# 2D/3D knowledge inference for intelligent access to enriched visual content

Raluca-Diana Petre Sambra-Petre

**THESE DE DOCTORAT CONJOINT TELECOM SUDPARIS et L'UNIVERSITE PIERRE ET MARIE CURIE**

**Spécialité: Informatique et Télécommunications**

**Ecole doctorale:** Informatique, Télécommunications et Electronique de Paris

**Présentée par**

**Raluca-Diana ŞAMBRA-PETRE**

**Pour obtenir le grade de**
**DOCTEUR DE TELECOM SUDPARIS**

# MODELISATION ET INFERENCE 2D/3D DE CONNAISSANCES POUR L'ACCES INTELLIGENT AUX CONTENUS VISUELS ENRICHIS

**Soutenue le 18 Juin 2013 à Paris**

**devant le jury composé de :**

| | | |
|---|---|---|
| **Président de jury:** | **Madame le Maître de Conférences, HDR** | **Catherine ACHARD** |
| **Rapporteur:** | **Monsieur le Professeur** | **Marc ANTONINI** |
| **Rapporteur:** | **Monsieur le Professeur** | **Constantin VERTAN** |
| **Examinateur:** | **Monsieur le Professeur** | **Miroslaw BOBER** |
| **Examinateur:** | **Monsieur le Docteur** | **Olivier MARTINOT** |
| **Directeur de thèse:** | **Monsieur le Professeur** | **Titus ZAHARIA** |

**Thèse n°: 2013TELE0012**

# 2D/3D KNOWLEDGE INFERENCE FOR INTELLIGENT ACCESS TO ENRICHED VISUAL CONTENT

# ACKNOWLEDGMENT

First, I would like to express my gratitude to my thesis supervisor, Professor Titus Zaharia, for his guidance throughout my thesis and for all fruitful discussions that we had. His wide knowledge and his logical way of thinking have been of great value for me. I would also like to thank him for being attentive towards me and providing me with invaluable encouragements.

I would like to address my special thanks to the members of my Ph.D. defense committee.

To Professor Catherine Achard, from Pierre and Marie Curie University, President of this Jury, I would like to express all my thanks for her interest in my research work.

I would also like to express all my gratitude to Professors Marc Antonini from Nice Sophia Antipolis University and to Professor Constantin Vertan, from POLITECHNICA University of Bucharest, for accepting the hard task of being reviewers. Their reviews, comments and fruitful suggestions helped me to improve the manuscript and to give it its final shape.

A special thank to Miroslaw Bober from University of Surrey who provided encouraging and very constructive feedback.

I am very thankful to Mr. Olivier Martinot, Research Department Director at Alcatel-Lucent Bell Labs, for our collaboration and for accepting to be member of the examination jury.

I would also like to mention that this work has been performed within the framework of the UBIMEDIA Research Lab, established between Institut Mines-Télécom and Alcatel-Lucent Bell-Labs.

My gratitude goes also toward my colleagues and friends, from ARTEMIS and beyond, which supported me during all these years. To Madame Evelyne Tarroni I would like to thank for her help and patience in efficiently solving all necessary administrative problems.

I also like to thank my colleagues Ruxandra and Bogdan for sharing with me much more than just an office. One of the best things that my PhD experience brought me is the beautiful friendship with Afef. I thank her for always lifting my spirit when I felt down and overwhelmed with problems.

A very special thank to my dear friend Alina, which has always been available to hear about my difficulties and who offered me unconditional support.

I will never thank enough my husband, Andrei, who gave me strength and motivation to overcome all difficulties. He offered me moral support, as well as priceless technical and linguistic advices and helped me to improve my work and my experience.

I would finally like to thank my family for their continuous love and support. A special thank you goes to my grandfather who, at the age of 81 years, tried to understand my doctoral work and to help me with his ideas.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# I. INTRODUCTION

Graphics hardware and software development domains have seen a vast expansion in the last decades. Due to the spectacular evolution in digital technologies, the amount of multimedia content (*i.e.,* still images, videos, 2D/3D graphics…) available today is continuously increasing. Within this context, disposing of powerful search and retrieval methods becomes a key issue for intelligent and efficient access to audio-video material.

When large databases of multimedia content are involved, user access to specific material of interest is not possible without efficient search engines.

Multimedia retrieval tools may be divided into two main families: concept-based and content-based techniques. In the first case, some metadata, such as keywords and tags, are associated to the multimedia data and the retrieval is performed starting from textual indices. However, the linguistic barriers represent an important drawback of such approaches. Also, a prior, manual annotation is required, which is a tedious and highly subjective process.

In contrast, in content-based retrieval the search process analyzes the actual content of the data (*e.g.,* colour, shape, texture, and motion feature for describing the visual appearance…). By using computer vision algorithms, the salient features of the multimedia content are revealed and transformed into numerical representations, so-called descriptors. Such descriptors allow an objective comparison of different audio-video materials, making it possible to perform similarity retrieval of multimedia data.

Moreover, such objective descriptors can be used for classification purposes. More precisely, they can be employed to evaluate the similarity between multimedia materials. Thus, if we dispose of categorized content, we can analyse its similarity with respect to any unknown multimedia material in order to automatically assign one of the existing classes to the new material.

A large number of existing methods uses prior knowledge in order to accomplish this kind of objective. Such approaches generally exploit machine learning (ML) techniques. They automatically learn to recognize complex structures based on sets of both positive and negative

examples. ML techniques involve two main stages. First, some characteristic features are extracted starting from a set of examples involved in the training phase. Then, these features are used in order to recognize new cases. Such methods should be able to generalize the features of a given class while ensuring the accuracy of the recognition process.

However, when a large number of categories is involved, the number of recognition criteria (and implicitly the number of exploited features) increases and thus the computational complexity may become intractable. In addition, in order to allow generalization, a large variety of examples should be used in the training set. Moreover, a given object may present highly different appearances due to pose variation. Thus, for effective ML-based 2D object classification purposes, the training set should include not only a variety of examples but also different instances of the same object, corresponding to different poses (Figure I.1).


*Figure I.1 Different views of a 3D object representing a bicycle.*
*The first (profile) and the last (front) views are completely different in terms of 2D shape, but can be related if the 3D model is available.*

This thesis specifically addresses the issue of still image and video object classification. In our work, we propose to overcome the limitation of existing ML approaches by exploiting the information contained in categorized 3D model repositories. In order to enable the transfer of semantic labels from 3D models to 2D objects, shape-based 2D/3D indexing methods are employed. The 2D/3D description (also known as view-based description) consists of characterizing a 3D model through a set of 2D views. Further, the shape features are extracted and used in the recognition process. The choice of using only the shape information is motivated by the fact that the shape is a feature shared by all object within a class, compared to the colour or the texture which may change from one object to another. The availability of 3D models (involved in the recognition process) is not an issue because nowadays a large amount of 3D object collections

can be found on the internet. Moreover, these repositories are already classified, which represents an important advantage within the proposed framework.

The main purpose of this thesis is the exploitation and evaluation of 2D/3D indexing techniques within the context of 3D model retrieval as well as for 2D or 2D + t object classification purposes.

Chapter II introduces the main 2D/3D indexing techniques. The background definitions and terminologies are here recalled and several important issues related to the view-based description process are discussed. A review of the state of the art methods is also presented.

Chapter III presents the 2D/3D indexing approaches adopted in our work. We notably introduce here a new clustering-based method for adaptive selection of representative views and a novel contour-based descriptor, so-called Angular Histogram (AH). The 3D model repositories exploited in our work are also presented and an objective intra and inter class variability analysis is proposed. The performances of the various 2D shape descriptors and viewing angle strategies retained are experimentally evaluated within a 3D model retrieval framework.

The object classification issue is addressed in the fourth chapter. The view-based indexing methods presented previously are here employed to allow semantic inference between 3D and 2D content. The underlying principle consists of exploiting the *a priori* knowledge contained in classified 3D models and to transfer it, with the help of view-based indexing, to unknown 2D objects. Such methods can be applied to both still objects (SO, *i.e.*, objects extracted from still images) and video objects (VO, *i.e.*, objects extracted from videos and composed of several instances). We propose here a classification framework which ensures fast computation and allows combining several indexing methods. A main contribution of our work, compared to state of the art 3D model-based classification approaches, is the capacity to deal with a large number of semantic classes (up to 161 categories). In order to experimentally evaluate the performances of the recognition framework, we have created several test sets, including objects extracted from real and synthetic images and videos.

In order to allow integrating the proposed 2D object recognition framework in real applications, we have also developed a segmentation approach, designed to assist the user to extract an object of interest from an image. The proposed method, presented in chapter V, adopts the scribble-based segmentation paradigm. The user interaction consists of specifying a set of lines, corresponding to both foreground and background scribbles. The segmentation process is based on colour distributions, estimated with Gaussian Mixture Models (GMM). In order to overcome the compression artefacts that may appear, a modified GMM model is proposed. The experimental evaluation demonstrates the superiority of the modified GMM model which is able to appropriately take into account compression artefacts.

Finally, an important aspect in 2D/3D object retrieval and recognition is to dispose of appropriate user interfaces. The proposed DIANA (*Digital Image Analysis aNd Annotation*) system is a Web platform integrating the various developments proposed in this thesis. Chapter VI presents the main tools and functionalities proposed by the DIANA platform.

Chapter VII concludes the manuscript, highlights the main contributions proposed in this work and opens some perspectives of future research.

# II. 2D/3D INDEXING

**Abstract.** *This chapter introduces the view-based 3D model indexing. The principle of the 2D/3D description methods is first presented. The background definitions and terminologies are briefly recalled and several important issues related to 2D/3D indexing are discussed.*

*In the second part of the chapter we review the state of the art methods. We propose a classification of the various approaches, based on the viewing angles selection strategy employed. We conclude with an analysis of the advantages and limitations related to each family of 2D/3D indexing methods.*

**Keywords:** *shape descriptor; projection strategy; 3D meshes; 2D/3D indexing; similarity measure.*

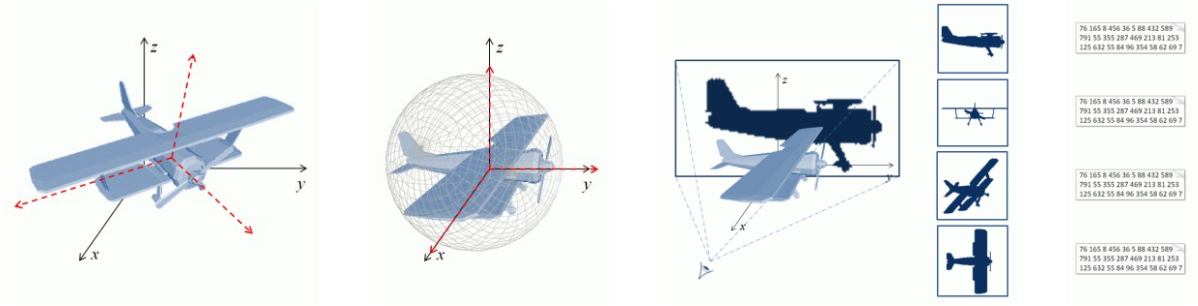
The concept of *2D/3D indexing* refers to a class of 3D model description methods. Their particularity is that a 3D model is not directly characterized in the 3D space, but instead through a set of 2D projection views, which provide the 2D appearance of the model from different perspectives/angles of view.

The underlying principle of 2D/3D indexing approaches is based on the following observation: two similar 3D models should present similar views when projected in 2D images from similar perspectives (*e.g.,* frontal projection, profile projection…). Such a strategy allows comparing two different 3D models through their corresponding 2D views. In addition, and more interestingly, such view-based approaches make it possible to compare not only 3D models, but also to match 3D models with 2D objects.

This chapter first presents the principle of the 2D/3D indexing paradigm, with the necessary theoretical background and main concepts involved. The state of the art methods are then described and analyzed, and the main methodological challenges that still need to be solved are identified.

## II.1. THEORETICAL BACKGROUND

The 2D/3D indexing includes several stages (Figure II.1). First, a pre-processing pose normalization step is required in order to ensure a canonical representation of the 3D mesh geometry. Next, the model is projected into 2D, resulting in a set of views. Finally, each view is described with the help of a 2D shape descriptor.



**a.** **b.** **c.** **d.**

*Figure II.1 Main stages of 2D/3D Indexing.*
*a.&b. Pose normalization; c. 3D-to-2D projection; d. 2D shape description.*

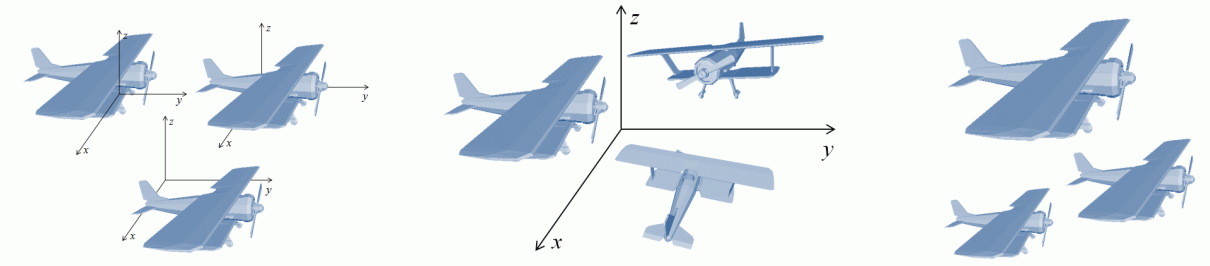
When analyzing this process, some fundamental questions rise up: how many projection views are needed to obtain an accurate representation of the considered 3D shape? Which are the angles of view of the model that optimally represent its shape? Which shape descriptors are suited for this purpose?

The various solutions proposed in the state of the art for each stage involved are detailed in the following sections.

### II.1.1. Pre-processing: normalization and invariance issues

Existing 3D models are most often specified with arbitrary orientations, positions and scales in the 3D virtual space. In our case, we have considered exclusively 3D models represented as 3D mesh models (*cf.* Annexe A1). Thus, the model geometry is specified by the set of 3D position of the mesh vertices, in a given 3D coordinate system. However, not all shape descriptors are intrinsically invariant to geometric transforms. Therefore, in order to ensure at least an extrinsic invariant behaviour, it is necessary to apply some position/pose/size normalization.

Thus, the objective is to prepare the 3D model for 2D/3D indexing by offering invariance with respect to similarity transforms (*i.e.,* translation, rotation, isotropic scaling and combinations of them – Figure II.2). After the normalization process, similar models should present similar size, orientation and position in the 3D virtual space. In addition, the normalization process should be robust to small, local deformations of the model.




**a.** **b.** **c.**

*Figure II.2 Affine transformations of a 3D Model.*
*a. translation; b. rotation; c. scaling.*

The normalization process includes the following three steps:

- **Centring**: consists of positioning the 3D model with respect to the origin of the coordinate system and ensures invariance to translation.
- **Alignment**: consists of orienting the 3D model in the virtual space, which ensures invariance to rotation.
- **Scaling**: consists of resizing the object in order to ensure scale invariance.

The following sections detail the main pose normalization techniques encountered in the literature.

#### II.1.1.1. Model centring

The centring consists in displacing the 3D model such that its "centre" coincides with the origin of the coordinate system. There exists several ways to define the centre of a 3D model.

##### II.1.1.1.1. The centre of the bounding box

A first approach [Paquet00] defines the centre of a 3D model as the centre $C_{BB}$ of its corresponding bounding box and is defined as:

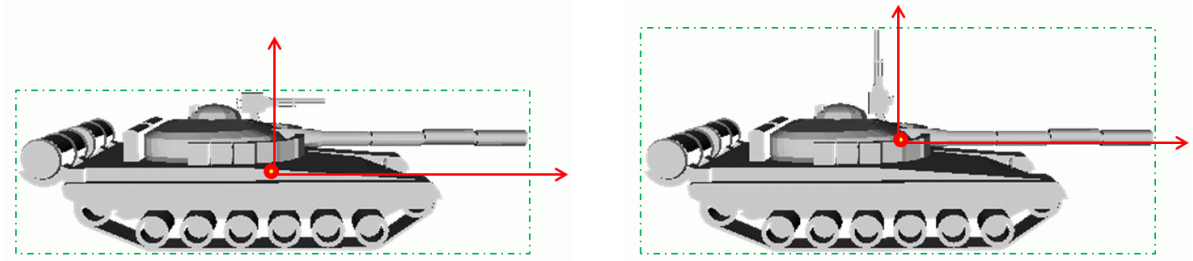
$$C_{BB} = \left(\frac{x_{max} - x_{min}}{2}, \frac{y_{max} - y_{min}}{2}, \frac{z_{max} - z_{min}}{2}\right), \quad (II.1)$$

where:

- $x_{min}$, $x_{max}$, $y_{min}$, $y_{max}$, $z_{min}$, $z_{max}$ respectively denote the minimum and maximum coordinate values of the 3D mesh vertices, along the $x$, $y$ and $z$ axes.

Let us observe that the centre of a 3D model's bounding box is not invariant with respect to rotations. Therefore, such a bounding box centring approach should be applied after the alignment phase.

The main drawback of the bounding box-based centring is the sensibility to minor shape modifications, as illustrated in Figure II.3. Here, two 3D models of tanks are presented, which correspond to the same real-life objects. The difference between them concerns the position of the tank machine gun, which points horizontally in Figure II.3a and vertically in Figure II.3b. As a result, the corresponding bounding boxes are significantly different and centring approach will lead to an erroneous result. Let us note that this situation is often appearing in practice, in the case of articulated shapes (*i.e.*, shapes composed of multiple parts that can independently exhibit rigid motion).



**a.** **b.**

*Figure II.3 Sensibility to minor modification of bounding box-based centring.*

### II.1.1.1.2. The gravity centre

The gravity centre ($G$) is also known as centre of mass or centre of inertia and represents the barycentre of the 3D mesh $M$, defined as described in the following equation:

$$G_{x,y,z} = \frac{1}{A}\sum_{i=1}^{N_T} A^i g^i_{x,y,z}\,, \quad (II.2)$$

where:

- $N_T$ is the number of the mesh triangles;
- $A^i$ represents the area of the $i^{th}$ triangle;
- $g^i_{x,y,z}$ are the coordinates of the gravity centre of the $i^{th}$ triangular face.

As the entire surface of the 3D model is taken into consideration when computing $G_{x,y,z}$, the barycentre method is less sensitive to minor modification of the shape than the bounding box approach. Thus, in our work we have considered that the centre of a 3D model is given by the barycentre, computed as described in Equation II.2.

### II.1.1.2. Pose alignment

The pose alignment phase aims at ensuring rotation invariance. For this purpose, an object-dependent, orthogonal 3D coordinate system needs to be constructed first. An orthogonal transform is then applied in order to transform the initial coordinate system in an object-dependent one.

The most commonly employed approach is to consider the system defined by the three *axes of inertia* of the 3D mesh. The axes of inertia, also known as *principal axes*, are obtained with the help of a *Principle Component Analysis* (PCA) approach [Mather04], [Schwengerdt97]. The PCA technique involves a mathematical procedure that transforms a set of possibly correlated variables (*e.g.*, the coordinates of the 3D model's vertices) into a set of uncorrelated variables, called principal components.

Various types of information and data can be used for 3D model alignment, including vertex coordinates and normal vectors.

In the PCA framework, such data are considered as realizations of a 3D random vector. A $m \times 3$ matrix, denoted by $X$, is constructed; it stores on each row a realization $X^t$ of the random vector:

$$X = \begin{bmatrix} X_x^1 & X_y^1 & X_z^1 \\ X_x^2 & X_y^2 & X_z^2 \\ \vdots & \vdots & \vdots \\ X_x^m & X_y^m & X_z^m \end{bmatrix} = \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^m \end{bmatrix} = [X_x \quad X_y \quad X_z]. \tag{II.3}$$

Most often, the number of observations $m$ is equal to the number of mesh vertices $V$ and the observations $X^t$ represent the $x$, $y$ and $z$ coordinate values of the mesh vertices.

The $(3 \times 3)$ covariance matrix $\Sigma$ of $X$ is then computed as described by the following equations:

$$\Sigma_{ij} = cov(X_i, X_j) = E\left[(X_i - \mu_i)^T \cdot (X_j - \mu_j)\right], \quad i, j \in \{x, y, z\}. \tag{II.4}$$

$$cov(X_i, X_j) = \sum_{k=1}^{m} (X_i^k - \mu_i) \cdot (X_j^k - \mu_j), \tag{II.5}$$

with

$$\forall\, i \in \{x, y, z\}, \quad \mu_i = E[X_i] = \sum_{k=1}^{m} \frac{X_i^k}{m}. \tag{II.6}$$

In Equation II.4, $E$ [.] and (.) $^T$ respectively denote the statistical expectation and the matrix transpose operators.

By definition, the covariance matrix is symmetric and positive definite. Thus, it can be diagonalized with the help of an orthogonal transform constructed as a matrix $V$ with the eigenvectors as columns. In addition, the corresponding eigenvalues are real and positive numbers, which provide a measure of extent of the object along each eigen-direction. More precisely, the following equation is satisfied:

$$V^{-1}\Sigma V = D. \qquad (II.7)$$

where $D$ is a diagonal matrix storing the eigenvalues of Σ.

The eigenvectors $V_i$ that compose the matrix $V$ are also called principal axes or axes of inertia. The planes defined by each couple of eigenvectors are referred to as principal planes.

Finally, the 3D object pose alignment is accomplished by applying the following transformation:

$$X^T \rightarrow V \cdot X^T. \qquad (II.8)$$

Let us note that depending on the space where the PCA is performed, two families of PCA methods can be distinguished: discrete and continuous PCA. The discrete case corresponds to the matrix formulation presented here above, where a finite set of observation is employed. Matrix $X$ can store the coordinates of the vertices, the coordinates of the face gravity centres or the normal vectors associated with polygonal face. The main limitation of such approaches is the dependence on the distribution of the vertices, in the case where the mesh vertices are irregularly distributed. In order to overcome this drawback, [Paquet00] propose to weight the contribution of each gravity centre by the area of the corresponding triangular face.

