

# Recent Advances in MPEG-7 Cameras

Frederic Dufaux and Touradj Ebrahimi

Institut de Traitement des Signaux  
Ecole Polytechnique Fédérale de Lausanne (EPFL)  
CH-1015 Lausanne, Switzerland  
Frederic.Dufaux@epfl.ch, Touradj.Ebrahimi@epfl.ch

Emitall Surveillance S.A.  
CH-1820 Montreux, Switzerland  
Frederic.Dufaux@emitall.com, Touradj.Ebrahimi@emitall.com

## ABSTRACT

We propose a smart camera which performs video analysis and generates an MPEG-7 compliant stream. By producing a content-based metadata description of the scene, the MPEG-7 camera extends the capabilities of conventional cameras. The metadata is then directly interpretable by a machine. This is especially helpful in a number of applications such as video surveillance, augmented reality and quality control. As a use case, we describe an algorithm to identify moving objects and produce the corresponding MPEG-7 description. The algorithm runs in real-time on a Matrox Iris P300C camera.

**Keywords:** smart camera, MPEG-7, video analysis, content description

## 1. INTRODUCTION

Recent advancements in acquisition, data processing and storage technologies have enabled the emergence of a new class of cameras, referred to as smart cameras, which perform video analysis to generate a content-based metadata description of the scene. These smart cameras are especially suitable for applications such as video surveillance, augmented reality or industrial manufacturing. In these applications, the camera output is meant to be received by a machine rather than by a human being. In this case, it is advantageous to extract the information relevant for the target application directly in the camera. It is then sufficient to only transmit this information, which is directly interpretable by a machine, unlike a video signal.

An MPEG-7 smart camera has first been introduced in [1]. The MPEG-7 camera is a particular case of a smart camera which produces a stream compliant with the MPEG-7 Multimedia Content Description Standard [2][3]. The Eptacam MPEG-7 network camera [4] is based on the same concept.

In this paper, we propose an MPEG-7 camera using a Matrox Iris P300C camera [5]. It features an Intel 400 MHz Celeron processor, enabling embedded image processing. As a use case, we consider a system which identifies moving objects, extracts their shape, and generates the corresponding MPEG-7 description. The algorithm runs in real-time on the camera.

This paper is structured as follow. We first introduce the concept of smart camera in Sec. 2. An overview of MPEG-7 and the MPEG-7 camera is given in Sec. 3. An implementation example is described in Sec. 4, where as experimental results are proposed in Sec. 5. Finally, we draw some concluding remarks in Sec. 6.

## 2. SMART CAMERA

In this section, we define what we mean by a smart camera and how it differs from a conventional one.

### 2.1. Conventional camera

A camera is a device which captures audio-visual information from a scene and converts it to an electric signal. Essentially, a CCD or CMOS sensor collects incoming photons from the scene, resulting in a flow of electrons. In order to view the captured scene, the electrical signal is rendered on a display, with the expectation that the rendering is faithfully representing the original scene. In this conventional paradigm, it is essentially assumed that the output of the camera (i.e. images or video sequences) is consumed by human beings, and the goal is to achieve high perceptual quality and faithful rendering.

Whereas in the past cameras were analog, thanks to the rapid progress in digital technology the trend is now toward digital cameras. Because of the large volume of data involved, compression is commonly performed directly in the camera. As interoperability is paramount in most consumer and professional applications, most camera manufactures rely on standard compression techniques resulting in compatible output formats. For instance, JPEG is commonly used in still image cameras, whereas Motion JPEG, MPEG-2 and MPEG-4 are often used in video cameras. Figure 1 illustrates the concept of conventional camera.

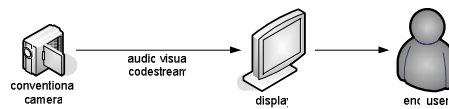


Figure 1 – Conventional camera.

In order to improve the efficiency of the above process, research efforts are still on-going on topics including sensor technology, image processing, compression, transmission, storage or display.

### 2.2. Smart camera

We now discuss a new class of cameras, which we will refer to as smart camera. Nowadays, in many applications, the camera output is not directly consumed by a human being but rather by a machine either alone or in conjunction with a human being.

