for human observer.

A NVD provides outputs and potentially also sources for media data in case the backchannel mechanism is supported. Configuration of these entities requires *DeviceIO Service* to be supported.

After configuring the physical in- and outputs of the NVD a client can define a layout for each output. A layout defines the arrangement of display areas on a monitor, e.g. single view or split screen. During the session set-up a NVD signals if a certain set of predefined layouts is available.

Areas on a physical display being addressable and configurable are called pane. A layout assigns the area of the display and the pane configuration. The order of the panes in the layout also determines the order of the panes on the monitor if overlapping panes (windows) are supported by the device. The first pane on the list is displayed in the foreground at start.

Audio and video data for a particular NVT media profile are combined in one pane configuration. If a NVT has a microphone attached and supports audio encoding, audio data may be transmitted and listened to at the NVD. If a NVT has loudspeakers attached and supports audio decoding, audio data may be transmitted from the NVD and output at the NVT. The latter is enabled by the backchannel mechanism defined in ONVIF.

The NVD is required to decode JPEG data and if it supports audio also G.711µLaw – exactly the same codecs that are mandatory for the NVT.

The NVD also has to support the *Event Service*. Particular events have been defined to signal if unsupported data formats are being received or packet loss occurred.

The NVD also makes use of the *Receiver Service* in order to pull the data. Data may be received from NVS or NVT directly. The pane configuration holds a reference to the appropriate receiver object containing the URI from where to fetch data.

If data are being received from a NVS the *Replay Service* will provide it. Every NVS has to implement the *Replay Service* defined by ONVIF. It defines extension to the RTSP protocol in order to support e.g. timing and track referencing.

A RTP header extension is being defined containing monotonically increasing NTP timestamps, indicating absolute UTC time associated with the access unit. An additional Rate-control header field is introduced distinguishing between server and client side based playback speed control. If not present the server will be in control, providing support for modest client implementations.

Several features are supported in order to make replay convenient and provide for surveillance typical requirements. One of it is reverse replay. In that case transmitted data are segmented into chunks always starting with an INTRA coded picture. These chunks are sent in reverse order while the packets inside the chunk are being sent in regular order. For single stepping the client is expected to cache a chunks data. Depending on the Rate-control header field setting the RTP timestamps take on different values (decreasing or increasing within a chunk) which has to be taken into account presenting the pictures to an observer as well as for saving it to file.

The *Replay Service* is in ONVIF also used to provide for data export. The client can download the required data in order to process and safe it to an appropriate exchange format.

### 5.1 Video surveillance exchange format

Data access and distribution within the ONVIF system has been described above. For the cooperation with governmental authorities as well as between different forces of legislation and execution data might to be taken out of the ONVIF system. A standardized data exchange format is necessary at this point enabling data exchange without any loss or damage and to reduce costs for further analysis and viewing equipment. In the following such an exchange format is described purposefully designed for the application in the surveillance domain.

The MPEG-A Multimedia Application Format (MAF) is MPEG standard that has the aim of providing the standardized package format of audiovisual contents, content description metadata and intellectual property management and protection (IPMP) metadata and rights expression language (REL) metadata etc. for specific application domains. The MAF defines not only the file structure based on ISO base Media File Format, but also the components of resources and metadata to be included. The MAF may consist of MPEG and non-MPEG standard technologies, upon requirements for specific industry applications. The VSAF standard is Part 10 of MPEG-A (ISO/IEC 23000-10:2009) which specifies the storage format for video surveillance recording and its corresponding metadata which adopts MPEG-4 AVC|H.264 as the video resource and a set of MPEG-7 Multimedia Description Scheme (MDS) tools (ISO/IEC 15938-5:2003) and Visual descriptors (ISO/IEC 15938-3:2002). Its purpose is to enable interoperability for various video surveillance systems as well as to provide the description about the file structure and visual contents.