In the case of continuous PCA (CPCA) approach [Vranic01], the analysis is performed in the infinite-dimensional space of the mesh surface $\sigma$. Here, the covariance matrix is defined as the matrix of second order surface moments.

$$\Sigma = \begin{bmatrix} m_{200} & m_{110} & m_{101} \\ m_{110} & m_{020} & m_{011} \\ m_{101} & m_{011} & m_{002} \end{bmatrix}, \qquad (II.9)$$

where:

- $A$ is the total area of the 3D mesh surface;
- $p = (x_p, y_p, z_p)$ is a point on the surface of the 3D mesh;
- $m_{ijk}$ is the moment of order $n = i + j + k$, defined as:

$$m_{ijk} = \frac{1}{A} \iint_{p \in \sigma} x_p^i y_p^j z_p^k ds. \qquad (II.10)$$

Compared to discrete PCA approaches, CPCA is more accurate, at the price of an increased computational complexity [Chaouch09].

In our work we have adopted the weighted PCA methods [Paquet00] which ensures good results with a limited computational complexity. Figure II.4 presents some examples of 3D models and corresponding principal axes for various shapes, obtained with the weighted PCA approach. We observe that in such cases, the PCA alignment corresponds to intuitive notions, such as frontal, profile and bottom views.

Whatever the approach considered, the main drawback of the PCA-based alignment is related to its incapacity of dealing with miss-alignment problems, in the case where different axes of inertia are computed for similar 3D models [Zaharia01, Tangelder04].

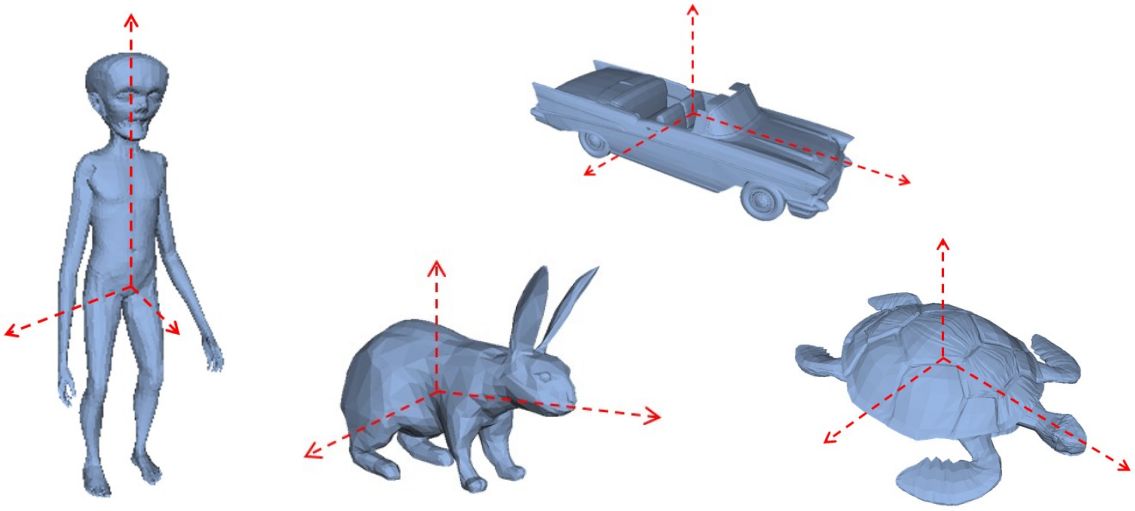


*Figure II.4 Example of principal axes determined with weighted PCA.*

Three miss-alignment situations may appear:

- The two sets of principal axes have significantly different orientations. This problem, illustrated in Figure II.5 a and b, is a stability issue, which arises in the case where some object details affect the symmetry of the objects.
- The same orientations are detected for both models, but the order of the principal components is not the same. Let us note that the ordering of the eigenvectors in the construction of the transform matrix $V$ in Equation II.7 is important, since different orders lead to completely different transforms. Most of the time, such an ordering is performed with respect to the values of the corresponding eigenvalues. However, such a mechanism can lead to erroneous alignments, as illustrated in Figure II.5 a and c.
- The principal components have the same directions for both models, but different orientations: solely the PCA cannot uniquely determine the orientation of the eigenvectors, but gives only their direction. This ambiguity leads to miss-alignments in the case of non-symmetric objects, as illustrated in Figure II.5 a and d.

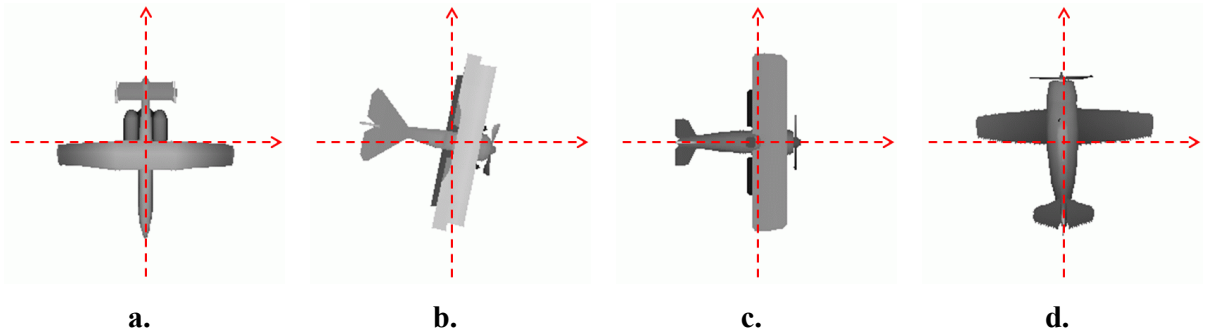



*Figure II.5 Example of similar models presenting differently detected principal axes. a. the reference model; b. the principal axes has a different orientation w.r.t the model's geometry; c. the principal axes have different order compared to those of the reference model; d. the principal axes have different directions than those of the reference model.*

Another alignment approach is proposed in [Papadakis08]. Let $M$ be a mesh, $A$ its total surface area, and $\alpha$, $\beta$ and $\gamma$ rotation angles around the $Ox$, $Oy$ and $Oz$ axes, respectively. Let $M(\alpha,\beta,\gamma)$ denote the rotated model. The aim of the so-called *rectilinearity-based alignment* is to determine the set of angles $(\alpha,\beta,\gamma)$ maximizing the fallowing ratio:

$$(\alpha,\beta,\gamma) = arg\min_{\alpha,\beta,\gamma} \frac{A(M)}{A_{Pxy}\big(M(\alpha,\beta,\gamma)\big) + A_{Pyz}\big(M(\alpha,\beta,\gamma)\big) + A_{Pzx}\big(M(\alpha,\beta,\gamma)\big)}, \qquad (II.11)$$

where:

- $A_{Pxy}\big(M(\alpha,\beta,\gamma)\big)$, $A_{Pyz}\big(M(\alpha,\beta,\gamma)\big)$ and $A_{Pzx}\big(M(\alpha,\beta,\gamma)\big)$ represent the areas of the projections on the $(xy)$, $(yz)$ and $(zx)$ planes of the rotated model.

Compared to discrete PCA, the rectilinearity-based alignment present higher computational complexity.

### II.1.1.3. Model scaling

The objective of the model scaling stage is to determine an intrinsic scale for each 3D model in order to normalize the object in size and achieve representation invariance.

#### II.1.1.3.1. Bounding sphere approach

In order to determine the bounding sphere, the farthest vertex from the centre of the object is first determined and the corresponding distance ($d_{max}$) is computed. The maximum distance $d_{max}$ represents the radius of the bounding sphere. The normalization is accomplished by resizing the model to the unit sphere.

This normalization technique is invariant to rotation and translation. However, such a naive approach is highly sensitive to minor changes of the 3D mesh or to articulated shape (*cf.* Section II.1.1.1.1) which can modify the value of $d_{max}$.

In order to overcome such a limitation, different techniques exploit the eigenvalues computed in PCA in order to statistically determine the intrinsic scale.

#### II.1.1.3.2. Eigenvalue-based normalization

In [Elad02], authors propose to rescale the model such as the largest eigenvalue ($D_{11}$) becomes equal to 1.

Such an approach presents the same drawback as PCA: in some cases, for two similar models, the principal axes can have different directions and orientations. Thus, the corresponding eigenvalues are different and the scale normalization is not the same for both models.

In order to overcome this drawback, in [Zaharia02] the authors propose to consider the three eigenvalues $D_{11}$, $D_{22}$, $D_{33}$ (*cf.* Equation II.7). The object is resized such as the expression $1.5\sqrt{D_{11} + D_{22} + D_{33}}$ becomes equal to 1.

### II.1.1.3.3. Distance to surface

In [Vranic04], authors propose to determine the intrinsic scale of the model based on the mean distance $\overline{d_{PP-\sigma}}$ between the object surface σ and the principal planes.

$$\overline{d_{PP-\sigma}} = \sqrt{\frac{\overline{d_x^2} + \overline{d_y^2} + \overline{d_z^2}}{3}}, \qquad (II.12)$$

where:

– $\overline{d_x^2}, \overline{d_y^2}, \overline{d_z^2}$ represent the mean distance from the model surface to (*yz*), (*xz*) respectively (*xy*) planes:

$$\overline{d_t} = \frac{1}{A} \iint_{p \in \sigma} |p_t - G_t| ds; \qquad t \in \{x, y, z\}, \qquad (II.13)$$

where:

– $p=(p_x,p_y,p_z)$ is a point on the surface $\sigma$ of the model;
– $G=(G_x,G_y,G_z)$ is the centre of the 3D model;
– $A$ is the area of the 3D model surface;

Normalization is accomplished by resizing the model such that the computed distance $\overline{d_{PP-\sigma}}$ becomes equal to a given value.

Another distance-based method [Vranic04] proposes to define the intrinsic scale based on the mean distance $\overline{d_{C-\sigma}}$ from the centre $C$ of the model to its surface σ.

$$\overline{d_{C-\sigma}} = \frac{1}{A} \iint_{p \in \sigma} \|p - g\| ds. \qquad (II.14)$$

Here again, the 3D model is resized such as the distance $\overline{d_{C-\sigma}}$ becomes equal to a given value.

Compared to other scale normalization techniques, the distance to surface-based approach is more robust to small shape deformations but computationally more complex.

In our work, we have adopted the eigenvalues-based scaling due to its robustness and low computational complexity required within our framework (the eigenvalues are already computed in the alignment phase).

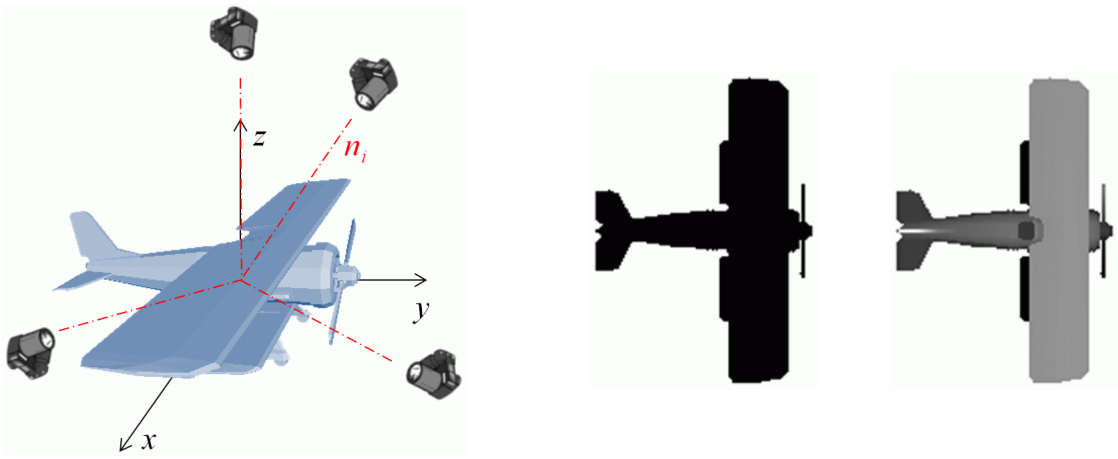
Once normalization in orientation, size and position is completed, the 2D views of the 3D model can be acquired, with the help of a 3D-to-2D projection mechanism, as described in the following section.

### II.1.2. 3D-to-2D projection

The projection of the 3D model into 2D images represents a key phase in 2D/3D indexing process. The mesh model $M$ is projected and rendered in 2D from $N_P$ different viewing angles (*i.e.*, positions of a virtual camera in the 3D space) (Figure II.6 a), resulting in a set of $N_P$ projections, denoted by $P_i(M)$, with $i$=1..$N_P$.

The projection may be a binary image (the silhouette of the object) or a gray level image representing the depth map (Figure II.6 b and c). However, only the silhouette representation allows matching between 2D and 3D content, as there is no depth information available in 2D images.

A viewing direction $\{n_i\}$ is associated to each viewing angle; it represents the direction of the straight line that connects the position of the camera with the origin of the considered Cartesian system (which, after normalization, coincides with the centre of the 3D model).




**a.** **b.** **c.**

*Figure II.6 3D-to-2D projection.*
*a. viewing directions; b. example of silhouette image; c. example of depth image.*

When projecting a 3D model for 2D/3D indexing purpose, the key aspects that need to be considered concern the number of considered views and the specification of the viewing angles.

#### II.1.2.1. The number of views

The number of views ($N_P$) defines how many views are generated for each object. A large number of views ensures good description of the 3D model, which is suited for indexing and retrieval purposes. However, the associated computational aspects have to be taken into account. The time required for projection and descriptor extraction, as well as memory/storage requirements, are proportional to the number of views. Therefore, a "good" balance has to be ensured between the level of detail of the 2D/3D representation and the computational costs involved.

### II.1.2.2. The viewing angles

The viewing angles (directions) $\{n_i\}$ define the perspectives used to generate the set of views. Obviously, there is an infinity of potential viewing angles. However, some views are more salient than other. In addition, there exists couples of very similar views. For 2D/3D indexing purposes, only the shape information is exploited. Under the assumption of a parallel projection model, the silhouettes obtained from opposite perspectives (*e.g.,* front and back, left and right, up and down) represent mirror reflections of each other. In order to avoid representation redundancy, solely a demi-space around the model should be considered for specifying the views.

There exist three main families of viewing angle selection strategies. A first and largely popular family makes the assumption that the most salient views correspond to the projection onto the principal planes. Therefore, the viewing directions correspond to the principal axes of the model. Throughout this work, this class of approaches will be referred to as PCA-based viewing angle selection.

The second family of approaches considers that there are no preferential viewing directions. Therefore, the viewing angles are distributed as evenly as possible around the object. Most often, in this case the camera repartition is obtained using the vertices of a regular polyhedron (*e.g.,* octahedron, dodecahedron...).

Finally, the third class of viewing angles selection strategy attempts an intelligent selection of the views. First, a large number of evenly distributed views is generated. Next, the similarity between views is analyzed and a subset of *representative views* is selected to represent the 3D model. In order to measure the similarity between two views, each view is characterized with the help of shape descriptors. A discussion on descriptor extraction and similarity measurement can be found in the following sections.

The main disadvantage of the representative views strategies is the computational costs which includes the achievement and description of a large number of views as well as the high number of pairwise comparison between couples of views needed. A second drawback is related to the intrinsic nature of such strategies: two different 3D models will be described by completely different, object-dependent views. How is it possible, in this case, to specify a matching strategy that can be exploited for 3D model retrieval purposes?

Whatever the viewing angle selection strategy adopted, the 2D shape descriptors involved for describing the resulting silhouettes strongly influence the discriminative power of the representation. Descriptor-related aspects are discussed in the following section.

### II.1.3. 2D shape descriptors

Descriptors are mathematical representations of the salient features of the multimedia content which allow an objective and quantitative comparison between various objects. For 2D-3D matching purposes the shape attribute represents the most popular feature considered.

Let us first recall the various criteria enounced in the literature that a shape descriptor should satisfy [Zaharia04], [Tangelder04]:

- **Scope:** the descriptor should be able to characterize any kind of shape.
- **Uniqueness:** a given shape is described by only one descriptor and a given descriptor corresponds to a single shape.
- **Efficiency:** for a large database, the system should be able to quickly describe models and to perform a fast retrieval. Therefore, rapidity and low complexity of the feature extraction is necessary. Some descriptors allow early rejection of non-similar models based on a subset of features. This ability is useful for speeding up the matching process.
- **Robustness:** the descriptor should be almost insensitive to noise and to small extra features.
- **Sensitivity:** a descriptor should present the capacity to describe and take into account even fine details of the shape.
- **Discrimination power:** the descriptor should be able to capture the properties that best discriminate the shape.
- **Multiresolution support:** the descriptor should not depend on the resolution of the shape.
- **Ability to perform partial matching:** such a feature is useful in the case of incomplete shapes, such when a part of the object is invisible (for example, in the case of occluded objects).
- **Geometric and topologic invariance:** the description of a given object should not depend on the scale, orientation or position that it has in the image.
- **Agreement with the human perception:** it is important that the similarities given by a descriptor correspond to the human perception.
- **Ability to match articulated objects:** a descriptor should extract similar features for different instances of an articulated object.
- **The storage size of the descriptor:** an important property especially in the case of big databases where a large number of descriptors must be stored.

Such criteria are taken into account in various manners by the different approaches of the state of the art, which can be categorized within three main families of 2D shape descriptors:

- **Region shape descriptors:** in this case, the input information to be described represents the support region of the 2D shape (Figure II.7b).
- **Contour shape descriptors:** solely the external contour information is retained (Figure II.7c). Consequently, the principal limitation of such approaches concerns the limited area of applicability, since shapes with more complex topologies (*e.g.*, holes, multiple connected components...) cannot be accurately described.
- **Graph-based descriptors:** the principle consists of setting-up a part-based representation, achieved with the help of a graph or a multi-level graph (Figure II.7d). In some cases, the nodes of the graph may store attributes of the corresponding region of the 2D object. The advantage of such elaborate representations is the possibility to represent accurately complex shape and, in particular, articulated shapes. However, specifying fast similarity measures for graph-based representations, which is mandatory for indexing and retrieval applications, is still an open issue of research.

**a.** **b.** **c.** **d.**

*Figure II.7 Different representations of a 2D shape.*
*a. The 3D model; b. Region support; c. Contour representation; d. Graph representation.*

The various 2D shape descriptors of the state of the art are presented and discussed in Section II.2.

Whatever the type of descriptor considered, it is of outmost importance to define appropriate similarity measures between them.

### II.1.4. Similarity measurement

The similarity measurement employs the mathematical representation of the shape features (*i.e.*, the descriptor) in order to associate a quantitative appreciation to the similarity between shapes.

Depending on the shape descriptor considered, one of the following similarity measurement methods can be used:

- **Distance (metric)-based methods**, suitable for feature vector representation and supposed to compute metrics such as the Euclidian distance.
- **Graph matching methods,** specifically adapted for graph-based representations.

#### II.1.4.1. Distance metric-based method

The distance metric measures the dissimilarity between two vectors. A small value denotes that the two vectors are very similar while higher values correspond to dissimilar vectors. In order to ensure good similarity estimation, a distance metric must satisfy several properties, recalled here below.

##### II.1.4.1.1. Distance metric properties

Let *X, Y, Z* be three vectors in a n-dimensional space and $d(X,Y)$ a function defined as

$$d: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}; \qquad X,Y,Z \in \mathbb{R}^n. \tag{II.15}$$

Then, the function *d* is a *distance* if it satisfies the following properties:

- **Identity:** $d(X,X) = 0$; the distance between two identical vectors should be equal to zero.
- **Positivity:** $d(X,Y) \geq 0$; the distance between two different vectors should always have a positive value.

- **Symmetry:** $d(X,Y) = d(Y,X)$; the distance from *X* to *Y* should be equal to the distance from *Y* to *X*.
- **Triangle inequality:** $d(X,Z) \leq d(X,Y) + d(Y,Z)$; the distance from *X* to *Z* is at most as large as the sum of the distances from *X* to *Y* and from *Y* to *Z*.
- **Transformation invariance:** $d(g(X),Y) = d(X,Y)$, where *g* is a transformation in a given group. In the particular case of shape description, the group of similarity transforms is most often considered. This means that the distance between two shapes is independent of their position, size or orientation.

Various distance metrics can be used. They are recalled in the following section.

#### II.1.4.1.2. Distance metrics

Let $X = (x_1, x_2, \ldots x_n)$ and $Y = (y_1, y_2, \ldots y_n)$ be two points in the $\mathbb{R}^n$ space. Several metrics are defined in order to measure the distance between *X* and *Y*:

**Minkowski distance of order *p*:**

$$d(X,Y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}. \qquad (II.16)$$

**Manhattan distance ($L_1$ norm):**

$$d(X,Y) = \sum_{i=1}^{n} |x_i - y_i|. \qquad (II.17)$$

**Euclidean distance ($L_2$ norm):**

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \qquad (II.18)$$

**Weighted Euclidean distance:**

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2}, \qquad (II.19)$$

where $w_i$ represents the weight of each component of the *n*-dimensional space.

**Hausdorff distance:**

$$d(X,Y) = \max\left\{ \sup_{x \in X} \inf_{y \in Y} d(x,y), \sup_{y \in Y} \inf_{x \in X} d(x,y) \right\}, \qquad (II.20)$$

where:

– *d* is a distance metric, such as Euclidian distance;

- $\inf_{x \in X} d(x, y)$ represents the shortest distance between $X$ and a given element $y$ of $Y$ and is given by the closest element $x$ of $X$ to $y$;
- $\sup_{y \in Y} \inf_{x \in X} d(x, y)$ represents the longest distance from $Y$ to $X$;

Let us note that the Hausdorff distance [Atallah83, Dubuisson94] is very sensitive to noise, since even a single outlier can affect its value.