For instance, in industrial manufacturing a smart camera can be used for automatic quality control, verifying that the pieces produced are within the bounds of the requirements. In a tunnel monitoring system, a smart camera can count the number of vehicles and automatically detect the occurrence of an accident or fire. In a video surveillance system, a smart camera can identify suspicious behaviors or objects, e.g. someone leaving unattended luggage, and automatically trigger an alarm or draw the attention of the system operator.

In the above application examples, the best imaging system is not the one which produces higher perceptual quality or a more faithful rendering of the scene, but rather which helps in improving the detection rate for the specific task. More specifically, the camera has to perform some image/video analysis processing resulting in scene understanding. The smart camera output is then some metadata describing the result of this processing, possibly along with the conventional audio-visual stream faithfully rendering the scene. In this paper we refer to such a camera as a smart camera and the metadata as a content-based description of the scene. The concept of smart camera is depicted in Figure 2.

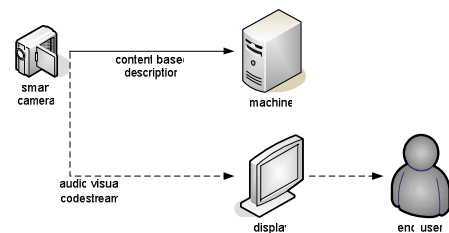


Figure 2 – Smart camera.

### 3. MPEG-7 CAMERA

In this section, we consider a special case of smart camera which produces an MPEG-7 compliant stream as output. We first review the MPEG-7 standard and then describe the MPEG-7 camera.

#### 3.1. MPEG-7

The MPEG-7 Multimedia Content Description Standard has been defined by the Moving Picture Experts Group (MPEG) to standardize content-based description for multimedia data [2][3]. It targets applications such as multimedia information search, filtering, management and processing.

With the goal to define the minimum for interoperability, the scope of MPEG-7 is restricted to the standardization of the content-based description, as illustrated in Figure 3. In particular, the generation of the description (e.g. feature extraction, indexing, annotation tools, etc...) and the consumption of the description (e.g. search engine, filtering tool, retrieval process, browsing, etc...) are not in the scope of MPEG-7 but are left to research and competition.

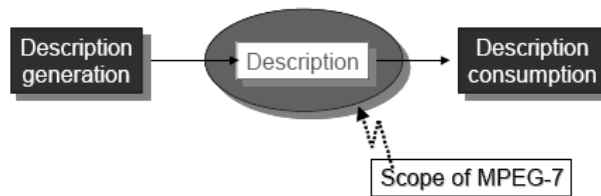


Figure 3 – Scope of MPEG-7 (from [3]).

More specifically, MPEG-7 specifies Descriptors (D), Description Schemes (DS) and a Description Definition Language (DDL). The relationship between these elements is shown in Figure 4.

- Descriptors (D) are used to represent features, and more precisely to define the syntax and semantics of each feature representation.
- Description Schemes (DS) defines the structure and semantics of the relationships between Ds and DSs.
- Description Definition Language (DDL) is the language used to define DSs. In particular, it allows the creation of new DSs and Ds, and the extension of existing DSs. DDL is based on the widely established Extensible Markup Language (XML).

MPEG-7 also defines system tools to support multiplexing of descriptions, synchronization of descriptions with content, transmission mechanisms and file format.

#### 3.2. MPEG-7 camera

An MPEG-7 camera is a special type of smart camera which outputs an MPEG-7 compliant content-based description of the scene [1]. A block diagram of the MPEG-7 camera is given in Figure 5. The camera applies image or video analysis algorithms, such as change detection, face detection, feature point extraction, object segmentation and tracking. The resulting extracted features are then used to instantiate a number of MPEG-7 DSs. Obviously, the choice of analysis algorithms to be used, features to be extracted and MPEG-7 DSs to be instantiated depends on the target application and its requirements.

The MPEG-7 XML description is then encoded either in textual or binary form. The resulting code-stream can subsequently be stored, transmitted and consumed by MPEG-7 devices. If needed, the camera can also output a conventional audio-visual code-stream.