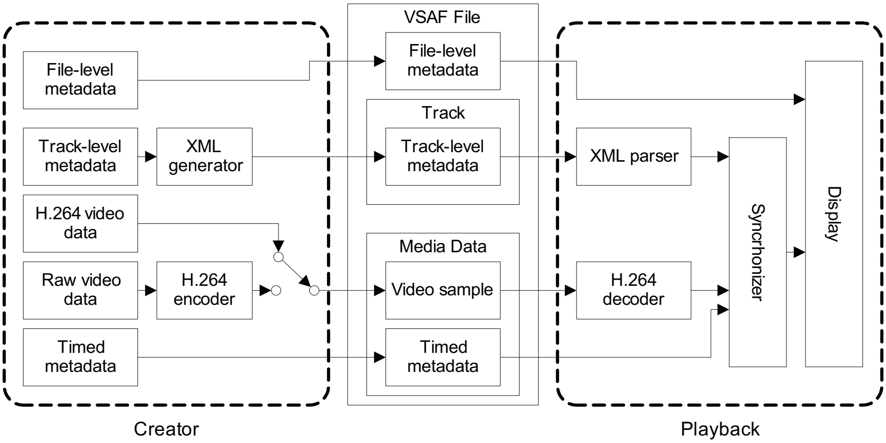


Fig. 9. Creating and using VSAF file format

Fig. 9 shows the conceptual illustration of creating and using VSAF file format for video surveillance data and metadata in a single file. Generally there are two types of data used to create VSAF file: video data and metadata. VSAF specification requires at least one track of video data type encoded using MPEG-4 AVC|H.264 Baseline profile (up to level 3.1). If the source camera does not provide such bitstream then the video shall be first encoded conform to the specification. In addition, different video track, contain the same or different content with the main AVC video track, encoded with other coding technology is also permitted. The metadata stored in VSAF file comprises of binary metadata and textual metadata. Binary metadata is stored in binary format according to the VSAF file structure while textual metadata is stored as XML instantiation of MPEG-7, thus an XML generator shall be needed to create the instantiation.

The structure of VSAF file format in brief is as follow. The file is composed of structure called box, an object-oriented structure in which the data is stored according to specified syntax and semantics. The actual video data is stored in a box called media data box in which the data is usually partitioned into samples called chunks. To enable playback of these chunks, synchronization information such as time stamp, data offset (location) and data size are stored in binary timed metadata box within the media data box. The bitstreams specific information such as profiles and levels for codec types, bitrates, frame rates and screen size are also needed for decoding the bitstreams. This information is stored in boxes within another box called track box where one box corresponds to one video data. In VSAF file format, the track box also contains a box that stores textual metadata and, in addition to ISO base file format, a box for binary metadata is also specified in root (file-level) box. VSAF file structure also enables movie fragments to support the playback of the surveillance video while the data is still being recording. With this structure, features such as instant replay or live metadata generation can be enabled.

It is not uncommon for surveillance video to have long time period of recording while on the other hand the storage or transmission device may have provided limited space or bandwidth. To overcome this problem, the VSAF specification enables the VSAF to be stored in so-called VSAF fragments where each fragment covers a limited amount of time and can be connected to other VSAF fragment. The connection between VSAF fragments is determined by linking their universal unique identifier (UUID). Each VSAF fragment is linked to a predecessor and successor fragment via UUID. A current VSAF fragment is set when the fragment does not have any predecessor or successor fragments. By doing so, a long-time surveillance recording can be fragmented into several linked VSAF fragments (i.e. VSAF files).

Playing VSAF file is performed by first parsing the file format for the metadata and video data. In order to playback the video, it shall be first decoded from MPEG-4 AVC | H.264 bitstream and synchronized with timed metadata. The XML metadata shall be first parsed using XML parser and synchronized with the video to show the appropriate scene description. Obtaining information such as querying specific object using a color as cue or finding the record of object’s trajectory can be easily performed because the XML instantiation provides specific location (timestamp information) of the queried object in the video data.

VSAF specification enables the use of metadata to describe information about the content and the file video itself. Two formats of metadata are used to describe the video: binary and textual metadata. Binary metadata are used to store information about the time stamp of video sample (timed metadata) and to store the information of identification and creation time of VSAF file (file-level metadata). This metadata is stored in VSAF as binary data according to VSAF file format specification. The timed metadata is used to describe the timestamp information of the video stored in the file in which every video sample has it own time information. The file-level metadata is used to describe the information regarding the identification and creation time of a VSAF file and the textual annotation about the contents within the VSAF. It also allows for the use of classification scheme to describe the content. The metadata should be located in the top level of the VSAF file in order to enable easy access for identification information of the VSAF file. In the file structure hierarchy of VSAF, this metadata is located in the file level, hence being called the file level metadata.