**Earth mover's distance (EMD):** represents the minimum effort required to transform the distribution of $X$ into the distribution of $Y$. The distributions are interpreted as mounds of sand and the distance represents the cost of turning one mound into the other, knowing that moving an amount $\varepsilon$ of earth from a point $i$ to a point $j$ takes $\varepsilon \cdot d(i, j)$ effort (with $d$ a distance metric such as the Euclidian distance) [Rubner00].

**Kullback-Leibler divergence (KLD):** is not a real distance because it is a non-symmetric measure. $X$ and $Y$ are considered as two distributions having the associated codes $C_X$, respectively $C_Y$. KLD measures the number of extra bits required to code samples of $X$ using $C_Y$ rather than using $C_X$.

### II.1.4.2. Graph matching methods

If the shape of 2D objects is represented by a graph, then a graph matching procedure is necessary in order to compare the two shapes.

The aim of a graph matching method is to determine the best correspondence between the two graph representations. The resemblance level between graphs is given by a function which measures the similarity between couples of corresponding vertices and edges. The graph matching approach can also be seen as an energy minimization algorithm [Bengoetxea02].

There exist two classes of matching methods. When the two graphs have the same size (*i.e.*, the same number of vertices) an isomorphic mapping can be found between them. This family is called *exact matching*. On the contrary, when the two graphs have different sizes, the one-to-one correspondence becomes impossible. This case is referred to as *inexact matching*.

The combinatorial nature of the graph matching approaches leads to high computational complexity, which most often is NP-complete [Garey79, Conte04].

Let us now detail how the various aspects presented above are taken into account in the literature.

## II.2. STATE OF THE ART

The literature presents a large variety of 2D/3D indexing methods, mainly developed for 3D model retrieval purpose. Since the choice of viewing directions is a fundamental issue for successful 2D/3D description, we have categorized various families of approaches with respect to the viewing angle selection strategy (*cf.* Section II.1.2.2). The classification adopted in our work includes:

- **Methods using PCA-based projections** make the underlying assumption that the views corresponding to the projection on the principal planes present higher relevance than other views.
- **Methods using evenly distributed viewing angles** offer the same importance to all the views around the model.
- **Methods using representative views** include a clever selection of the views used in the 2D/3D description.

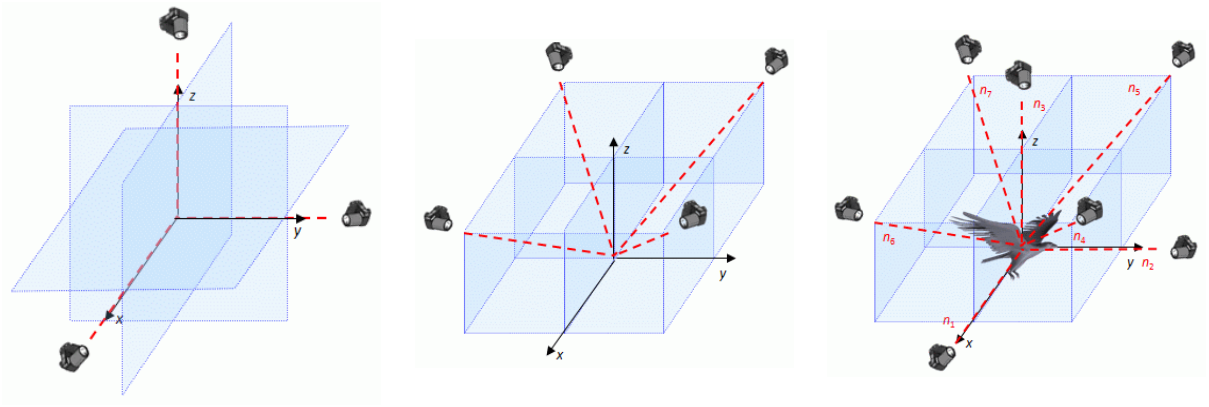
Let us begin our analysis with the PCA-based projection approaches.

### II.2.1. Methods using PCA-based projection

As a representative of the 2D/3D shape-based retrieval approaches, let us first mention the *MultiView* description scheme (DS) proposed by MPEG-7 standard [Bober02, Manjunath02, ISO/IEC02].

The pre-processing stage involves translation and scaling, the 3D object being transformed such that its gravity centre coincides with the coordinate system origin and fits the unit sphere. The rotation invariance is achieved by applying a principal component analysis (*cf.* Section II.1.1.2).

Three views are generated by projecting the 3D model onto the principal planes (Figure II.8a). Four secondary views can be added in order to ensure better description. The secondary viewing directions correspond to the diagonal of the octants defined by the principal planes (Figure II.8b). Figure II.8c illustrates the repartition of the seven cameras and the corresponding viewing directions.




**a.** **b.** **c.**

*Figure II.8 Selection of the principal and secondary axes.*
*a. The three principal viewing directions which corresponds to the first three principal axes;*
*b. the secondary viewing angles; c. the seven cameras reparation;*

Concerning the shape descriptors, let us first mention the MPEG-7 framework [ISO/IEC02], where two different MPEG-7 2D shape descriptors, the *Contour Shape* (CS) and the *Region Shape* (RS), have been considered.

In the case of the RS descriptor, the object's support function is decomposed within a base of *Angular Radial Transform* (ART) functions [Kim99]. Thus, the image is be represented as a weighted sum of ART coefficients. In order to achieve rotation invariance, solely the absolute values of the coefficients are used. The similarity measure simply consists of $L_1$ distances between ART coefficients. The 2D-ART is invariant under similarity transforms, and it is suitable for meshes of arbitrary topologies, which can present holes or multiple connected components under the projection operator. A more detailed presentation of the RS descriptor can be found in Section III.2.2.1.

The second 2D shape descriptor promoted within the MPEG-7 DS is *Contour Shape* (CS), which employs the *Contour Scale Space* (CSS) representation [Mokhtarian92]. More restrictive, the MPEG-7 CS descriptor assumes that the shape of the object can be described by a unique, closed contour. The descriptor is obtained by successively convolving an arc-length parametric representation of the curve with a Gaussian kernel. The curvature peaks are thus robustly determined in a multi-scale analysis process and serve to characterize the contour shape, with their value and corresponding position (expressed as curvilinear abscise).The associated similarity measure between two contours in CSS representation is the EMD [Rubner00]. A more detailed discussion on the CS descriptor is presented in Section III.2.2.4.

Whatever the 2D shape descriptor considered, when comparing two 3D models $M_A$ and $M_B$, a distance value $e_{ij}$ is obtained for each pair of views $P_i(M_A) - P_j(M_B)$. An error matrix $E=(e_{ij})$ is thus computed. The global similarity measure between the two models is then defined as:

$$d(M_1, M_2) = \min_{p \in \Pi}\{Trace[p(E)]\}, \qquad (II.21)$$

where

- $\Pi$ represents the set of all possible permutations between the columns of matrix $E$;
- $p$ is a permutation in $\Pi$;
- $p(E)$ represents the permuted version of matrix E.

Let us note that such a similarity measure is highly expensive since the number of possible permutations is $N_P!$, where $N_P$ is the number of projections. In practice, such a similarity measure can be applied only for a reduced number of views and becomes computationally un-tractable when the number of views increases (3!=6, 7!=5040).

In [Mahmoudi02], authors re-visit the MPEG-7 CSS representation. Here, the contour of each projection image is represented in CSS and decomposed into a set of segments called *tokens*, *i.e.*, sets of 2D points delimited by two inflexion points. The tokens are then clustered and hierarchically organized in a M-Tree structure [Ciaccia97]. This organization allows to considerably decrease the computation time, as proved by their experimental evaluation. To compare two tokens, a sum of geodesic distances between points is computed. The obtained descriptor is intrinsically invariant to similarity transforms.

The M-Tree-based CSS algorithm was further developed in [Mahmoudi07], where a Bayesian voting procedure is employed. To each part $p_i$ of the contour is associated a posterior probability $Pr(p_i|P_j(M))$ that reflects what is the chance to have a given view $P_j(M)$ in the image, knowing the presence of the part $p_i$. Based on the posterior probability, a rank $R(Q, P_j(M))$ can be associated to query image $Q$ with respect to a given view $P_j(M)$. The rank denotes the similarity between the query and the view and is calculated as the sum of posterior probabilities $Pr(p_i|P_j(M))$ for all parts $p_i$ composing the contour of $Q$. A notable consequence of the Bayesian voting procedure is that it allows partial matching. Also, the experimental evaluation presented in [Mahmoudi07] has shown that the Bayesian approach increases the performance of the M-Tree-based CSS algorithm.

Another method based on multi-scale shape representation is proposed in [Napoléon08]. Here, the authors employ a 2D/3D approach based on the MCC (*Multi-scale Convexity/Concavity*) representation introduced in [Adamek04]. The 3D object is scaled with respect to its bounding sphere and CPCA is applied in order to normalize the pose of the model. A number of three to nine silhouettes are then computed. The viewing directions used correspond to the three principal axes and to their six bisectors.

Each silhouette is further represented by its contour, normalized to $N = 100$ sample points. As in the case of the CSS descriptor, a scale-space analysis is performed here. Each silhouette contour is successively convolved with $K = 10$ Gaussian functions, with increasing kernel bandwidths $\sigma$ (Figure II.9b). Then, the displacement $d(u, \sigma_i)$ of the position of sample $u$ between two consecutive scale levels $\sigma_{i-1}$ and $\sigma_i$ is computed. The MCC representation is a $100 \times 10$ matrix composed of the displacements of all 100 contour sample points for the 10 scale levels (Figure II.9c).

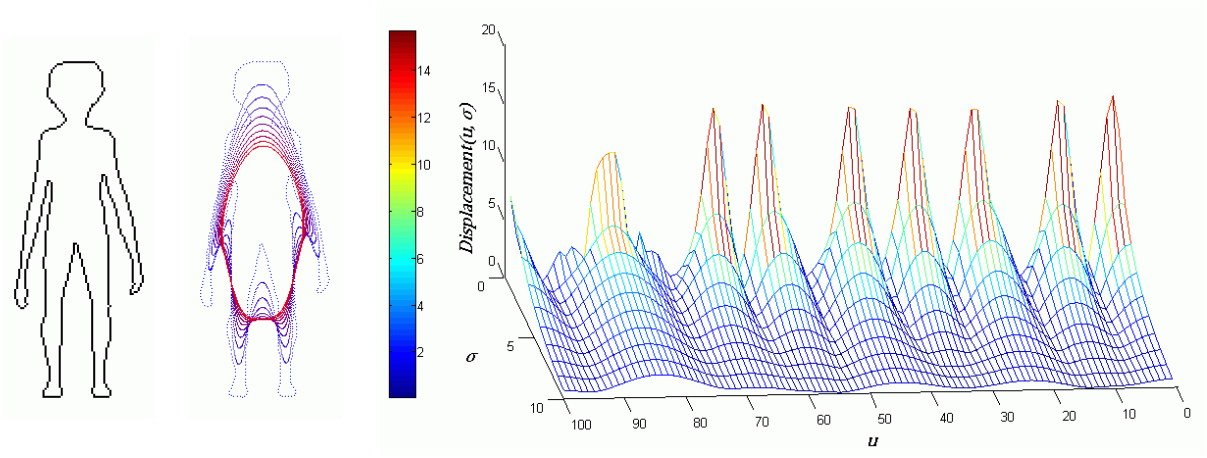
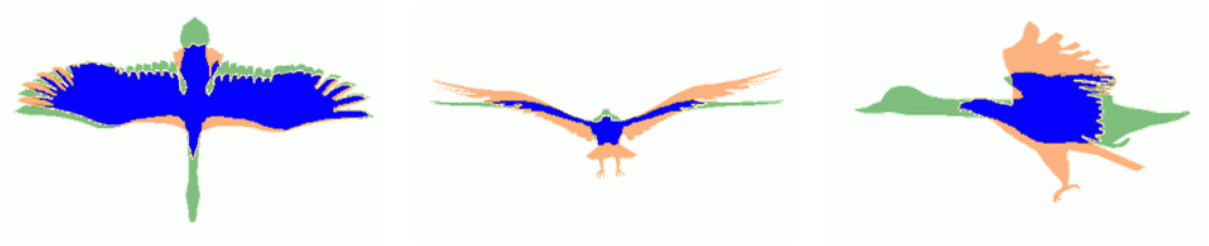



*Figure II.9 MCC-based representation.*
*a. the initial contour; b. the convolved contours; c. the MCC representation;*

The similarity measure used to compare two MCC representations is given by the $L_1$ distance. In order to ensure 2D rotation invariance, all the 100 possible matching between two sets

of samples are tested. Furthermore, as the order or the direction of the principal axes may present some errors, 48 possible poses of an object (corresponding to 6 possible permutations of the principal axes and to 8 possible orientation configurations) are tested. Thus, the computation complexity represents the main drawback of the MCC descriptor. Also, the size of the feature vector (*i.e.,* 1000 values/descriptor × 9 views) may become a disadvantage when large 3D model databases are involved.

In [Napoléon07] the so-called *Silhouette Intersection* (SI) method is proposed. Only the three views, corresponding to the CPCA principal directions are here retained. The signature of a model is simply constituted by the three binary silhouettes obtained. The distance between two silhouette images is defined as the number of pixels belonging to the symmetric difference [Alt98] of the two silhouettes (*i.e.,* the green and orange pixels in Figure II.10). The global distance between two 3D objects is defined as the sum of silhouette distances between pairs of images corresponding to the same axis.



**a.** **b.** **c.**

*Figure II.10 Silhouette intersection.*

Even if this method is very simple and the computational complexity is very low, the SI approach is not robust against small variations of the shape (*e.g.,* in Figure II.10b the two silhouettes are very similar but because the position of the wings is slightly different, the corresponding distance is very large). Another drawback of the SI algorithm is the strongly dependence of the results on the PCA alignment.

In [Vranic04], authors propose the so-called *Enhanced silhouette based approach* (ESA), which exploits the three views corresponding to the projection onto the principal planes. The contour of each silhouette is extracted and sampled using a uniform angular distribution (*i.e.,* the polar angles of adjacent selected points differ by a constant) with respect to the centre of the silhouette. The sampled contour $c(t)$ is decomposed using Fourier series:

$$u_n = \frac{1}{N}\sum_{t=0}^{N-1} c(t)e^{\frac{-j2\pi nt}{N}}, \qquad n = 0..N-1. \tag{II.22}$$

where:

– $c(t)$ stores the distance from the centre of the model to the $t^{th}$ sample.

The feature vector of the 3D model contains the absolute values of the decomposition coefficient for each contour. Since samples of the contour are defined with respect to the centre of the 2D object, the Fourier descriptor is invariant to rotation and translation.

A different, single view approach is presented in [Liu09]. Authors propose to exploit a unique projection onto the principal plane of maximal eigenvalue. The projection is described using two descriptors: region-based Zernike Moments [Mukundan98] and contour-based Fourier descriptors [Zahn72].

The first descriptor employs the decomposition of the support region function $f(\rho,\theta)$ onto the basis of Zernike moments. In order to ensure rotation invariance, the feature vector contains only the absolute values of the decomposition coefficients. Section III.2.2.3 offers a detailed description of Zernike moments computation.

The second descriptor exploits the uniform sampled contour of the silhouette, expressed as a weighted sum of Fourier functions (Equation II.22). The decomposition coefficients $\{u_n\}$ are used as feature vector. The descriptor achieves, through the Fourier transform, invariance to translation and rotation. The scale invariance is obtained by dividing each coefficient by the continous component $u_0$.

Finally, let us mention the approach proposed in [Shih07]. A voxelized, volumetric representation is determined prior to performing the PCA. The viewing angle directions used to project the model correspond to the three principal axes. Each of the obtained images is decomposed into *L=60* concentric circles defined around the object's gravity centre (Figure II.11). The feature vector associated to each projection stores for each circle the number of pixels representing the object. The so-called *principal plane descriptor* (PPD) proposed is defined as the set of all three feature vectors. The scale invariance is obtained by dividing the feature vector by the total number of valid pixel present in the three projections. Let us note that, because of the concentric circular regions involved, the PPD is intrinsically invariant under 2D rotation.


*Figure II.11 Principle of the PPD approach.*
*a. the 3D model; b.-d. the object's projections onto the principal planes;*
*e. concentric circles used to determine the descriptor.*

However, the method implicitly assumes an ordering of the three principal directions based on the corresponding eigenvalues. Such an approach may lead, in the general case, to miss-alignments, as shown in [Zaharia01, Tangelder04]. This problem is illustrated in Figure II.12.

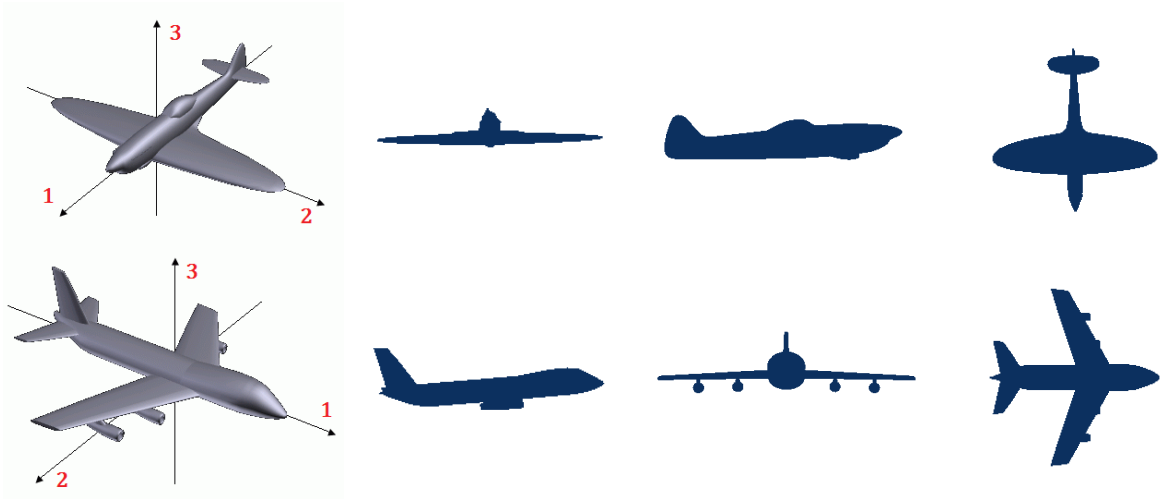



*Figure II.12 PCA miss-alignment.*
*a. The directions of the principal axes are different for two similar models.*
*b.-d. Couples of miss-matched images.*

PCA-based methods offer the advantage of obtaining a representation associated with a canonical, object-dependent coordinate system that partially solves the 3D transform invariance issues. However, the principal axes may present strong variations when dealing with similar model (Section II.1.1.2). Furthermore, a second limitation is related to the eventual miss-alignments that might occur. The reliability of the PCA is a key factor within this process that should be taken into account appropriately.

In order to overcome such limitations, a second family of methods, described in the next section, proposes to perform the 3D/2D projection according to a set of dense and evenly distributed viewing angles.

### II.2.2. Methods using evenly distributed viewing angles

Instead of computing preferential projection planes, this second family of approaches uses a set of dense and evenly distributed viewing angles.

In [Wen08] an extension of the SI algorithm [Napoléon07], so-called *Enhanced Silhouette Intersection* (ESI), is introduced. Instead of exploiting the PCA directions, the vertices of a regular dodecahedron are used to generate ten views, acquired after object normalization in translation, scale and rotation. As in the case of the SI method, the distance between two images is given by the number of pixels included in the symmetric differences between the corresponding support regions. However, when computing the global distance between two 3D models, instead of summing up the distances between similar views, a weighted sum is proposed. This choice is based on the simple assumption that the relevance of a projection is proportional to the root square of its area. Both the descriptor extraction procedure and the dissimilarity measure are fast to compute. The experimental evaluation proposed in [Wen08] shows that, compared with the SI algorithm, the ESI algorithm provides superior retrieval results. However, articulated object

matching is not supported and even small variations of the shape can strongly influence the retrieval results.

In [Chen03], authors introduce the *LightField Descriptor* (LFD) which encodes ten silhouettes of the 3D object obtained by projection from the vertices of the dodecahedron. Translation and scaling invariance of the image are extrinsically achieved by normalizing the size of the projection images. Furthermore, the silhouettes are described by both Zernike moments [Mukundan98] (III.2.2.3) and Fourier descriptor [Zahn72]. A number of 35 coefficients of Zernike moments are used as well as 10 coefficients for the Fourier descriptor. Thus, the resulting descriptor includes 45 coefficients for each projection image, and 450 for each LFD associated with a 3D model.

To compare two LFDs, the similarity measure used is the $L_1$ distance between the descriptor's coefficients. The minimum sum of the distances between all possible permutations of views provides the dissimilarity between the 3D models. Let us note that in the case of LFD there are 60 possible permutations and for each of them 10 individual distances between pairs of images need to be computed. In addition, as one LFD is not totally invariant under rotation, a set of 10 LFDs per model is used to improve the robustness. This leads to a total number of 5460 comparisons to be computed.

The need for multiple matches for each two objects makes LFD very time consuming. In order to reduce the computational cost, a multi-step fitting approach is adopted. In the first stage a reduced number of images per model and of coefficients is used in order to filter the results and retain a reduced number of candidate models. This procedure allows the early rejection of non-relevant models. The results obtained show that this algorithm outperforms most 3D shape descriptors at the cost of a significantly increased computational complexity.

A modified version of the LFD method is proposed in [Yang08]. Authors start from the observation that two different objects can have similar projections (Figure II.13), under the assumption that the scaling normalization is performed upon the silhouette images.