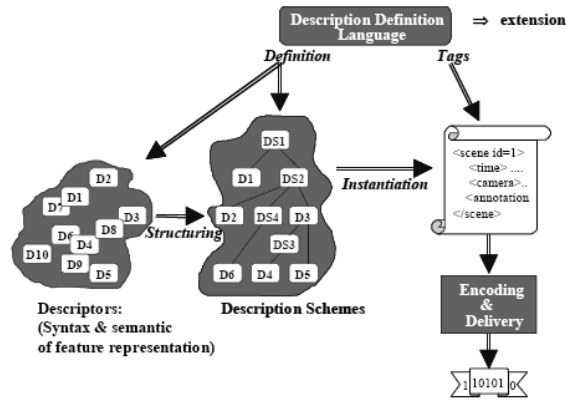


Figure 4 – MPEG-7 D, DS and DDL (from [3]).

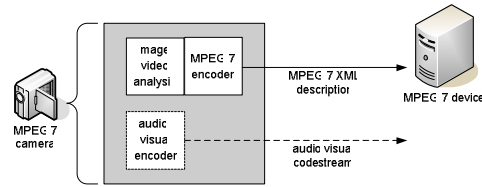


Figure 5 – MPEG-7 camera.

#### 4. IMPLEMENTATION EXAMPLE

In this section, we describe a concrete implementation example of the proposed MPEG-7 camera. As a use case, we consider a system which identifies moving objects, extracts their shape, and generate the corresponding MPEG-7 description. Figure 6 shows a block diagram of the processing taking place in the proposed MPEG-7 camera. A thorough discussion of each step is given hereafter.

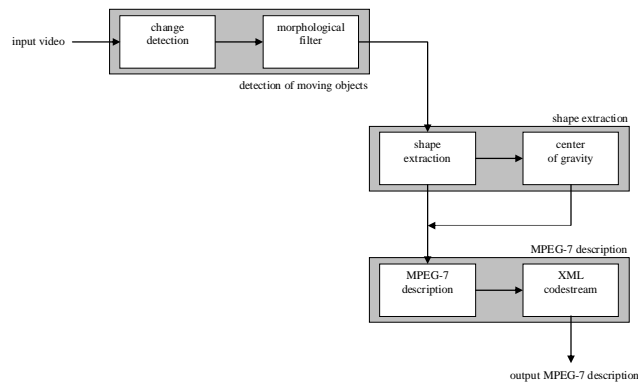


Figure 6 – Block diagram of proposed system.

##### 4.1. Detection of moving objects

The first step is to analyze the input video in order to identify moving objects. We assume that the camera remains static. For the sake of low complexity, a simple frame difference algorithm is applied. More precisely, the background is captured and stored. Regions corresponding to changes are merely obtained by taking the pixel by pixel difference between the current video frame and the stored background, and by applying a threshold.

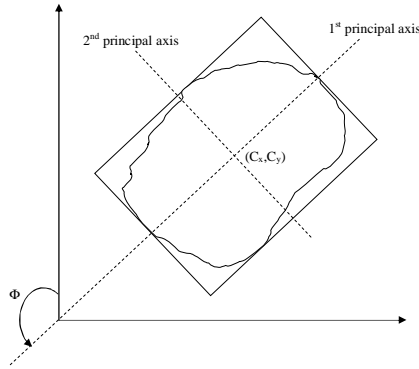
More formally, for each pixel  $(x, y)$  a difference  $D_n(x, y) = F_n(x, y) - B(x, y)$  is calculated, where  $F_n(x, y)$  is the  $n$ -th frame and  $B(x, y)$  is the stored background. A change mask  $M(x, y)$  is generated based on a threshold  $T$ , according to the following decision rule:

$$M(x, y) = \begin{cases} 1 & \text{if } |D_n(x, y)| > T \\ 0 & \text{otherwise} \end{cases}$$

In order to smooth and clean up the resulting segmentation mask, a morphological filter is applied. More specifically, an opening, i.e. erosion followed by dilation, is applied to eliminate small regions. This is followed by a closing, i.e. dilation followed by erosion, which removes small gaps between adjacent regions and holes in objects.