On the other hand, the textual metadata is used to describe the information regarding the content of the VSAF. It is specified by selecting some parts of MPEG-7 MDS tools and Visual descriptors for the VSAF. It describes the track identification information, camera equipment, timing information for each track, text annotation to describe the event, decomposition of frames, locations of the objects as ROI's in the frame, color appearance of the objects, and identification of the object. The specification of VSAF has a limited set of MPEG-7 Visual descriptors such as the dominant color and scalable color descriptors. Since the track-level metadata describes the video content, the metadata with the MPEG-7 dominant color and scalable color descriptor values are located inside the respective track boxes of VSAF, hence being called the track level metadata.

Metadata structures defined in MPEG-7 can be utilized to capture results of the object detection described in Section 3. More specifically, time information, location information and object's identity can be described using MPEG-7 MDS while visual information such as the location of the detected object in the frame and its trajectory can be described in MPEG-7 Visual. VSAF also enables embedding other metadata formats such as from ONVIF without any change to the VSAF file structure by merely adding an additional metadata element in the textual metadata.

## 6. Conclusion

In this chapter we provided an overview of modern surveillance system composition exploiting available standardized technology supporting recent trends in the surveillance domain: networked cameras and web services, host-based automated data analysis as well as interoperable media formats for data streaming, storage and export.

## 7. References

OASIS Organization for the Advancement of Structured Information Standards; www.oasis-open.org

ONVIF Open Network Video Interface Forum; Core Specification; Version 2.0; www.onvif.org

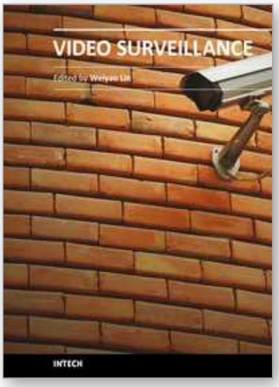
ISO/IEC 23000-10:2009, Information technology – Multimedia application format (MPEG-A) – Part 10: Video surveillance application format, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50554

ISO/IEC 15938-3:2002, Information technology -- Multimedia content description interface – Part 3: Visual http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=34230

ISO/IEC 15938-5:2003, Information technology -- Multimedia content description interface -- Part 5: Multimedia description schemes
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=34232

ISO/IEC 14496-10:2009, Information technology -- Coding of audio-visual objects -- Part 10: Advanced Video Coding
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=52974

ITU-T Recommendation H.264 (03/10): Advanced video coding for generic audiovisual services
http://www.itu.int/rec/T-REC-H.264-201003-I/en



**Video Surveillance**

Edited by Prof. Weiyao Lin

ISBN 978-953-307-436-8
Hard cover, 486 pages
**Publisher** InTech
**Published online** 03, February, 2011
**Published in print edition** February, 2011

This book presents the latest achievements and developments in the field of video surveillance. The chapters selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. Besides the introduction of new achievements in video surveillance, this book also presents some good overviews of the state-of-the-art technologies as well as some interesting advanced topics related to video surveillance. Summing up the wide range of issues presented in the book, it can be addressed to a quite broad audience, including both academic researchers and practitioners in halls of industries interested in scheduling theory and its applications. I believe this book can provide a clear picture of the current research status in the area of video surveillance and can also encourage the development of new achievements in this field.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Houari Sabirin and Gero Bäse (2011). Video Surveillance of Today: Compressed Domain Object Detection, ONVIF Web Services Based System Component Communication and Standardized Data Storage and Export using VSAF – a Walkthrough, Video Surveillance, Prof. Weiyao Lin (Ed.), ISBN: 978-953-307-436-8, InTech, Available from: http://www.intechopen.com/books/video-surveillance/video-surveillance-of-today-compressed-domain-object-detection-onvif-web-services-based-system-compo

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821