**a.** **b.** **c.**

*Figure II.13 Dissimilar objects presenting similar views after scale normalization.*

The Modified LFD (MLFD) approach proposes to skip the resizing step of the original LFD algorithm. Tested on the Princeton Shape Database [Shilane04], MLFD slightly outperformed the LFD performance (the Nearest Neighbour measure increases by 5.3%, the First and Second Tier measures by 4.1%, respectively 3.6% and the Discounted Cumulative Gain increases by 2.3%). This shows that, for 3D model retrieval purpose, the normalization issues needed for achieving invariance should be considered in the 3D space rather than in the domain of 2D projections and/or descriptors.

In [Daras09] the *Compact Multi-View Descriptor* (CMVD) is proposed. The authors tested the CMVD on both binary and depth images. The descriptor extraction starts with the normalization stage, which includes translation, rotation and scaling. In order to compute the principal axes, both PCA and VCA [Pu05] are performed. A number of 18 projections are obtained by placing the camera on the vertices of a 32-hedron and each of them is described by three sets of coefficients. First, 78 coefficients of the 2D Polar-Fourier Transform are computed. The usage of polar coordinates ensures rotation invariance. Secondly, 2D Zernike moments up to the 12th order are obtained, resulting in 56 coefficients. Finally, they consider the 78 coefficients, corresponding to 2D Krawtchouk moments $Q_{nm}$ [Yap03]:

$$Q_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \bar{K}_n(x; p_1, N-1) \bar{K}_m(y; p_2, M-1)\, f(x,y), \qquad (II.23)$$

where:

- $f(x,y)$ is the support region function of the 2D silhouette;
- $N \times M$ is the size of the image;
- $p_1, p_2 \in (0,1)$.
- $\bar{K}_n(x; p_1, N-1)$ is the $n^{th}$ order weighted Krawtchouk polynomial, defined as:

$$\bar{K}_n(x; p, N-1) = \sqrt{\frac{w(x; p, N-1)}{\rho(n; p, N-1)}}\ {}_2F_1\left(-n, -x; -(N-1); \frac{1}{p}\right), \qquad (II.24)$$

and

$${}_2F_1\left(-n, -x; -(N-1); \frac{1}{p}\right) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k z^k}{(c)_k k!}. \qquad (II.25)$$

When comparing two projection images, the $L_1$ norm is used to compute the distance between two descriptor vectors. Authors also take into account the fact that in some cases the first principal axis may not be successfully selected among the three principal axes. In order to deal with such an issue, $3 \times 8 = 24$ different alignments are considered. The total dissimilarity between two sets of images is obtained by summing up the dissimilarities between corresponding images. The distance between two models is the minimum distance that results when comparing the 18 projections of the first model with each of the 24 sets of images of the second model. In terms of computational complexity, the view generation process is the most time consuming. The 2D rotation invariance is ensured by the considered image descriptors. Experiments conducted on

several 3D model databases have shown that CMVD performs similarly to LFD while offering a reduced computational complexity.

In general, the methods that use evenly distributed viewing angles generate a higher number of projections (*e.g.,* 100 views for LFD, 18 views for CMVD) than those based on PCA analysis (between 3 and 9 views). Obviously, a larger number of images will carry more information, which results in a more complete description. However, the main limitation of such approaches remains their high computational complexity.

In order to overcome this drawback, some authors propose to reduce the number of views generated by the evenly distribution of the camera. Thereby, a subset of representative views is selected in order to reduce the computation complexity and the possible redundancy within the initial set.

### II.2.3. Methods using representative views

This family of approaches proposes to perform an intelligent selection of the views associated to each 3D model in order to reduce their number. The selection process supposes to cluster the silhouettes according to their 2D similarity. Thus, a reduced number of representative projections is obtained, that can be used to appropriately describe the 3D object.

The representative views selection problem can be formalized as follows: given a set $P = \{P_i(M)\}, i = 1..N$ of 2D views of a 3D model $M$ and an associated similarity distance $d: P \times P \rightarrow \mathbb{R}$, determine the subset $\{P_{Ri}(M)\} = P_R \subset P$ that maximize $d(P_{Ri}, P_{Rj}),\ \forall P_{Ri}, P_{Rj} \in P_R$ while minimizing $d(P_i, P_R) = \min_j d(P_i, P_{Rj})\ \forall P_i \in P \backslash P_R$. In other words, the selected representative views have to be as different as possible while any other views have to be close to one of the representative views.

A representative views selection algorithm is proposed in [Cyr01], where the authors present a method based on a similarity *aspect graph*. First introduced in [Koenderink76], the aspect graph represents a set of representative object projections and their spatial connection. In [Cyr01], authors propose to place the camera in the third principal plane, around the principal axis. A number of 36 silhouettes are obtained by uniformly sampling the (0, 180°) interval and the similarity is computed between each two projections. The *aspects* are defined as groups of similar silhouettes with respect to the considered similarity measure. They are obtained with the help of a clustering algorithm that maximizes the intra-class similarity while minimizing the inter-class similarity. A prototype view is determined for each aspect. Finally, the prototype views are represented as graph structure; each node corresponds to a stable view and each two adjacent stable views are connected by an edge of the graph. The similarity metric used for prototype selection employs the shock graph representation proposed in [Sebastian01]. The authors integrate the 2D/3D indexing method in a 3D model recognition framework. Only one projection image is used as query for each model, and compared to all the prototypes in the database. The number of comparisons is thus proportionally with the number of prototypes per model. The object is

matched against the one in the database having the most similar prototypes to the query. The same similarity measure is used to compare prototypes as the one exploited for their selection.

The main drawback of this method is the computational complexity of both prototype selection and similarity measure used for retrieval purposes. Also, since the viewing angles lie in a unique plane, the selection of this plane has to generate a robust response in order to ensure a 3D pose invariant behaviour.

[Denton04a] continued to exploit the aspect graph concept and proposed the *canonical set* selection algorithm. The viewing angle selection strategy is similar to the one adopted in [Cyr01] (*i.e.,* views acquired along a circle within the third principal plane), but the number of projections is increased to 180 for each 3D model. The novelty of their method consists in the selection of the representative views that compose the so-called *canonical set*. The set of projections $P = \{P_i(M)\}$ is modelled as a weighted graph where each node $p_i \in V$ corresponds to a projection and the weight $w_{ij}$ between two adjacent nodes is equal to the similarity between corresponding views:

$$w_{ij} = d(P_i, P_j). \qquad (II.26)$$

The graph contains an edge $(p_i, p_j)$ between each two nodes if the potential weight $w_{ij}$ is larger than a given threshold σ. The selection of the canonical set issue can be then formulated as determining the smallest subset $V^* \subset V$ such as for any node $p_i \in V \backslash V^*$ there exists an edge $(p_i, p_j)$ with $p_j \in V^*$ and which maximizes the weight $d(V^*)$ of the cut $(V^*, V \backslash V^*)$. In order to solve this multi-objective optimization problem, authors employ semi-definite programming [Alizadeh95].

The *canonical set* method was further extended in [Denton04b] to *bounded canonical set* (BCS). Here, the authors propose to limit the size of the canonical set to 2 or 3 views, thus ensuring a compact description of the 3D model.

In [Yamauchi06] a similar method, so-called *Stable View*, is proposed. The approach aims at selecting the viewing angles based on the degree of representativity of the corresponding projection images. Here, 162 silhouette images are rendered according to a uniform viewing angle distribution, defined over the unit sphere as a spherical triangular mesh. The obtained images are described using Zernike moments [Mukundan98] (III.2.2.3), up to the 15$^{th}$ order. The similarity between each two adjacent projections is computed using the $L_2$ norm distance. Then, a spherical weighted graph is constructed using the viewing angles as vertices. The weight of each edge is equal to the similarity between the views connected by the considered edge. Stable view regions represent sub-parts of the graph with similar corresponding projections (*i.e.*, sets of edges with low weights). Two stable view regions are separated by so-called *heavy edges*. Thus, based on the edge weights, the graph is partitioned into eight sub-graphs representing the stable view regions. Furthermore, a representative viewpoint needs to be found for each stable region. In order to achieve this goal, a pertinence value is assigned to each viewpoint. This value is based on the mesh saliency, a measure that evaluates the mesh curvature evolution when smoothed at different filter scales. The saliency measure is also used to sort the views according to the amount of information they carry. The algorithm is complex because of the weighted graph construction

procedure, which is the most time consuming stage (162 vertices of 6 adjacent edges each which generate 486 edges and as many weights to be computed). However, the approach proves to be effective for determining representative viewpoints.

The graph-based selection of representative views takes into account the position of the viewing angles around the model. Thus, a selected view can represent only its neighbours. However, views that are distant in the 3D virtual space can be very similar in the feature vector space (for example in the case of objects presenting symmetries). Therefore, it may be more useful to select views that represent their neighbours in the feature vector space and also to allow adapting their number to the geometry of the model.

In [Ansary07], authors propose to employ clustering techniques for representative views selection. After the normalisation of the 3D model in size, position and orientation, their algorithm generates a set of 320 views, equally spaced on the unit sphere. Each view is further described by the first 49 coefficients of the Zernike Moments [Mukundan98] (III.2.2.3). Thus, the views can be represented in a 49 dimensional space and grouped with the help of a k-means clustering algorithm [Silverman02].

However, the goal of [Ansary07] is to provide an algorithm able to adapt the number of representative views to the geometry of the 3D model. Therefore, they propose to use a Bayesian Information Criteria (BIC) [Schwarz79] in order to obtain an Adaptive Views Clustering (AVC).

The algorithm starts with all views in a single cluster. Further, an iterative process is applied, where at each step the existing clusters are divided in two sub-clusters. The BIC is computed for the initial cluster and for the corresponding sub-cluster and the configuration giving the maximum score is retained. The process continues until the maximum number of accepted clusters (*i.e.*, 40) is reached. The global BIC score is computed for each intermediate configuration and the best one is finally considered.

The authors also make the assumption that not all views have the same importance. Thus, a probability is associated to each representative view depending on the size of the corresponding cluster. Moreover, the probability to observe a model is computed based on the number of representative views associated to that object and on the total number of representative views in the database. Thus, the distance between a query and a model of the database can be expressed by the conditional probability of apparition of that model, knowing the query. Their experimental evaluation proves that, for 3D model retrieval purposes, view-based indexing methods (*i.e.*, AVC and LFD) outperform global 3D shape descriptors. Also, the results prove a better effectiveness of the probabilistic measure compared to the Euclidian distance. However, AVC does not reach retrieval scores as good as LFD.

The clustering selection in the feature vector space is also exploited in [Gao11, Gao12a] where an *Agglomerative Hierarchical Clustering* (AHC) approach [Cormack71, Fernandez08] was adopted for representative view selection. The AHC is a bottom-up clustering method where each cluster has sub-clusters, which in turn have sub-clusters until obtaining clusters with only one element. This hierarchical structure is obtained by starting with every single element in a cluster. Then, an iterative fusioning process is applied, where the most similar clusters (with

respect to a given similarity metric) are successively merged until obtaining a unique cluster (or the suited number $N_{RV}$ of representative views). The projection with the minimal distance compared with all other views in the same cluster is selected as the representative view of that cluster. [Gao12a] propose to select $N_{RV}$=10 projections among 60 or 41 evenly distributed views. The similarity is given by the Euclidian distance between the first 49 coefficients of Zernike Moments [Mukundan98, Mukundan08] (*cf.* Section III.2.2.3).

Moreover, the authors propose to also analyse the set of projections that describe the 3D model query. After the AHC, the representative views associated to the query are treated as sub-queries of different relevance (formally expressed by a weighting factor). The authors make the underlying assumption that the sub-queries that are close to the top retrieved models should be assigned higher weights. Therefore, a k-partite graph [Long06] is constructed. Each part of the graph corresponds to a 3D model and each vertex to a representative view. Initially, the weight of each representative view is proportional to the size of the corresponding cluster. Then, a k-partite graph reinforcement approach is performed in order to update the weights of the sub-queries. The experimental evaluation proposed in [Gao12a] proves that by weighting the sub-queries the 3D model, the retrieval performances are slightly increased.

In [Gao12b], the graph representation of the representative views is extended to hyper-graphs. In the case of hyper-graphs, an edge can connect three or more vertices, which allows creating more complex and realistic structures. All the views of all the objects in the database are grouped into clusters. Further, each cluster is considered as an edge that connects objects with similar views. The weight of an edge depends on the mean similarity between two elements of the cluster. By varying the number of clusters, multiple hyper-graphs can be generated. The hyper-graphs are merged together and the retrieval is performed on the fusion of the hyper-graphs. A training stage is first required in order to determine the weight of each hyper-graph in the fusion. Finally, the relevance score matrix is obtained as the minimum solution of the graph regularization problem.

The experimental evaluation proposed in [Gao12b] proves that the hyper-graph algorithm outperforms AVC and CMVD indexing methods.

In general, the representative view selection leads to greater complexity of the 2D/3D indexing process. For 2D object recognition purpose, the extraction complexity is not a drawback, since this stage is performed offline. However, in the case of 3D model retrieval, the query has to be described in a fast way and thus the computational complexity penalizes the methods employing representative views.

### II.2.4. Conclusions

Table II.1 synthesizes the various descriptors presented in this section. For each method, the extraction and the matching complexities, respectively denoted by $C_E$ and $C_M$, are qualitatively estimated (*i.e.,* + for low complexity and +++ for high). The numbers of views per model as well

as the viewing angle selection procedure are also indicated. The last column recalls the 2D descriptors associated to the projection images.

PCA-based methods present low complexity for both extraction and matching. They also offer the advantage of providing an object-dependent representation. However, as the principal axes may present strong variations, the reliability of the PCA remains a key factor to be solved.

The methods employing a dense repartition of the cameras (*e.g.*, LFD, CMVD) attempt to overcome the PCA-related limitations and provide best retrieval performances [Ansary07]. Yet, the price to pay is the cost of large descriptors and complex matching strategies.

The descriptor size can be reduced by using a clever selection of views. However, the representative view selection leads to higher extraction costs, which can be very penalizing especially for 3D model retrieval purposes.

The literature shows a wide palette of useful approaches. However, when comparing the different 2D/3D indexing methods it is difficult to evaluate objectively in what measure the choices involved at each stage of the retrieval process (*i.e.*, normalization, viewing angle selection, 2D shape description, matching strategy...) affects the retrieval performances. This issue is treated in detail in the following chapter. The power of 2D/3D indexing approaches is here first investigated for 3D model retrieval purpose. This will give us a first qualitative evaluation of the adopted elements. Then, in Chapter IV, the view-based 3D model indexing will be integrated in the 2D shape classification framework.

*Table II.1 Overview of 2D/3D indexing approaches*

| Method | $C_E$ | $C_M$ | No of Views | Viewing angle selection | 2D descriptor |
|---|---|---|---|---|---|
| **MPEG-7 2D/3D ART** [ISO/IEC03] | ++ | ++ | 7 | PCA-based | Angular Radial Transform |
| **MPEG-7 2D/3D CSS** [ISO/IEC03] | + | ++ | 7 | PCA-based | Curvature scale space Descriptor |
| **M-Tree CSS** [Mahmoudi07] | ++ | ++ | 7 | PCA-based | Curvature scale space Descriptor |
| **MCC** [Napoléon07] | ++ | +++ | 3 – 9 | PCA-based | Curve evolution at Gaussian filtering |
| **SI** [Napoléon07] | + | + | 3 | PCA-based | Binary images |
| **ESA**[Vranic04] | + | + | 3 | PCA-based | Fourier descriptor |
| **PPA** [Liu09] | + | + | 1 | PCA-based | Zernike moments & contour-based Fourier descriptor |
| **PPD** [Shih07] | + | + | 3 | PCA-based | Sums of pixels within concentric circles |
| **ESI** [Wen08] | + | + | 10 | Even distribution | Binary images |
| **LFD** [Chen03] | +++ | +++ | 100 | Even distribution | Zernike moments & Fourier descriptor |
| **MLFD**[Yang08] | +++ | +++ | 100 | Even distribution | Zernike moments & Fourier descriptor |
| **CMVD** [Daras09] | ++ | ++ | 18 | Even distribution | Zernike moments, Polar Fourier and Krawtchouk moments coefficients |
| **Aspect graph** [Cyr01] | +++ | +++ | 5 – 10 | Aspect graph prototypes | Shock graph |
| **Canonical Set** [ Denton04a] | +++ | + | N/A | Canonical set | Shock graph |
| **BCS** [Denton04a] | +++ | +++ | 2 – 3 | Bounded canonical set | Shock graph |
| **Stable views** [Yamauchi06] | +++ | N/A | 8 | Spherical graph stable views | Zernike moments coefficients (up to order 15) |
| **AVC** [Ansary07] | +++ | ++ | 1 – 40 | Adaptive clustering | Zernike moments |
| **AHC** [Ga012] | +++ | +++ | 10 | Agglomerative clustering | Zernike moments |
| **Hyper-graph** [Gao12b] | +++ | +++ | 20/41 | Hyper-graph structure | Zernike moments |

*$C_E$ – Descriptor's extraction complexity, $C_M$ – Matching complexity.*

# III. VIEW-BASED 3D MODEL RETRIEVAL

**Abstract.** *In this chapter we tackle the issue of view-based 3D model retrieval. The objective here is twofold. On one hand, we investigate the impact of various 2D shape descriptors and viewing angle strategies upon the 3D model retrieval performances. On the other hand, we propose an analysis of existing 3D model repositories considered in our work, in terms of intra and inter-class variability/separability, which can be useful under the perspective of 2D/3D semantic categorization.*

*The various 2D shape descriptors adopted are presented and discussed. In addition, a novel contour-based shape descriptor, so-called Angular Histogram (AH) is proposed. The retained viewing angle selection strategies are also detailed and analyzed. A clustering-based approach for the adaptive selection of representative views is also introduced here.*

*Our experiments first concern the analysis of 3D model repositories. The MPEG-7 and Princeton data sets are here considered and analyzed with the help of a set of objective criteria. The study involves the various descriptors and viewing angle selection strategies retained.*

*Finally, a comparative and objective evaluation of the various descriptors and viewing angle selection techniques for 3D model retrieval purposes is proposed.*

**Keywords:** *shape descriptor, 3D meshes; 2D/3D indexing, multiview matching, 3D model database, MPEG-7 standard; projection strategy.*

## III.1. INTRODUCTION

One of the main applications of 2D/3D indexing framework concerns, naturally, the field of 3D model retrieval. The objective here is to identify/retrieve pertinent 3D models in a given repository. The queries can be formulated in several ways:

- by example: a 3D model, similar to those suited by the user, is given as example,
- by 2D projections/sketches: a set of 2D views (real images or sketches) representing different perspectives of the suited model are provided at the input;
- by 3D sketches: the user models itself a 3D sketch that represents the query. Such a method requires from the user advanced 3D modelling skills and dedicated modelling tools, which is penalizing for general-purpose applications,
- by text: the query is based on text keywords,
- multimodal queries, involving combinations of the above-mentioned individual queries.

In the current chapter, only queries by example will be considered, where the input is an example 3D model provided by the user. The example is further described with the help of view-based, 2D/3D descriptors and compared to all the objects in the database. The most similar models are determined, sorted by decreasing similarity order and proposed to the user as response to his query. A synoptic scheme of a 3D model retrieval system is illustrated in Figure III.1.

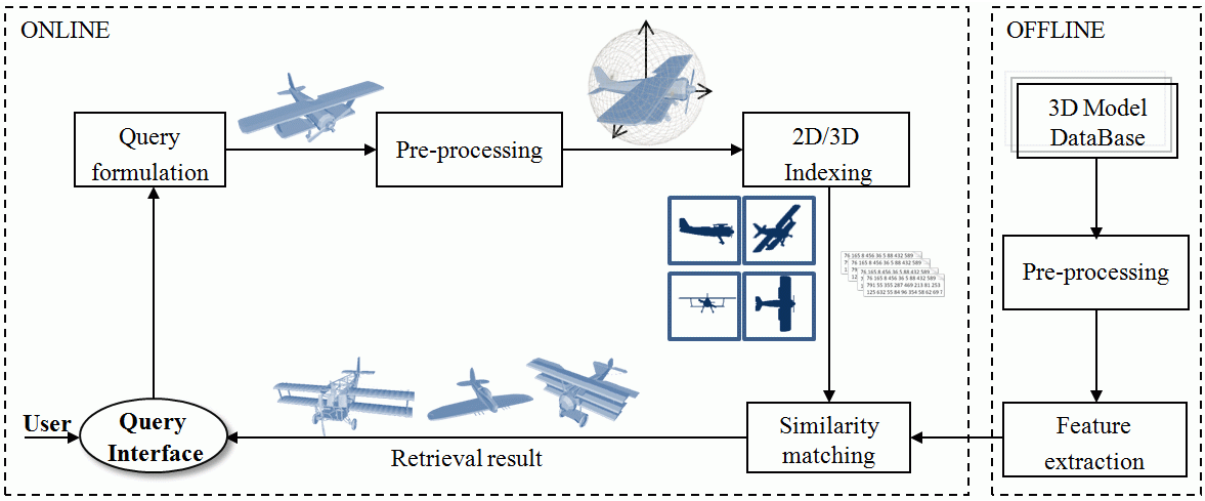



*Figure III.1 The 3D model retrieval scheme.*

The 2D/3D indexing of the database is performed only once, offline, and the resulting descriptors are stored and employed each time when a new query is formulated. Therefore, it is important to employ adequate descriptors which guarantee a limited storage size. On the other hand, the comparison of the example model with those on the database is done online and thus a speed requirement appears. Therefore, the similarity computation and matching has to be fast enough, especially when large 3D model databases are involved.

In this chapter, we address the issue of view-based 3D model retrieval. First, the adopted 2D/3D indexing methods are presented. Further, the 3D model retrieval framework is detailed. The first part of our experiments is dedicated to the analysis in term of intra and inter-class variability/separability of the 3D model repositories involved in our work. Next, we provide a comparative study of the retrieval performance of various 2D shape descriptors and viewing angle selection techniques retained. These experiments allow a first evaluation of the various 2D/3D indexing techniques employed further in Chapter IV for 2D object recognition purposes.

Let us start by presenting the 2D/3D indexing techniques adopted in our work.