#### 4.2. Shape extraction

The change detection results in a binary mask specifying changed pixels. The next step is to represent the shape of each resulting object. For this purpose, and for the sake of simplicity, a bounding box is used. A bounding box is the tightest rectangular box that fully encompasses an object such that the faces/sides of the box are parallel to the principal axes of that object, as illustrated in Figure 7.



**Figure 7 – Object shape representation using bounding box.**

The center of gravity  $(C_x, C_y)$  is given by

$$C_x = \frac{\sum_{x,y} xM(x, y)}{\sum_{x,y} M(x, y)} \quad \text{and} \quad C_y = \frac{\sum_{x,y} yM(x, y)}{\sum_{x,y} M(x, y)}$$

Considering the moment of inertia tensor

$$I = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{pmatrix},$$

with

$$I_{xx} = \sum_{x,y} y^2 M(x, y), \quad I_{yy} = \sum_{x,y} x^2 M(x, y) \quad \text{and} \quad I_{xy} = I_{yx} = -\sum_{x,y} xy M(x, y),$$

the angle  $\Phi$  is given by

$$\Phi = \frac{1}{2} \arctan \left( \frac{2I_{xy}}{I_{xx} - I_{yy}} \right).$$

### 4.3. MPEG-7 description

The resulting information describing the moving objects, namely the bounding box of each object, has then to be expressed using MPEG-7 description. The position and size of the bounding box is represented using the BoundingBox descriptor.

#### 4.3.1. BoundingBox descriptor

The syntax of the BoundingBox descriptor is given in Figure 8, and the semantics in Table 1.

```

<DType name="BoundingBox">
  <attribute name="LengthUnits" datatype="8bitinteger" required='true' />
  <attribute name="BoxHeight" datatype="real" required='true' />
  <attribute name="BoxWidth" datatype="real" required='true' />
  <attribute name="BoxDepth" datatype="real" required='true' />
  <attribute name="FractionOccupancy" type="real" required='true' />
  <DType name="CompositionInfo" minOccurs='0' maxOccurs='1' />
    <attribute name="BoxCenterH" datatype="real" required='true' />
    <attribute name="BoxCenterV" datatype="real" required='true' />
    <attribute name="Phi" datatype="real" required='true' />
    <DType name="3DCompositionInfo" minOccurs='0' maxOccurs='1' />
      <attribute name="BoxCenterD" datatype="real" required='true' />
      <attribut name="Psi" datatype="real" required='true' />
    </DType>
  </DType>
</DType>

```

**Figure 8 – BoundingBox descriptor syntax.**

Syntax	Semantics
<b>LengthUnits</b>	Units used to represent the bounding box. It typically uses normalized coordinates.
<b>BoxHeight, BoxWidth, and BoxDepth</b>	Dimensions of the tightest rectangular bounding box that encloses the object. BoxHeight, BoxWidth, and BoxDepth are measured along the principal axes in order of decreasing eigenvalues.
<b>FractionalOccupancy</b>	Ratio of the area (volume) of the object to the area (volume) of its bounding box. It takes value in the range [0,1].
<b>Is3D</b>	Boolean flag which takes the value “true” for 3-D object, and “false” for 2-D object.
<b>HasCompositionInfo</b>	Boolean flag which takes the value “true” if certain composition information about the object are provided within this descriptor, and “false” otherwise.
<b>BoxCenter<sub>h</sub>, BoxCenter<sub>v</sub></b>	Vertical and horizontal coordinates of the center of the bounding box, with respect to the image or 3-D world coordinate axes, measured in normalized coordinates for 2-D, or in meters for 3-D.
<b>Φ</b>	Angle (in radians) made by the principal axis, measured in the anti-clockwise direction, with the positive vertical axis (cf. Figure 7). It takes values in the range [0, π].
<b>BoxCenter<sub>d</sub></b>	Depth co-ordinate of the center of the bounding box.
<b>Ψ</b>	Angle between the principal axis and the horizontal axis. It takes values in the range [0, π].