## III.2. ADOPTED 2D/3D INDEXING METHOD

As discussed in Chapter II, the behaviour of the 2D/3D indexing methods is strongly influenced by the various choices involved in the 2D/3D processing chain. Within this framework, the adoption of appropriate viewing angles selection strategies and of discriminant 2D descriptors is determinant. Let us first detail the viewing angle selection strategies retained in our work.

### III.2.1. Viewing angle selection

As already underlined, the strategy adopted for projecting the 3D model represents an important phase in 2D/3D indexing. The resulting set of views has to be representative for the 3D model, to contain as more information as possible, while including a relatively small number of projections. As presented in Section II.1.2.2 there exist several families of viewing angle strategies, corresponding to various underlying hypotheses. In our work we have considered strategies from each family, in order to experimentally determine their influence on the results.

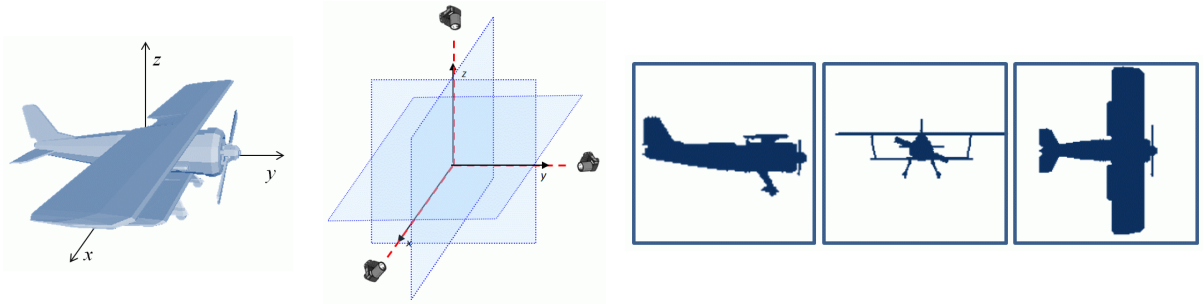
Concerning the pose normalization aspects (*cf.* Section II.1.1), the model is first translated such as its gravity centre coincides with the origin of the coordinate system (as described in Section II.1.1.1.2). Next, the PCA analysis is performed(*cf.* Section II.1.1.2) and the object is oriented such as its principal axes are aligned with those of the coordinate system. Finally, the model is resized with respect to the eigenvalues obtained through the PCA (*cf.* Section II.1.1.3.2). Let us recall that, in order to avoid redundancy within the set of views, only half of the bounding space is used for camera placement.

A first viewing angle selection strategy that we have adopted is based on PCA.

#### III.2.1.1. PCA-based viewing angle selection

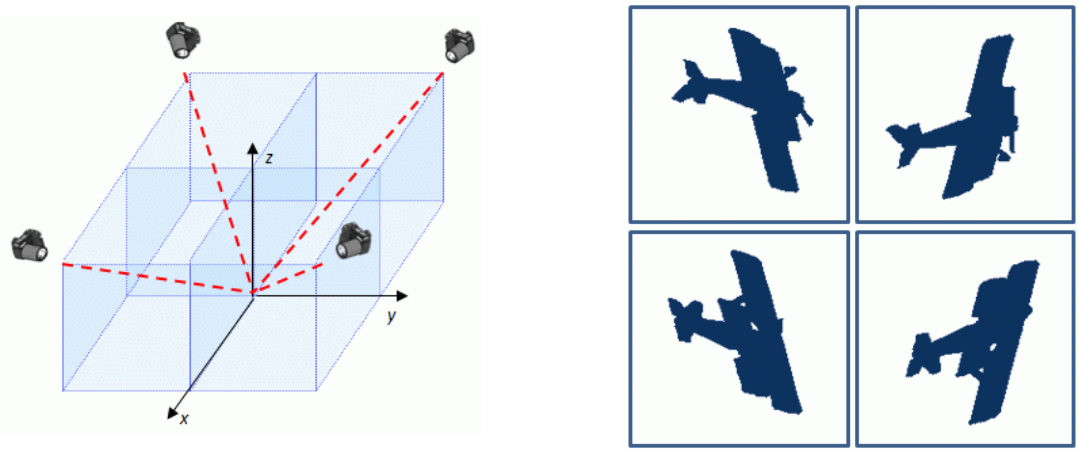
The PCA-based viewing angle selection strategy uses the principal components of the 3D model as viewing directions. The 3D model is projected onto the three principal planes, resulting in a set of views, so-called PCA3 (Figure III.2).

The principal planes describe eight octants, whose diagonal can be used as secondary viewing directions (Figure III.3). Thus, a set of seven views is obtained. From now on, this viewing angle selection strategy will be referred to as PCA7.




**a.** **b.** **c.**

*Figure III.2 PCA3 viewing angle selection strategy.*
*a. The aligned 3D model; b. the PCA3 viewing angles; c. the resulting projections.*




**a.** **b.**

*Figure III.3 Secondary views used by the PCA7 viewing angle selection strategy.*
*a. the secondary viewing angles; b. the resulting projections.*

Let us recall that the PCA3 and PCA7 strategies are those recommended by the ISO/MPEG-7 standard for the MultiViewDS [Bober02, ISO/IEC02].

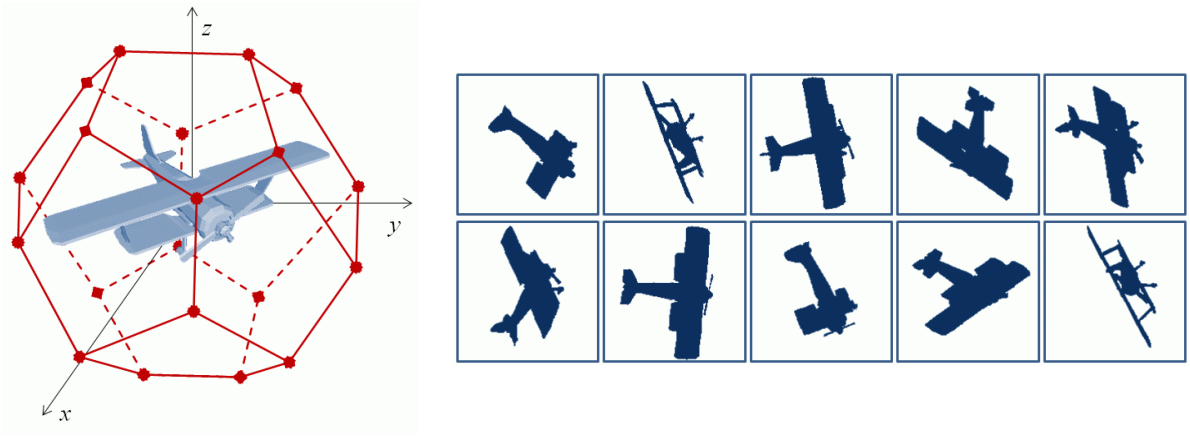
The second class of viewing angle selection strategies proposes to use a uniform repartition of the cameras around the 3D model.

### III.2.1.2. Uniform camera distribution

This viewing angle selection strategy is inspired by the one proposed to the LFD descriptor introduced in [Chen03]. A regular dodecahedron, placed around the 3D model, is considered. A number of 10 cameras are placed on the vertices of the dodecahedron and oriented towards the 3D model (*i.e.*, the origin of the considered coordinate system).

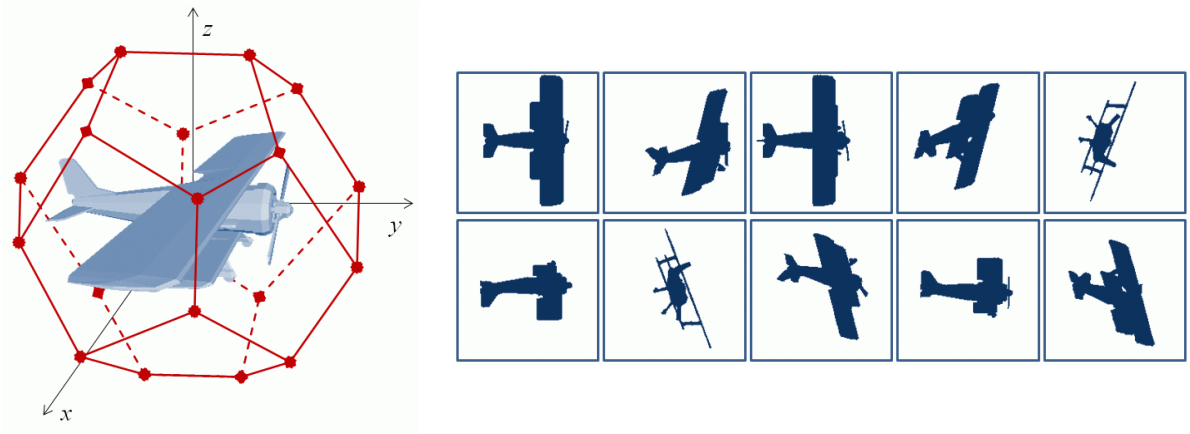
When using a uniform repartition of the camera, the 3D model can present an arbitrary orientation in the virtual space, as illustrated in Figure III.4a. However, for 3D-to-3D matching purposes, it is useful to achieve consequent sets of views (*i.e.,* two sets of views to contain similar perspectives of the associated 3D models).

Therefore, two sub-cases of dodecahedron-based cameras repartition have been considered. In the first one, the 3D model has a random orientation (Figure III.4), while in the second case, the 3D model is first aligned with the help of PCA (Figure III.5). From now on, this viewing angle selection strategies will be referred to as DODECA, respectively DDPCA.




**a.** **b.**

*Figure III.4 DODECA viewing angle selection strategy.*
*a. the viewing angles; b. the resulting projections.*




**a.** **b.**

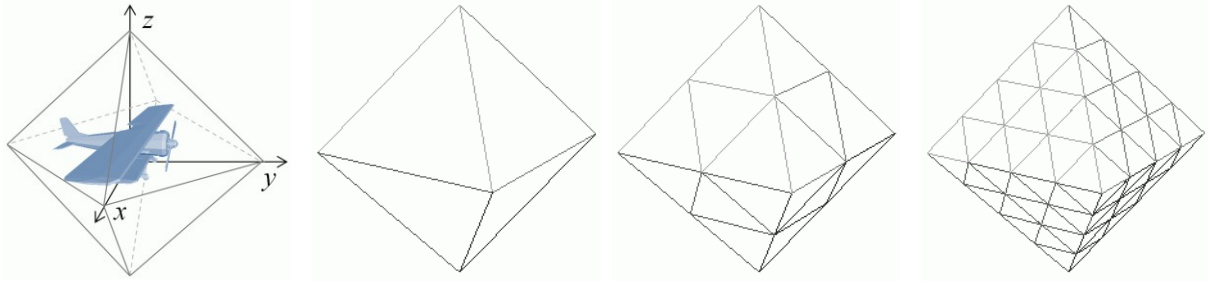
*Figure III.5 DDPCA viewing angle selection strategy.*
*a. the viewing angles; b. the resulting projections.*

Finally, we have considered an angle selection strategy based on representative views, described in the following section.

### III.2.1.3. Combined method

The third viewing angle selection strategy proposes to exploit in the same time the uniform repartition of the cameras and the PCA-based repartition. First, the 3D model is normalized in orientation, size and position such as its axes of inertia coincide with the coordinate system.

An octahedron, whose diagonals coincide with the axes of the coordinate system (Figure III.6a), is used for cameras placement. Positioning the cameras on the vertices of the octahedron results in a set of three views, corresponding to the projections on the principal planes (the same set obtained with the PCA3 projection strategy). The faces of the octahedron are further successively subdivided, resulting in a mesh with 18, respectively 66 vertices (Figure III.6 c and d). By placing cameras on the vertices of the subdivided octahedron meshes and orienting them towards the 3D model's centre, it results sets of 9, respectively 33 uniformly distributed views. From now on, the octahedron-based viewing angle selection strategies will be referred to as OCTA9, respectively OCTA33.

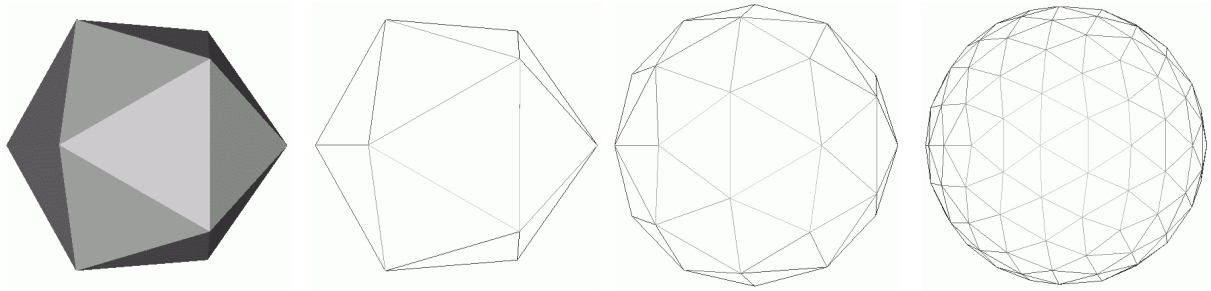



**a.** **b.** **c.** **d.**

*Figure III.6 Octahedron-based viewing angle selection strategy.*
*a. the 3D model; b. Octahedron wireframe representation (OCTA3); c. First subdivision level of the octahedron (OCTA9); d. Second subdivision level of the octahedron (OCTA33).*

### III.2.1.4. Representative views

In order to obtain a dense and uniform repartition of the viewing angles, we have exploited here the vertices of an icosahedron, whose faces were successively subdivided (Figure III.7).



**a.** **b.** **c.** **d.**

*Figure III.7 Icosahedron-based viewing angle selection strategy.*
*a. & b. Icosahedron; c. First level of subdivision of the icosahedron; d. second level of subdivision of the icosahedron.*

The 3D model, which presents an arbitrary orientation in the virtual space, is placed inside the subdivided icosahedron mesh. The cameras, placed on the vertices of the mesh and oriented toward the 3D model, generate a set of $N_P$=81 views.

Next, the objective is to determine a subset of $N_{RV}$ (with $N_{RV} < N_P$) representative views. For this purpose, an adapted k-means clustering approach [Silverman02] is employed. The aim of the clustering is to divide the $N_P$ views into $N_{RV}$ groups (*i.e.,* clusters) maximizing the intra-class similarity between views, while minimizing the inter-class similarity. A central view (centroid) is determined for each cluster. The set of representative views is defined as the set of all central views.

In order to evaluate the similarity between projections, a $N_D$-dimensional feature vector (*i.e.*, 2D shape descriptor) is associated with each view. A similarity measure (appropriate for each descriptor) is employed in order to compute the distance $d_{ij}$ between each two views $i$ and $j$. The adopted 2D shape descriptors, as well as the corresponding similarity measures, will be presented in Section III.2.2.

First, $N_{RV}$ views evenly distributed around the model are selected as initial centroids. Next, each of the available $N_P$ views is assigned to the cluster with the closest centroid with respect to the given similarity measure. A set of clusters is thus determined. Further, for each cluster, the view which presents the minimum mean distance to the other views within the same cluster is chosen as new centroid. The process is then iterated until convergence.

In order to homogenize the inter-cluster distances and thus to avoid the case where two clusters are close one to the other, while other clusters present distant views, an additional refinement stage is introduced. The minimum distance $d_{C\text{-}C}$ between two centroids and the maximum distance $d_{C\text{-}V}$ between a centroid and a view within the same cluster are computed. If there exist two clusters which are closer than the most distant view $d_{C\text{-}V}$, then that corresponding view will constitute a new cluster, while the two closest clusters are merged. This refinement test is performed after each re-calculation of the clusters.

The algorithm is formally expressed by the following pseudo-code:

0. Compute the distance matrix
1. Initiate the cluster centroids
2. Associate each view to the closest centroid
3. Compute the new centroid of each cluster
4. Test clusters
    4.1. Compute the maximum centroid-view distance ($d_{C\text{-}V}$)
    4.2. Compute the minimum centroid-centroid distance ($d_{C\text{-}C}$)
    4.3. if($d_{C\text{-}V} > d_{C\text{-}C}$)
        4.3.1. Join the two closest clusters
        4.3.2. Split the most spread cluster
        4.3.3. Compute the new centroids

Repeat steps 2-4 until convergence

In our work the number $N_{RV}$ of representative views was set to 6 and 10. From now on, the representative views selection methods will be refer to as RV6, respectively RV10. Figure III.8 illustrates the 81 views obtained by placing the camera on the vertices of the icosahedron and the views selected with RV10 strategy.

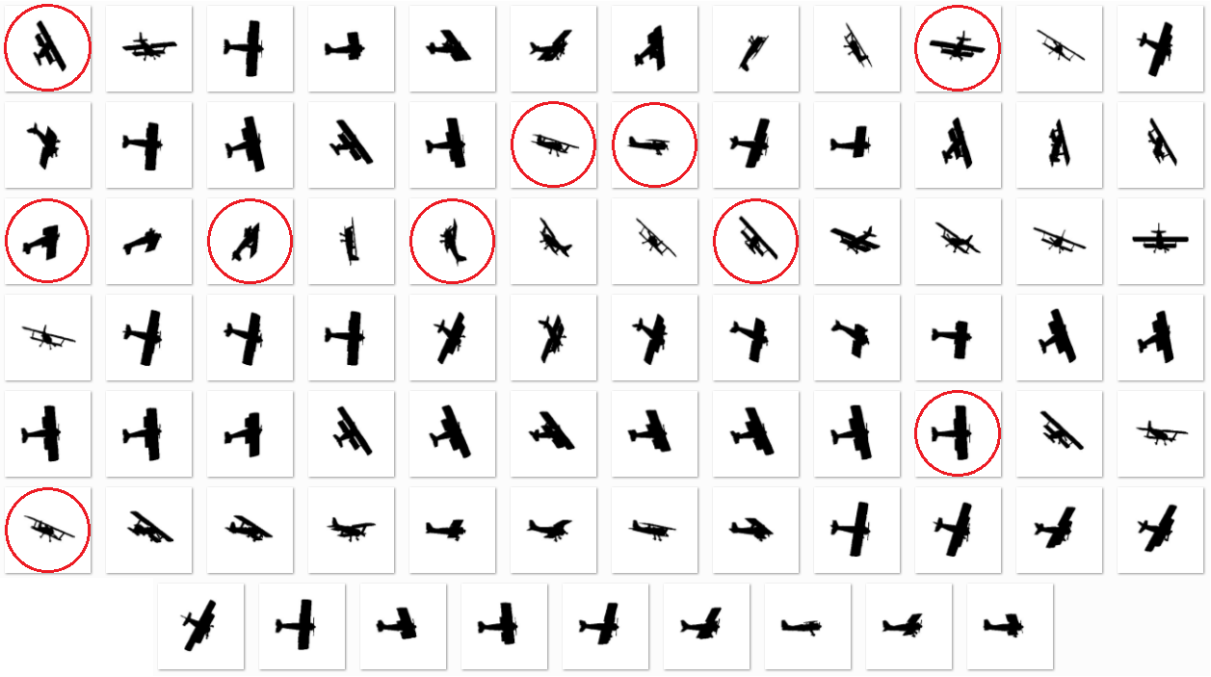


*Figure III.8 RV10 views selection strategy.*

With the help of the above-presented viewing angle selection strategies, a set of projections can be associated to a 3D model. In order to accomplish the 2D/3D indexing process, each view has to be characterized with the help of 2D shape descriptors. The next section present the 2D shape descriptors considered in our work.

### III.2.2. Retained 2D shape descriptors

For 2D/3D indexing purpose, the 2D shape descriptor employed has to satisfy some constraints, imposed by the large number of views that needs to be encoded and compared. Therefore, besides a good discrimination power, small storage size and fast similarity estimation represent very important criteria for 2D shape descriptor selection.

In our work we have adopted the following 2D shape descriptors, exploiting both region and contour information:

- **Region Shape (RS)** and **Contour Shape (CS)** descriptors, adopted by the MPEG-7 standard [ISO/IEC02, Bober02];
- **Zernike Moments (ZM)**, widely used for 2D/3D indexing purpose;
- **Hough Transform (HT)**, appropriate for describing shapes with arbitrary topologies;
- **Angular Histogram (AH)**, a new contour-based descriptor proposed in this thesis.

Let us now detail each of the retained descriptors.

### III.2.2.1. Region Shape

The RS descriptor, adopted within the MPEG-7 DS, employs the Angular Radial Transform (ART) [Kim99] in order to decompose the object's support function $f(\rho,\theta)$, expressed in polar coordinates $(\rho,\theta)$, within a base of radial functions $\{V_{mn}(\rho,\theta)\}$ defined as:

$$V_{mn}(\rho,\theta) = \frac{1}{\pi}\cos(\pi n\rho)e^{jm\theta}, \qquad (III.1)$$

where:
- $m \in \{0 \dots M-1\}$;
- $n \in \{0 \dots N-1\}$;
- $j = \sqrt{-1}$ is the imaginary unit;
- $\rho, \theta$ are the coordinates of the polar system.