**Table 1 – BoundingBox descriptor syntax and semantics.**

#### 4.4. Matrox Iris P300C camera

As a platform for the smart MPEG-7, a Matrox Iris P300C camera is used [5]. It features a VGA format color CCD sensor, an Intel 400 MHz ULP Celeron processor, 128 MB SDRAM, 64 MB flash and a 10/100Mbit Ethernet Controller. It runs Microsoft Windows CE .NET real-time operating system and can be programmed from a host PC in a Microsoft Windows development environment. The camera is shown in Figure 9, and a schematic of its architecture is given in Figure 10.



Figure 9 – Matrox Iris P-Series camera (from [5]).

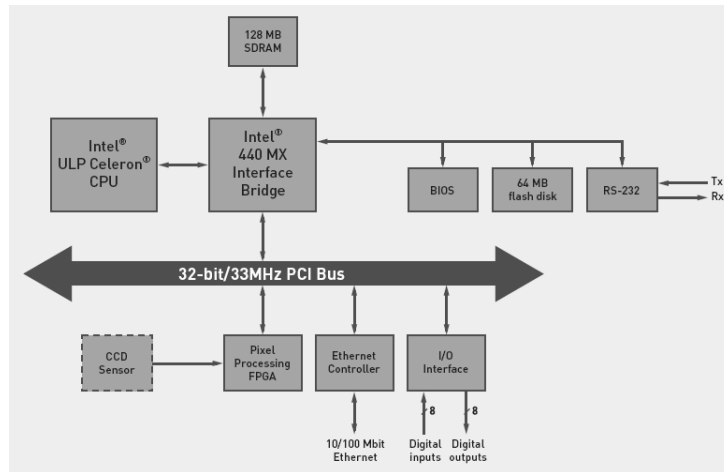


Figure 10 – Matrox Iris P-Series architecture (from [5]).

It is noteworthy to point out that thanks to its low complexity, the proposed algorithm, as described in Figure 6, runs in real-time on the Matrox Iris P300C camera.

### 5. SIMULATION RESULTS

In this section, some experimental results obtained using the algorithm described in Sec. 4 are presented. Figure 11 shows two examples for the well-known sequences “Road” and “Hall Monitor” respectively.

With such an MPEG-7 representation, only the information relevant to the target application is transmitted. Hence, it requires a very low bandwidth when compared to a video signal. For instance, such a system could be used to hide personal data in a video surveillance system, hence preventing the identification of people under surveillance. Therefore, the proposed smart MPEG-7 camera could reconcile the need for video surveillance systems while alleviating the loss of privacy issue.



**Figure 11 – Examples – top: Road, bottom: Hall Monitor;  
Left: original frame, middle: segmentation mask, right: bounding box.**

## 6. CONCLUSIONS

In this paper, we have presented an MPEG-7 smart camera. The camera features an embedded processor and performs image analysis tasks. The output of the camera is an MPEG-7 compliant data stream. Only the information relevant to the target application is transmitted, leading to a very efficient bandwidth usage. The MPEG-7 stream can directly be exploited in applications such as quality control in industrial manufacturing or video monitoring and surveillance system. We presented a specific implementation running in real-time on a Matrox Iris P300C camera.

## ACKNOWLEDGEMENT

EPFL's contribution to this work was partially supported by the European Commission under the IST research network of excellence K-SPACE of the 6th Framework program (contract FP6-027026). This paper expresses the view of the authors but not necessarily the view of K-SPACE.

## REFERENCES

- [1] T. Ebrahimi, Y. Abdeljaoued, R. Figueras i Ventura, and O. Divorra Escoda, "MPEG-7 Camera", IEEE Proc. Int. Conference on Image Processing (ICIP), Thessaloniki, Greece, Oct. 2001.
- [2] S.-F. Chang, T. Sikora, A. Puri, "Overview of the MPEG-7 Standard", IEEE Trans. on Circuits and System for Video Technology, Vol. 11, No. 6, June 2001.
- [3] "MPEG-7 Overview", ISO/IEC JTC1/SC29/WG11 WG11N6828, Oct. 2004.
- [4] <http://www.eptascape.com/products/eptacam.htm>
- [5] [http://www.matrox.com/imaging/products/iris\\_pseries/home.cfm](http://www.matrox.com/imaging/products/iris_pseries/home.cfm)