The decomposition coefficients $\{c_{mn}\}$ are given by the equation:

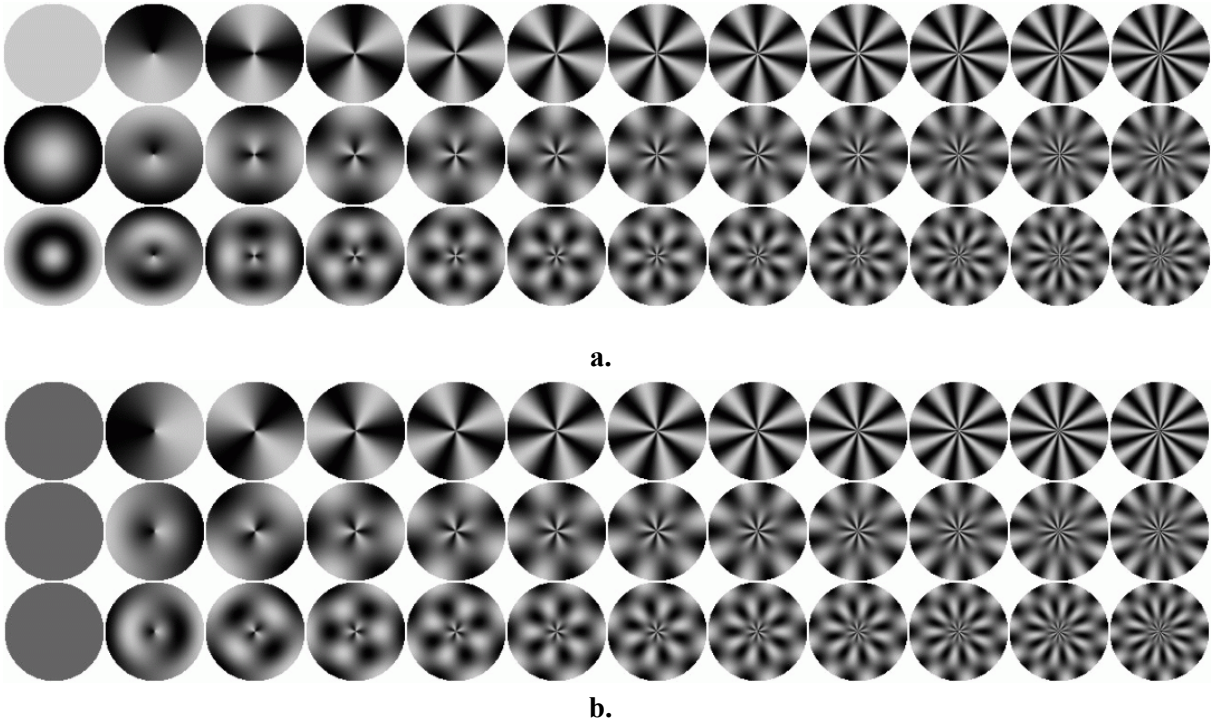
$$c_{mn} = \int_0^{2\pi}\int_0^{1} V_{mn}(\rho,\theta)f(\rho,\theta)d\rho d\theta. \qquad (III.2)$$

The RS descriptor is defined as the set of coefficients $\{c_{mn}\}$. In order to intrinsically achieve rotation invariance, only the absolute values $|c_{mn}|$ of the coefficients are used. The translation invariance is obtained by centring the 2D object with respect to its gravity centre. Finally, the scale invariance is achieved by resizing the shape to the unit disc.

The similarity measure simply consists of $L_1$ distance between ART coefficients. The 2D-ART is thus invariant under similarity transforms, and is suitable for shapes with arbitrary topologies, which can present holes or multiple connected components.

Concerning the parameters involved, we have adopted the values recommended by MPEG-7 (*i.e.*, $M = 12$ and $N = 3$), resulting in a base of 36 functions. As the first radial basis function $V_{00}(\rho,\theta)$ presents a constant value over the entire domain of definition, the first decomposition coefficient will represent the object's area. Therefore, this coefficient is discarded from the representation and the feature vector is limited to the other 35 coefficients. Let us also note that the decomposition coefficients take real values within the [0,1] interval. In order to reduce the storage size, the interval is divided into 16 non-uniform sub-intervals, numbered from 0 to 16, and for each real coefficient only the number of the corresponding interval is stored. When computing the similarity measure, each stored index is replaced by the central value of the corresponding sub-interval.

Figure III.9 a and b illustrates the real, respectively the imaginary parts of the Angular Radial Transform basis functions.



**a.**

**b.**

*Figure III.9 The Angular Radial Transform (ART) basis functions.*
*a. the real parts; b. the imaginary parts.*

### III.2.2.2. Hough Transform

The Hough Transform (HT) [Duda72] can be used as a region-based descriptor which expresses the 2D shape as a cumulative distribution. The HT of a 2D shape is the discrete version of the Radon Transform [Radon86], defined as:

$$g(s,\theta) = \int_{\mathbb{R}^2} f(x,y)\delta(x\cos\theta + y\sin\theta - s)dxdy, \qquad (III.3)$$

where:
- $f(x,y)$ is the object's support region function;
- $\delta$ is the Dirac distribution.

The HT employs the polar parameterization of straight lines in the $IR^2$ plane. Thus, each line in the Cartesian space is uniquely determined by $s$, the distance from the origin of the coordinate system to the line, and $\theta$, the slope of the line (Figure III.10a).

In order to obtain the HT of a point $(x_p, y_p)$ in the Cartesian space, a family of lines $\{(s_i, \theta_i)\}$ passing through that point is considered (Figure III.10b) and represented in the HT space $(s, \theta)$ (Figure III.10c). The HT of a 2D shape is obtained by accumulating the transforms of all points composing the object's support function.

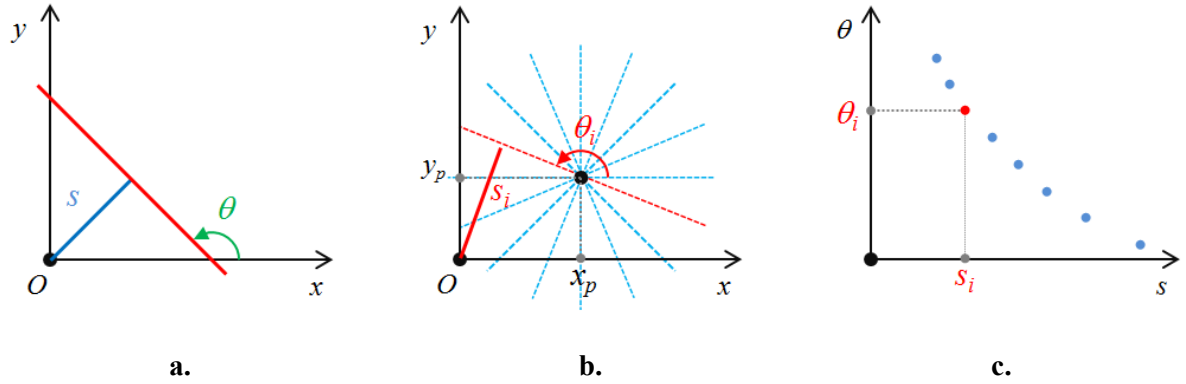



*Figure III.10 The Hough Transform.*
*a. polar coordinates of a straight line; b. the family of lines considered for HT;*
*c. HT representation of a family of lines associated to a Cartesian point.*

Exploiting both distance and angle measures, the HT is not invariant under similarity transformations (*i.e.,* translation, rotation, scaling or combinations of them). Therefore, before the computation of the HT, a PCA of the 2D shape is performed in order to determine the principal axes of the object and its intrinsic size. The 2D object is translated with its gravity centre in the origin of the coordinate system and rotated such that its principal axes coincide with those of the coordinate system. Finally, the object is scaled with respect to the square root of the summed eigenvalues.

In our work we have considered families of $N_\theta$=64 lines, uniformly sampling the set of possible orientations [0, 2π]. The distance parameter $s$ has been quantified uniformly to $N_s$=32 values. Thus, the HT representation contains $N_\theta \times N_s = 2048$ bins. The $L_1$ norm is used in order to compute the distance between two HT descriptors. Figure III.11 illustrates some examples of 2D shapes and the corresponding Hough Transforms.

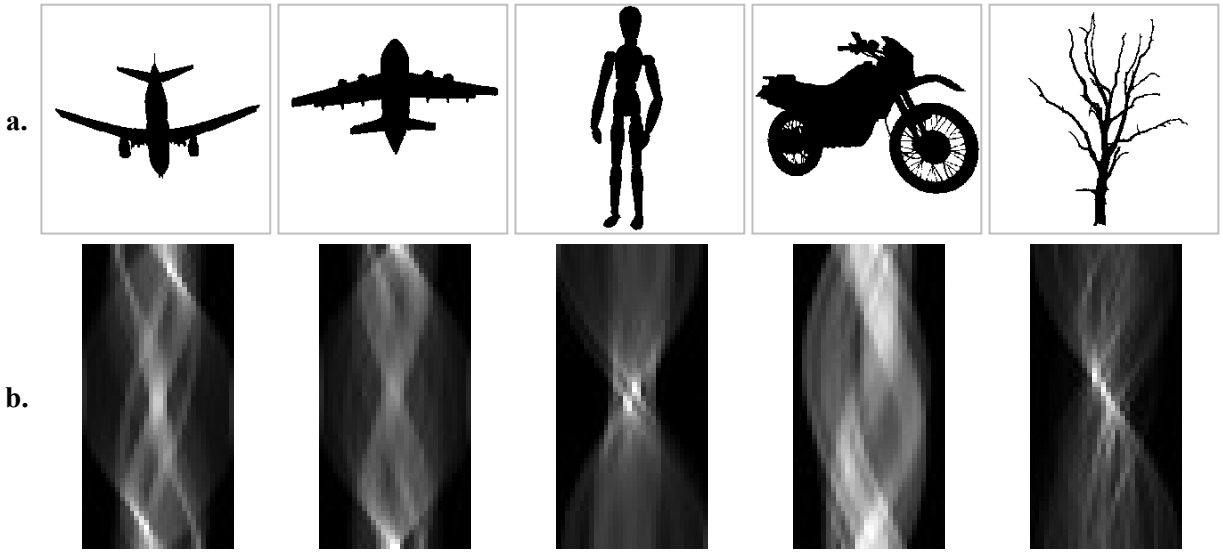



*Figure III.11 Examples of Hough Transforms.*
*a. 2D objects; b. corresponding Hough Transform images.*

### III.2.2.3. Zernike Moments

The last region-based descriptor retained is represented by Zernike Moments [Mukundan98]. This descriptor employs the decomposition of the support region function $f(\rho,\theta)$ on the basis of Zernike moments $V_{nm}^{*}(\rho,\theta)$ (Figure III.12):

$$V_{nm}^{*}(\rho,\theta) = e^{jm\theta} \sum_{s=0}^{(n-|m|)/2} (-1)^{s} \frac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!} \rho^{n-2s}, \qquad (III.4)$$

where:

- $\rho,\theta$ are the polar coordinates;
- $n$ represents the order and $m$ the repetition of a given function, with $n-|m| \vdots 2$ and $|m| \leq n$.

The decomposition coefficients $\{Z_{nm}(\rho,\theta)\}$ are given by the following equation:

$$Z_{nm}(\rho,\theta) = \frac{n+1}{\pi} \int_{0}^{2\pi} \int_{0}^{1} f(\rho,\theta)\, V_{nm}^{*}(\rho,\theta) d\rho d\theta. \qquad (III.5)$$

In order to ensure rotation invariance, only the absolute values of the decomposition coefficients are used for description. The invariance to other affine transformations is achieved in the same way as in the case of the RS descriptor (Section III.2.2.1).

In our work, we have considered Zernike Moments up to order 11 (*i.e.,* $n = 0..11$) which results in 42 functions. As the first component presents a constant value over the entire domain of definition, the corresponding coefficient is discarded from the representation, thus resulting in a 41 values feature vector. The similarity measure used in order to compare two shapes described by the Zernike Moments is simply the $L_1$ distance computed between the corresponding coefficients.

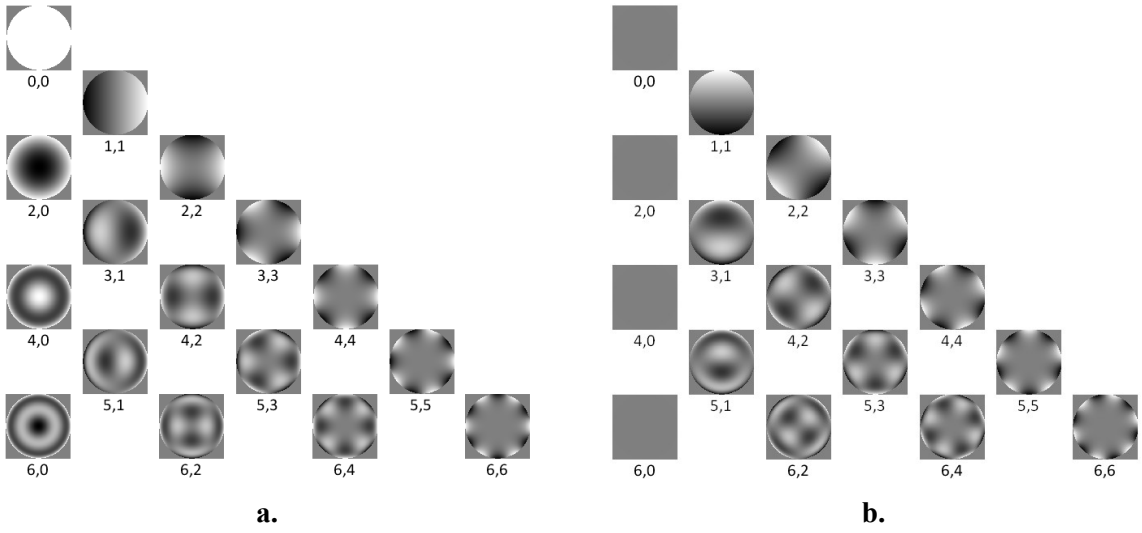



*Figure III.12 Zernike basis functions.*
*a.the real parts; b. the imaginary parts.*

### III.2.2.4. Contour Shape

The *Contour Shape* (CS) descriptor, adopted within the MPEG-7 DS, exploits the Contour Scale Space (CSS) representation proposed in [Mokhtarian92].

The first step in the CS descriptor computation is the extraction of the contour $c(n)$, with $n \in [0,1]$. The CSS representation is obtained by performing a multi-scale analysis of the contour. A bank of Gaussian filters $g(n,\sigma)$, with increasing standard deviation $\sigma$, is applied to $c(n)$, resulting in a set of contours $\{c(n,\sigma_i)\}$ (Figure III.13a):

$$c(n,\sigma_i) = c(n) * g(n,\sigma_i), \qquad (III.6)$$

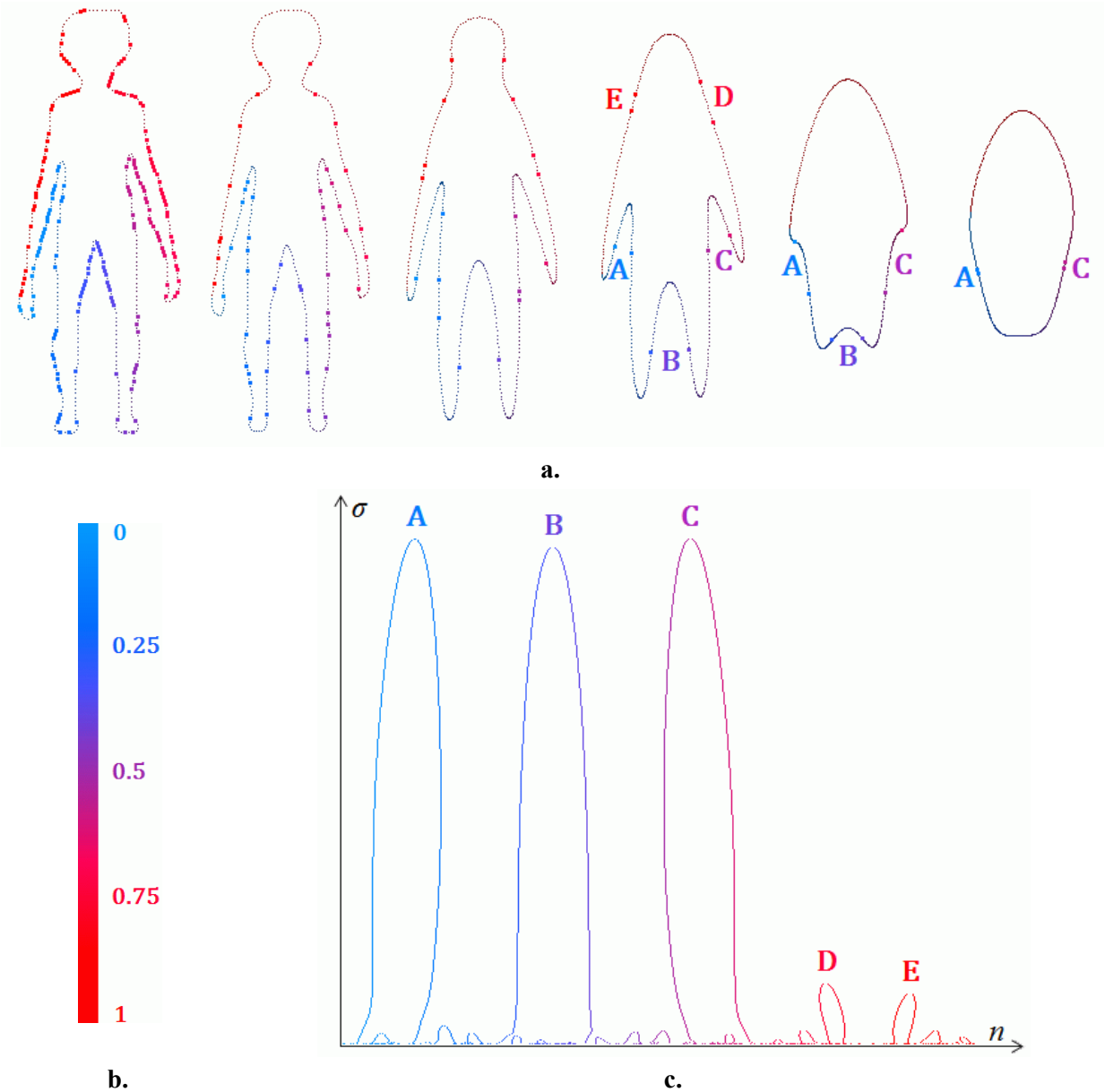



*Figure III.13 Contour Scale Space.*
*a. Contour filtering and inflexion points; b. colourbar (the colour indicates the curvilinear position of each sample point); c. CSS representation. The most important curvature peaks are marked with letters from A to E on both filtered contours and CSS representation.*

with:

$$g(n,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2}. \qquad (III.7)$$

Further, the inflexion points of each contour $c(n,\sigma_i)$ are computed and represented in the $(n,\sigma)$ space (Figure III.13c).

Finally the CS descriptor stores the predominant curvature peaks in the CSS representation (with the corresponding curvature values and associated curvilinear abscises). The resulting feature vector stores in total 26 values. The Earth Mover's Distance (Section II.1.4.1.2) is employed in order to measure the similarity between two contours described with the help of CSS representations. The CSS representation is invariant to translation and scaling. The rotation of the 2D shape leads to a cyclic permutation of the CSS representation.

### III.2.2.5. Angular Histogram

Finally, we propose a new descriptor, so-called *Angular Histogram* (AH). The shape contour is extracted and sub-sampled in a number $N_S$ of successive 2D points. Further, the angles defined by each three consecutive samples ($\alpha_i = \angle\big(c(i-1), c(i), c(i+1)\big)$) are computed and represented in a $N_\theta$-bins histogram. However, such a description encodes only the details of the contour. In order to offer in the same time local and global representation capabilities, several distances $\Delta n$ are considered between the three samples that define each angle ($\alpha_{i,n} = \angle\big(c(i-\Delta n), c(i), c(i+\Delta n)\big)$). Therefore, a total of $N_L$ histograms are computed, one for each value of $\Delta n$. The AH histogram is obtained by the concatenation of $N_L$ histograms of $N_\theta$ bins each.

As the samples of the contour take quantified values (due to the pixel representation), the angles $\alpha_i$ will also take quantified values. This behaviour can lead to errors in the computation of the histogram. In order to avoid such inconvenience, we introduce, before sampling, a contour filtering stage. Both $x$ and $y$ coordinates are low-pass filtered with a simple kernel K=[0.25 0.5 0.25].

Two AH descriptors are compared using the $L_1$ distance computed between corresponding AH coefficients. Let us note that the obtained descriptor is intrinsically invariant to similarity transformations.

In our work we have considered $N_\theta$=18 bins histograms for an 180° interval and $N_L$=5 different levels, resulting in a 90 integer values feature vector. Figure III.14 and Figure III.15 illustrate two shape contours and the associated angular histograms.

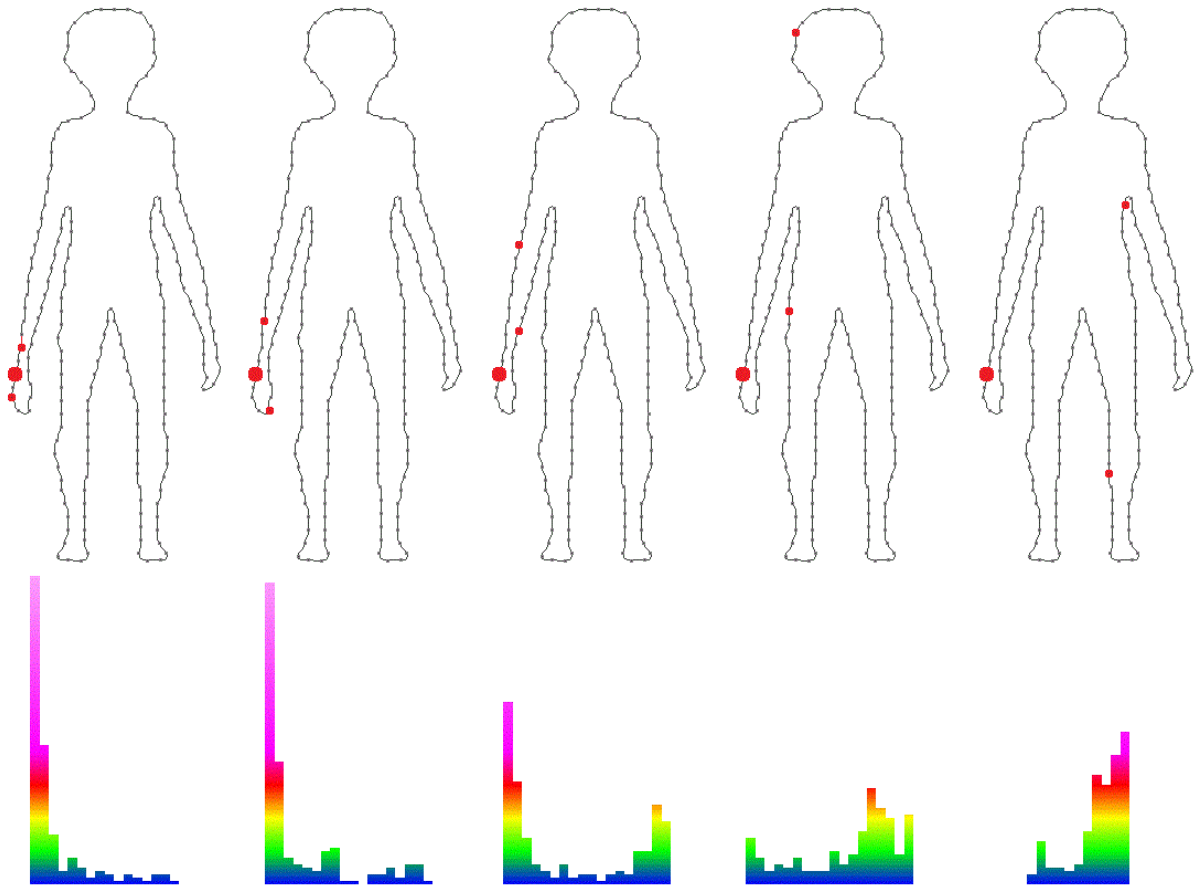


*Figure III.14 Angular Histogram — humanoid.*
*Each histogram corresponds to a different distance between the samples (red dots) used to compute the angles.*

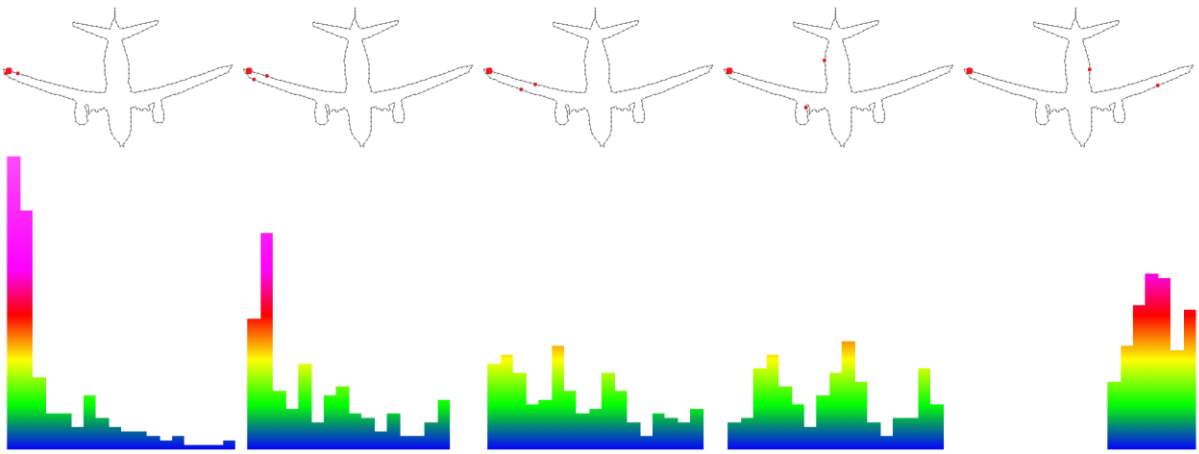


*Figure III.15 Angular Histogram — airplane.*
*Each histogram corresponds to a different distance between the samples (red dots) used to compute the angles*

## III.3. SIMILARITY AGGREGATION FOR 3D MODEL RETRIEVAL

In order to measure the similarity between two 3D models, described with the help of a set of 2Dviews (and associated 2D shape descriptors), the corresponding sets of views need to be put into correspondence and compared. A first, exhaustive matching approach consists of considering all the possibilities and to keep the optimal one (*i.e.*, the one which yields the smallest distance). This can be achieved by considering the set of all possible permutations between views. For each permutation, a global similarity measure is calculated by summing up all similarities measures between individual views. However, such a matching strategy is computationally intractable in practice: assuming that each model is described by a set of $N_P$ views, the total number of permutations equals $N_P!$. Even for a relatively low number of views, this results in a prohibitive number of permutations. For example, when $N_p = 10$, the number of permutations is of 3628800... Another matching solution consists of directly putting into correspondence the views obtained by the same camera, under the assumption that the 3D model presents a canonical representation in the virtual space, as the one obtained after PCA alignment. Figure III.16a illustrates two objects $M_A$, and $,M_B$, the associated projections and the corresponding distances stored as a similarity matrix. The distance $D(M_A, M_B)$ between the two 3D models is therefore obtained by summing up the distances stored on the diagonal of the matrix (Figure III.16b).

$$D(M_A, M_B) = \sum_{i=1}^{N_P} d\big(P_i(M_A), P_i(M_B)\big). \qquad (III.8)$$

From now on, this approach will be referred to as *Diagonal* matching strategy.

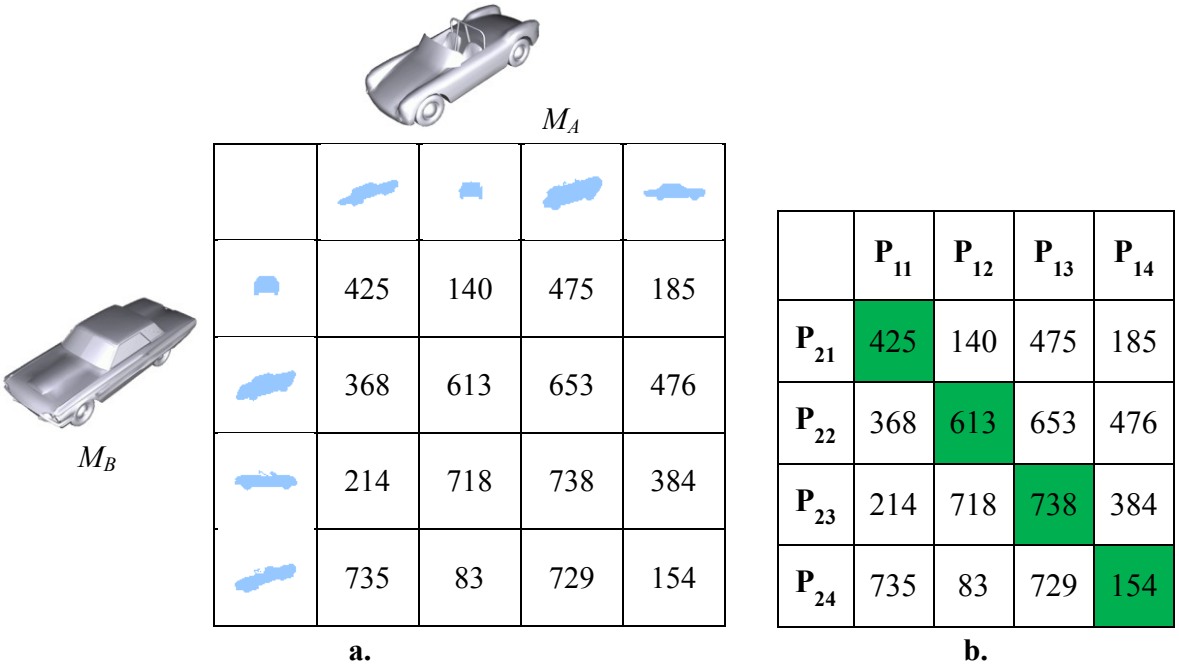



*Figure III.16 3D models matching.*
*a. The distance similarity matrix associated with two 3D models; b. Diagonal matching strategy.*

The main drawback of the *Diagonal* matching strategy is related to the limitations of the PCA alignment process, which fails in a certain number of situations (*cf*. Section II.1.1.2).

Therefore, we propose a different matching approach, so-called *Minimum*, which exploits a greedy strategy for fitting the various 2D views. When comparing two 3D models, the best match, corresponding to the minimal distance in the similarity matrix, if first determined (Figure III.17a). The corresponding views are considered as matched and ignored during the next steps. The process is successively applied upon the remaining sets of views, until all the projections are matched (Figure III.17 b-d). The global distance between two models is obtained by summing up the individual distances between the matched views.

| | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ |
|---|---|---|---|---|
| $P_{21}$ | 425 | 140 | 475 | 185 |
| $P_{22}$ | 368 | 613 | 653 | 476 |
| $P_{23}$ | 214 | 718 | 738 | 384 |
| $P_{24}$ | 735 | 83 | 729 | 154 |

**a.**

| | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ |
|---|---|---|---|---|
| $P_{21}$ | 425 | 140 | 475 | 185 |
| $P_{22}$ | 368 | 613 | 653 | 476 |
| $P_{23}$ | 214 | 718 | 738 | 384 |
| $P_{24}$ | 735 | 83 | 729 | 154 |

**b.**

| | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ |
|---|---|---|---|---|
| $P_{21}$ | 425 | 140 | 475 | 185 |
| $P_{22}$ | 368 | 613 | 653 | 476 |
| $P_{23}$ | 214 | 718 | 738 | 384 |
| $P_{24}$ | 735 | 83 | 729 | 154 |

**c.**

| | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ |
|---|---|---|---|---|
| $P_{21}$ | 425 | 140 | 475 | 185 |
| $P_{22}$ | 368 | 613 | 653 | 476 |
| $P_{23}$ | 214 | 718 | 738 | 384 |
| $P_{24}$ | 735 | 83 | 729 | 154 |

**d.**

*Figure III.17 3D models matching with the* Minimum *strategy.*
*a.-d. Intermediate steps of the Minimum strategy. The best match determined at each step is marked with green, while the projections which were already considered are marked with blue.*

## III.4. STORAGE AND COMPUTATIONAL ASPECTS

Within the context of 2D/3D indexing, the time required for both descriptor extraction and similarity measure computation represent key issues, since the number of shapes to be described and compared can be important in the case of large 3D model repositories. Another aspect to be taken into account is the descriptor size which has an impact on the required storage space.

Table III.1 presents the storage size ($D_S$), as well as the extraction ($T_E$) and similarity ($T_S$) computation times for each of the retained descriptors. The computation times reported here have been obtained on an Intel Xeon machine with 2.8GHz and 12GB RAM, under Windows 7 platform.

Regarding the storage size, all descriptors present similar values, except the HT which requires a significant larger storage space. The second drawback of the HT descriptor is the extraction time $T_E$ (about 0.1s for a given shape). However, if RS and ZM descriptors are faster to extract (11ms and 13ms, respectively), they require additional time (0.15s, respectively 0.5s) for the computation of the basis functions (Equations III.1 and III.4) employed for the object support function decomposition. Therefore, the most powerful descriptors in terms of extraction time are the AH (6.5ms) and CS (54ms). As the pre-processing is a mandatory stage for some descriptors (*e.g.*, HT, RS, ZM), the corresponding time was considered when computing $T_E$.

The last criterion is the similarity computation time $T_S$, which is determinant for the time of response to the queries. The slowest descriptors are HT and CS, which requires 120µs, respectively 76µs for each comparison. The RS, ZM and AH descriptors present all similar performance (84ns, 101ns, and 168ns, respectively): they are about 1000 times faster than HT and CS!

When a query is formulated, the time of response to the query $T_R$, in the case of the *minimum* aggregation strategy, can be expressed as:

$$T_R = T_A + N_P \cdot T_E + {N_P}^2 \cdot N_{DB} \cdot T_S \,, \qquad (III.9)$$

where:

- $N_P$ is the number of projections associated with each model;
- $N_{DB}$ is the number of models in the database.

Thus RS, ZM and AH descriptors allow performing more than 1,000,000 comparisons per second, which represents an important advantage for deploying real-life applications.

*Table III.1 Overview of adopted 2D shape descriptors*

| **Descriptor** | **$D_S$** | **$T_E$** (ms) | **$T_S$** (µs) | **Additional computation time ($T_A$)** |
|---|---|---|---|---|
| RS | 35 × integer | 11.33 | 0.084 | ART basis function computation (0.15s) |
| HT | 2048 × double | 111.04 | 120.13 | – |
| ZM | 41 × double | 13.13 | 0.101 | Zernike Moments computation (0.5s) |
| CS | 23 × integer | 54.21 | 73.17 | – |
| AH | 90 × integer | 6.57 | 0.168 | – |

*$D_S$ – Descriptor size; $T_E$ – Mean descriptor extraction time (including pre-processing); $T_S$–mean similarity computation time.*

## III.5. 3D MODEL DATABASES: VARIABILITY ANALYSIS

Let us now present and analyze the 3D model databases exploited in our work.

### III.5.1. MPEG-7 database

First, we have retained the MPEG-7 3D model database [Zaharia04]. The MPEG7 repository is composed of $N_M$=362 mesh models, divided into 23 semantic classes: airplanes, humanoids, cars, tanks, trucks, Formula 1 vehicles, motorcycles with three wheels, motorcycles with two wheels, helicopters, pistols, rifles, chess, screwdrivers, cylindrical shapes (missiles, cylinders, submarines), threes without leaves, trees, spherical objects, fingers and five letters categories (A to E).

Figure III.18 illustrates some sample models from the MPEG7 dataset.

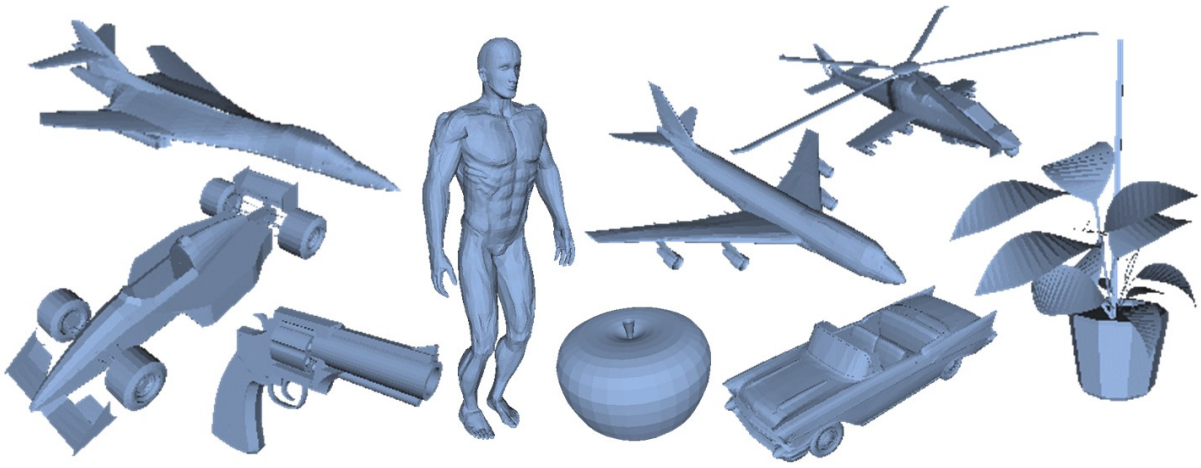


*Figure III.18 Sample models from the MPEG7 3D dataset.*

### III.5.2. Princeton database

The second 3D database considered is the Princeton Shape Benchmark (PSB) [Shilane04] which includes $N_M$=1814 models. The objects are divided into two groups of equal size corresponding to training and test model set. As there is no need for such a separation for our work, since we are not considering any learning methods, and in order to obtain a richer data set, we have joined the training and test models into the same corpus.

A multilevel, hierarchical classification is proposed for the PSB models. The first level includes two groups: natural and manmade objects. The second level is composed of 7 classes: vehicles, animals, household objects, buildings, furniture, plants and other. The third classification level includes 53, more detailed categories, like: winged vehicles, arthropod animals, lamp, hat, bridge, skeleton, flying creature, train... The full list of categories is provided in Annexe A2, Table A.2. Finally, the fourth and last classification level presents 161 categories as precise as biplane, commercial airplane, bee, walking human, race car, dining chair, barren tree... The list including all 161 categories can be found in Annexe A2, Table A.3. Figure III.19 illustrates some sample models from the Princeton Shape Benchmark.

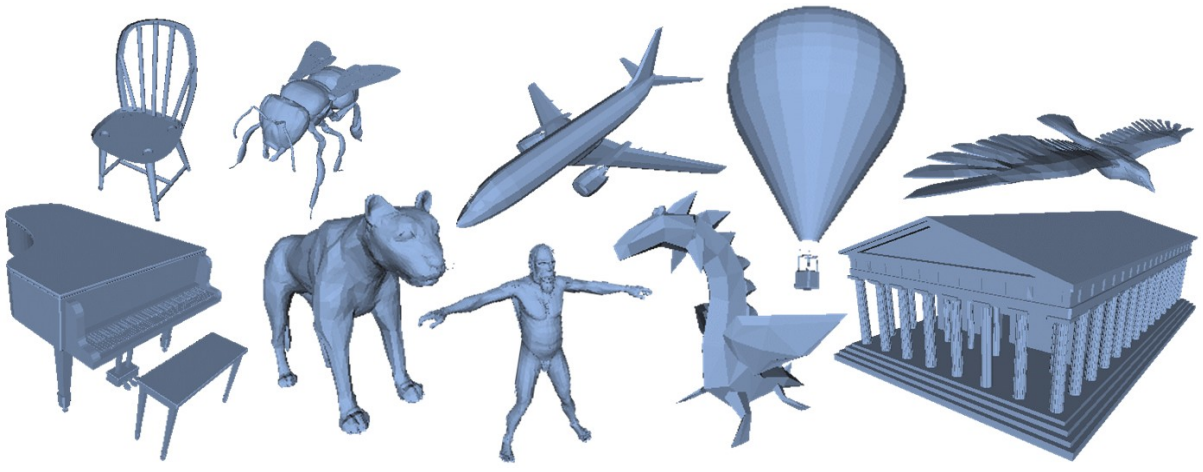


*Figure III.19 Sample models from the PSB 3D dataset.*

### III.5.3. Analysis of intra and inter class variability

Before an experimental evaluation of the 2D/3D indexing methods, it is important to analyse the test database in terms of intra and inter-class variability. In this section we propose an analytical evaluation of the intra and inter class variability of MPEG7 and PSB datasets, described previously. In the case of PSB, we have retained the two most detailed classification levels: the third level, where the 1814 objects are divided into 53 categories and the fourth level which includes 161 categories. From now on, the two cases will be referred to as PSB_53 and PSB_161. The MPEG7 dataset, which includes 362 models divided into 23 classes, will be referred to as MPEG7_23.

Let us start by presenting the evaluation measures adopted in our work.

#### III.5.3.1. Evaluation measures

The intra-class variability evaluates how similar are the models within the same semantic category, while the inter-class variability indicates how similar are the categories of objects included in the database. For both 3D model retrieval and 2D object recognition purpose it is important to have a large variety of objects and classes.

We propose to measure the intra-class variability $\delta_{DB}$ with the help of the mean distance $\delta_X$ between all the elements within a given category $X$:

$$\delta_{DB} = \frac{\sum_{X \in DB} \delta_X}{|DB|}, \qquad (III.10)$$

where $|DB|$ represents the cardinality of the DB, *i.e.*, the number of categories included in the database and

$$\delta_X = \frac{\sum_{M_i \in X} \sum_{M_j \in X} D(M_i, M_j)}{|X|^2}, \qquad (III.11)$$

where:

- $D(M_i, M_j)$ is a distance between the two 3D models $M_i$ and $M_j$. In our case this distance corresponds to the aggregated similarity measure (*cf.* Section III.3) between 2D shape descriptors within a 2D/3D indexing framework;
- $|X|$ represents the cardinality of $X$.

The distance between two classes $X$ and $Y$ is then defined as the mean distance between a model of $X$ and a model of $Y$:

$$\Delta(X, Y) = \frac{\sum_{M_i \in X} \sum_{M_j \in Y} D(M_i, M_j)}{|X| \cdot |Y|}. \qquad (III.12)$$

The distance $\Delta_X$ between a class $X$ and the database (DB) is given by the equation:

$$\Delta_X = \frac{\sum_{Y \in DB} \Delta(X, Y)}{|DB|}. \qquad (III.13)$$

A high value of $\Delta_X$ denotes that the elements of $X$ are easily to distinguish among the others objects of the database.

The inter-class variability ($\Delta_{DB}$) is defined as the mean distance between all the classes of the database:

$$\Delta_{DB} = \frac{\sum_{X \in DB} \sum_{Y \in DB} \Delta(X,Y)}{|DB|^2}. \quad (III.14)$$

Finally, we analyse for each pair of classes how independent they are one from the other. Thus, we define the separability $S(X,Y)$ between two classes of objects $X$ and $Y$ with respect to their class-to-class distance and intra-class variability:

$$S(X,Y) = \frac{\Delta(X,Y)}{\delta_X + \delta_Y}, \quad (III.15)$$

and the separability $S_X$ of a class as the mean separability between the considered class and all the categories of the database:

$$S_X = \frac{\sum_{Y \in DB} S(X,Y)}{|DB|}. \quad (III.16)$$

The global separability of the database is defined as the mean class separability:

$$S_{DB} = \frac{\sum_{X \in DB} S_X}{|DB|}, \quad (III.17)$$

and reflects how easy to distinguish are the elements of the database.

Figure III.20 illustrates the intra and inter class variability analysis in a 2D Euclidian space. Each class is represented by a circle, whose radius is equal to the intra-class variability ($\delta_X$). The more two circles are overlapped, the less the corresponding categories can be separated (*e.g.,* classes A and F).

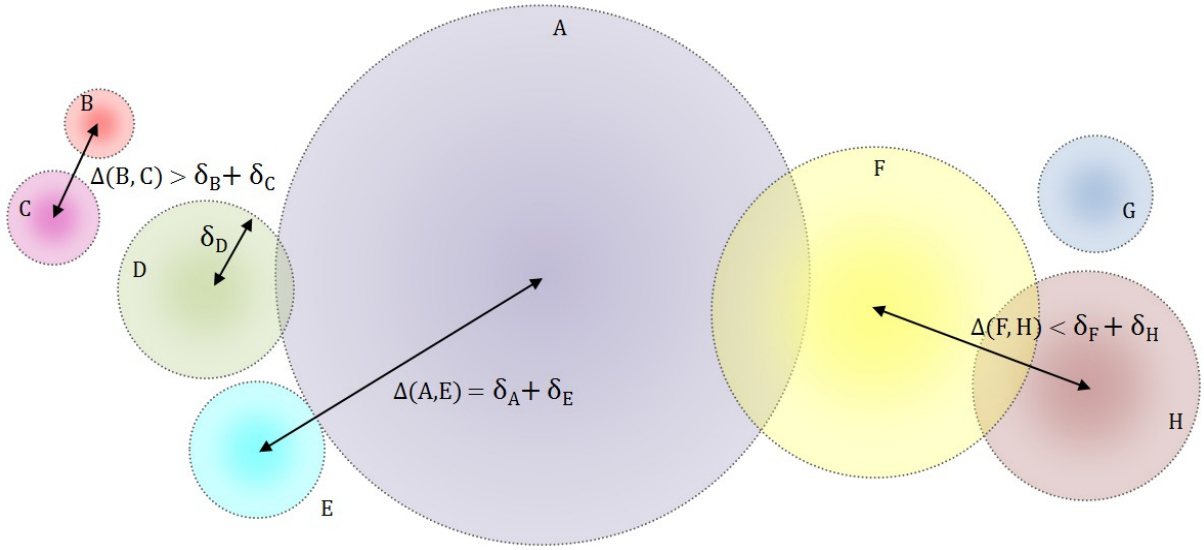



*Figure III.20 Intra and inter class variability.*

As the range of the distance value depends on the descriptors considered and on their corresponding similarity measures, a mean distance $\overline{D}$ has been computed over all 3D models in the databases and for each descriptor. The mean distance $\overline{D}$ is defined as follows:

$$\overline{D} = \frac{\sum_{M_i \in DB} \sum_{M_j \in DB} \sum_{pm_k \in PM} D_{pm_k}(M_i, M_j)}{N_{DB}^2 \cdot |PM|}, \quad (III.18)$$

where:

- $PM = \{PCA3, PCA7, DODECA, DDPCA, OCTA9, OCTA33, RV6, RV10\}$, *i.e.*, the set of all possible projection methods;
- $N_{DB}$ represents the number of models included in the database.

Table III.2 summarizes the mean distances obtained for the various descriptors retained in our work.

*Table III.2 Normalization reference*

| | **AH** | **CS** | **HT** | **RS** | **ZM** |
|---|---|---|---|---|---|
| $\overline{D}$ | 719.8 | 1.26 | 21.2 | 1.67 | 8.31 |

The mean distance is used as reference to normalize the $\delta_X$, $\delta_{DB}$, $\Delta(X,Y)$, $\Delta_X$ and $\Delta_{DB}$ measures, by dividing them with the corresponding value of $\overline{D}$. This makes it possible to define a common range for expressing the intra and inter class variabilities obtained for various descriptors, and thus to allow comparisons between them.

In Section III.3 we have presented two matching strategies, so-called *diagonal* and *minimum*. The *diagonal* approach is based on the assumption that the 3D model presents a canonical representation, which is not true for all viewing angle selection techniques. Therefore, for inter and intra class variability analysis, only the *minimum* matching approach is considered, as it is appropriate for all the viewing angle selection strategies retained in our work.

### III.5.3.2. Results and discussion

The global mean values of the inter-class variability $\Delta_{DB}$ (*cf.* Equation III.14) and separability $S_{DB}$ (*cf.* Equation III.17) for the MPEG7_23, PSB_53 and PSB_161 databases are presented in Table III.3, Table III.4 and Table III.5 respectively. Each column corresponds to one 2D shape descriptor (*i.e.,* AH, CS, HT, RS and ZM) and each row to one projection method (*i.e.,* PCA3, PCA7, DODECA, DDPCA, OCTA9, OCTA33, RV6 and RV10). The last row contains the average values obtained for a given database, over all the considered 2D/3D indexing methods.

First, it can be observed that the inter-class variability is very similar for all databases (the difference is less than 4%). The lower value of $\Delta_{PSB\_161}$ compared to $\Delta_{PSB\_53}$ can be explained by the fact that some categories in PSB_161 are very similar (*e.g.,* SUV and jeep vehicles).

Regarding the separability of each database, it can be observed that MPEG7_23 database presents the highest values, with a mean value of 1.155, which means that the categories of MPEG7_23 are better defined in the 2D/3D indexing space, in the sense that they present less overlapping.

In contrast, PSB_53 presents lower separability than PSB_161 (0.688 compared to 0.756), while including fewer categories. Actually, even if the classes of PSB_53 are less numerous, they are more spread, as indicated by the intra-class variability (Table III.6, Table III.7 and Table III.8).

The higher intra-class variability of PSB_53 (0,736) compared to PSB_161 (0,659) is naturally explained by the way the two databases are constructed: both contain the same 3D models (*i.e.,* the Princeton shape benchmark) but differently divided. PSB_161 presents very definite classes (*e.g.,* F117 aircraft, glider airplane, biplane, commercial airplane, fighter jet aircraft, multi-fuselage airplane...), while PSB_53 merges them in more general categories (*e.g.,* winged aircraft).

The MPEG7_23 presents even less intra-class variability, with a mean value of 0.511.

Figure III.21, Figure III.22 and Figure III.23 illustrate the intra-class variability for each category of MPEG7_23, PSB_53 and PSB_161 databases respectively, in the case of PCA7 viewing angle selection strategy, which is a representative choice illustrating the global behaviours (*cf.* Table III.6, Table III.7 and Table III.8). The correspondence between the index number and the name of each category can be found in Annexe A2.

In Figure III.21 we can observe that some classes of the MPEG7_23 database (*e.g.,* letters A to E categories) present very low variability. Actually, each letter category includes models which are differentiated one from each other only by their topologies. This observation confirms the validity of our analysis and explains the low global intra-class variability of the MPEG7_23 database. The PSB_53 and PSB_161 databases are more homogeneous in terms of intra-class variability (Figure III.22, Figure III.23). In addition, the lowest intra-class variability values are higher in the case of PSB databases than for MPEG7_23.

The class-to-class separability $S(X,Y)$ (Equation III.15) and distance $\Delta(X,Y)$ (Equation III.12) are illustrated in Figure III.24, Figure III.25 and Figure III.26 for MPEG7_23, PSB_53 and PSB_161 respectively. The colour-bar legend present at the bottom of each figure indicates the relation between the value of $S(X,Y)/\Delta(X,Y)$ and the colours in the images. Each row and each column of those images corresponds to one category (*cf.* Annexe A2). The numbering starts from left to right and from top to bottom. The lower-left half of each image stores the separability values and the upper-right half indicates the class-to-class distance.

The results show that some categories of objects present high distances to all or almost all other classes, which means that those categories are more isolated in the 2D/3D indexing space (*e.g.,* spherical category of the MPEG7_23 database). This behaviour is highlighted by the L-shaped red lines in Figure III.24, Figure III.25 and Figure III.26.

The database analysis also allows us to make a first evaluation of the projection methods and 2D shape descriptors. Thus, the separability measure let us evaluate how appropriate is each 2D/3D indexing method for each semantic class.

It can be observed that contour-based descriptors (*i.e.,* AH and CS) have a similar discrimination power for all categories of models. This behaviour is highlighted by the uniformity of the distance and separability colours in the corresponding images. On the other hand, the region-based descriptors seem to advantage some categories (*e.g.,* motorcycles with three wheels in MPEG7_23 database), while disadvantaging other categories (*e.g.,* trees models from MPEG7_23 database).

The fact that some descriptors are more suitable for certain classes of objects is also visible in Figure III.21, Figure III.22 and Figure III.23, where it can be observed that the intra-class variability is not the same for all descriptors. In other words, one descriptor can better encapsulate the similarities within a given class than the other descriptors. For example, in the case of PSB_53 database, AH indicates the lowest intra-class variability for swing-set models ($48^{th}$ category) but the highest value for wheel objects ($53^{th}$ category).This shows the potential interest of combining various descriptors within an intelligent aggregation mechanism for category recognition purposes.

The results analysis shows that the most challenging 3D model databases are the Princeton related ones, which offer the advantage of an increased number of categories, but present relatively low separability properties.

In the following section we present an experimental evaluation of the proposed 2D/3D indexing methods on the retained 3D model databases.

*Table III.3 MPEG7_23 database: inter-class variability and separability*

| $\Delta_{MPEG7\_23}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 1,108 | 0,960 | 1,033 | 1,025 | 1,057 |
| **PCA7** | 1,043 | 0,911 | 0,980 | 1,010 | 0,980 |
| **DODECA** | 1,006 | 0,892 | 0,962 | 0,999 | 0,952 |
| **DDPCA** | 1,024 | 0,898 | 0,955 | 1,012 | 0,955 |
| **OCTA9** | 1,059 | 0,914 | 0,984 | 1,014 | 1,002 |
| **OCTA33** | 1,018 | 0,882 | 0,960 | 0,992 | 0,960 |
| **RV6** | 1,012 | 0,871 | 0,943 | 0,936 | 0,903 |
| **RV10** | **0,999** | 0,855 | 0,925 | 0,917 | 0,887 |
| ***mean*** | **0.970** | | | | |

| $S_{MPEG7\_23}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 1,248 | 1,145 | 1,301 | 1,422 | 1,536 |
| **PCA7** | 1,214 | 1,108 | 1,338 | 1,512 | 1,580 |
| **DODECA** | 0,934 | 0,892 | 0,854 | 0,974 | 0,986 |
| **DDPCA** | 1,180 | 1,070 | 1,347 | 1,545 | 1,552 |
| **OCTA9** | 1,216 | 1,083 | 1,266 | 1,512 | 1,513 |
| **OCTA33** | 1,195 | 1,088 | 1,320 | 1,543 | 1,554 |
| **RV6** | 0,895 | 0,883 | 0,792 | 0,937 | 0,984 |
| **RV10** | 0,918 | 0,859 | 0,811 | 0,951 | 1,019 |
| ***mean*** | **1.155** | | | | |

*Table III.4 PSB_53 database: inter-class variability and separability*

| $\Delta_{PSB\_53}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 1,069 | 0,953 | 1,134 | 1,062 | 1,144 |
| **PCA7** | 1,004 | 0,932 | 1,065 | 1,030 | 1,059 |
| **DODECA** | 0,970 | 0,914 | 1,035 | 1,022 | 1,031 |
| **DDPCA** | 0,984 | 0,921 | 1,035 | 1,027 | 1,034 |
| **OCTA9** | 1,023 | 0,893 | 1,089 | 1,037 | 1,093 |
| **OCTA33** | 0,977 | 0,921 | 1,042 | 1,011 | 1,041 |
| **RV6** | 0,978 | 0,896 | 1,024 | 0,988 | 1,018 |
| **RV10** | 0,964 | 0,881 | 1,017 | 0,977 | 1,003 |
| ***mean*** | **1.007** | | | | |

| $S_{PSB\_53}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 0,706 | 0,665 | 0,661 | 0,690 | 0,717 |
| **PCA7** | 0,704 | 0,675 | 0,663 | 0,724 | 0,726 |
| **DODECA** | 0,694 | 0,667 | 0,639 | 0,720 | 0,713 |
| **DDPCA** | 0,705 | 0,671 | 0,661 | 0,725 | 0,720 |
| **OCTA9** | 0,708 | 0,677 | 0,660 | 0,711 | 0,724 |
| **OCTA33** | 0,710 | 0,665 | 0,666 | 0,729 | 0,728 |
| **RV6** | 0,682 | 0,649 | 0,621 | 0,696 | 0,701 |
| **RV10** | 0,684 | 0,651 | 0,624 | 0,694 | 0,708 |
| ***mean*** | **0.688** | | | | |

*Table III.5 PSB_161 database: inter-class variability and separability*

| $\Delta_{PSB\_161}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 0,948 | 0,948 | 1,046 | 0,984 | 1,123 |
| **PCA7** | 0,990 | 0,921 | 1,007 | 0,990 | 1,030 |
| **DODECA** | 0,958 | 0,903 | 0,995 | 0,988 | 0,997 |
| **DDPCA** | 0,972 | 0,910 | 0,999 | 0,984 | 1,003 |
| **OCTA9** | 1,009 | 0,914 | 1,012 | 0,987 | 1,063 |
| **OCTA33** | 0,965 | 0,884 | 0,986 | 0,987 | 1,010 |
| **RV6** | 0,970 | 0,881 | 0,968 | 0,989 | 0,991 |
| **RV10** | 0,954 | 0,866 | 0,957 | 0,990 | 0,973 |
| ***mean*** | **0.976** | | | | |

| $S_{PSB\_161}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 0,772 | 0,730 | 0,727 | 0,764 | 0,805 |
| **PCA7** | 0,773 | 0,744 | 0,740 | 0,812 | 0,822 |
| **DODECA** | 0,752 | 0,729 | 0,690 | 0,789 | 0,791 |
| **DDPCA** | 0,772 | 0,741 | 0,736 | 0,810 | 0,815 |
| **OCTA9** | 0,770 | 0,728 | 0,731 | 0,786 | 0,814 |
| **OCTA33** | 0,775 | 0,744 | 0,740 | 0,812 | 0,821 |
| **RV6** | 0,727 | 0,697 | 0,663 | 0,755 | 0,771 |
| **RV10** | 0,729 | 0,698 | 0,669 | 0,753 | 0,771 |
| ***mean*** | **0.756** | | | | |

*Table III.6 MPEG7_23 database: intra-class variability*

| $\delta_{MPEG7\_23}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 0,522 | 0,505 | 0,539 | 0,510 | 0,500 |
| **PCA7** | 0,496 | 0,485 | 0,506 | 0,478 | 0,459 |
| **DODECA** | 0,556 | 0,534 | 0,602 | 0,537 | 0,515 |
| **DDPCA** | 0,496 | 0,484 | 0,494 | 0,479 | 0,454 |
| **OCTA9** | 0,504 | 0,490 | 0,520 | 0,489 | 0,479 |
| **OCTA33** | 0,487 | 0,470 | 0,500 | 0,468 | 0,457 |
| **RV6** | 0,583 | 0,522 | 0,630 | 0,529 | 0,498 |
| **RV10** | 0,562 | 0,522 | 0,605 | 0,514 | 0,478 |
| ***mean*** | **0.511** | | | | |

*Table III.7 PSB_53 database: intra-class variability*

| $\delta_{PSB\_53}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 0,768 | 0,723 | 0,872 | 0,782 | 0,816 |
| **PCA7** | 0,722 | 0,697 | 0,816 | 0,725 | 0,743 |
| **DODECA** | 0,707 | 0,691 | 0,821 | 0,723 | 0,734 |
| **DDPCA** | 0,708 | 0,693 | 0,794 | 0,722 | 0,731 |
| **OCTA9** | 0,732 | 0,666 | 0,837 | 0,742 | 0,770 |
| **OCTA33** | 0,698 | 0,699 | 0,794 | 0,707 | 0,729 |
| **RV6** | 0,726 | 0,694 | 0,833 | 0,721 | 0,738 |
| **RV10** | 0,712 | 0,681 | 0,824 | 0,714 | 0,721 |
| ***mean*** | **0.736** | | | | |

*Table III.8 PSB_161 database: intra-class variability*

| $\delta_{PSB\_161}$ | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| **PCA3** | 0,693 | 0,661 | 0,786 | 0,698 | 0,713 |
| **PCA7** | 0,650 | 0,630 | 0,724 | 0,633 | 0,636 |
| **DODECA** | 0,645 | 0,629 | 0,747 | 0,640 | 0,636 |
| **DDPCA** | 0,640 | 0,624 | 0,705 | 0,629 | 0,624 |
| **OCTA9** | 0,665 | 0,637 | 0,746 | 0,656 | 0,664 |
| **OCTA33** | 0,632 | 0,603 | 0,706 | 0,618 | 0,624 |
| **RV6** | 0,673 | 0,639 | 0,778 | 0,651 | 0,652 |
| **RV10** | 0,661 | 0,627 | 0,764 | 0,645 | 0,639 |
| ***mean*** | **0.659** | | | | |

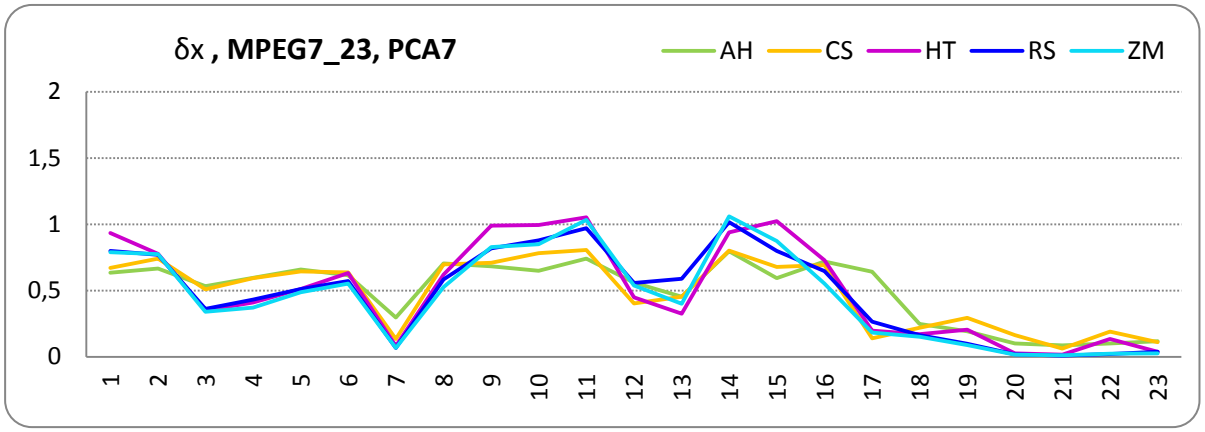



*Figure III.21 MPEG7_23 database: Intra-class variability with PCA7 strategy.*

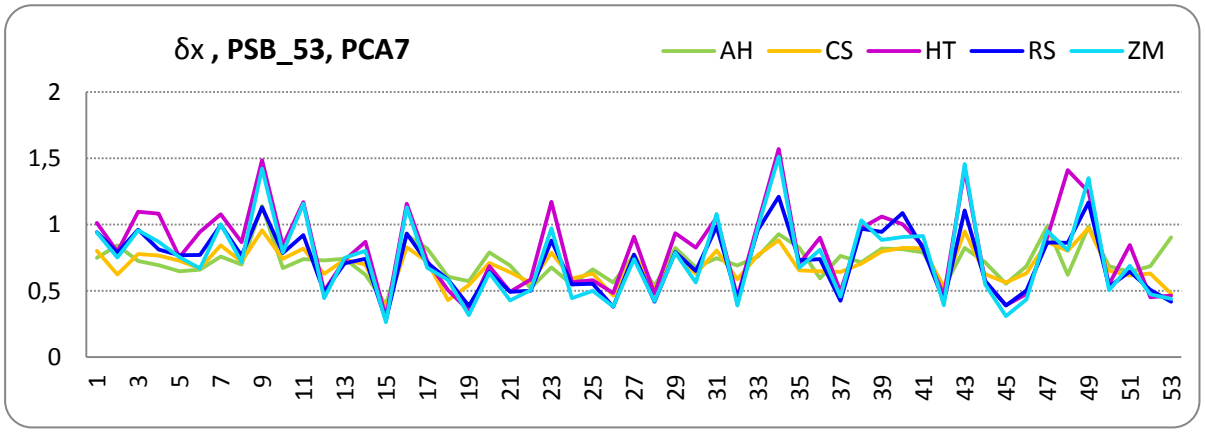



*Figure III.22 PSB_53 database: Intra-class variability with PCA7 strategy.*

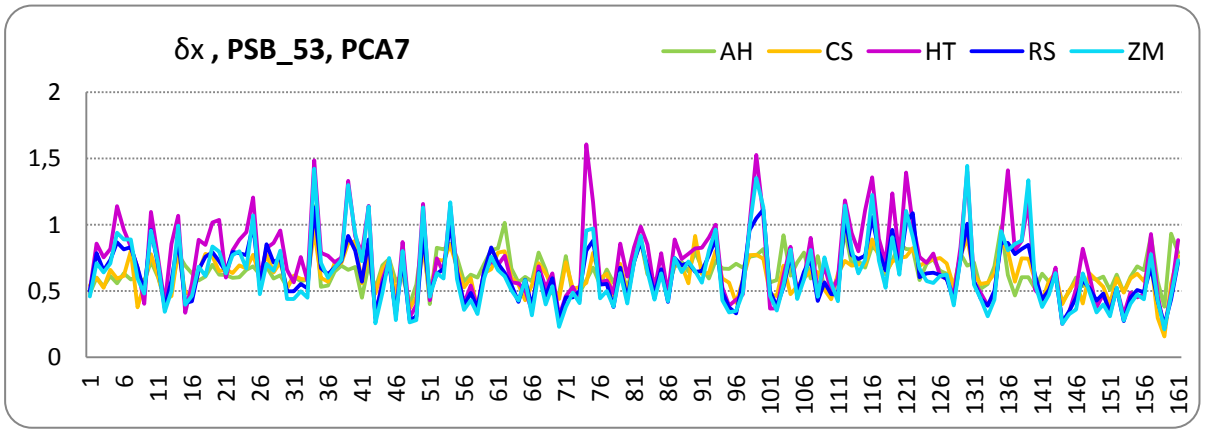



*Figure III.23 PSB_161 database: Intra-class variability with PCA7 strategy.*

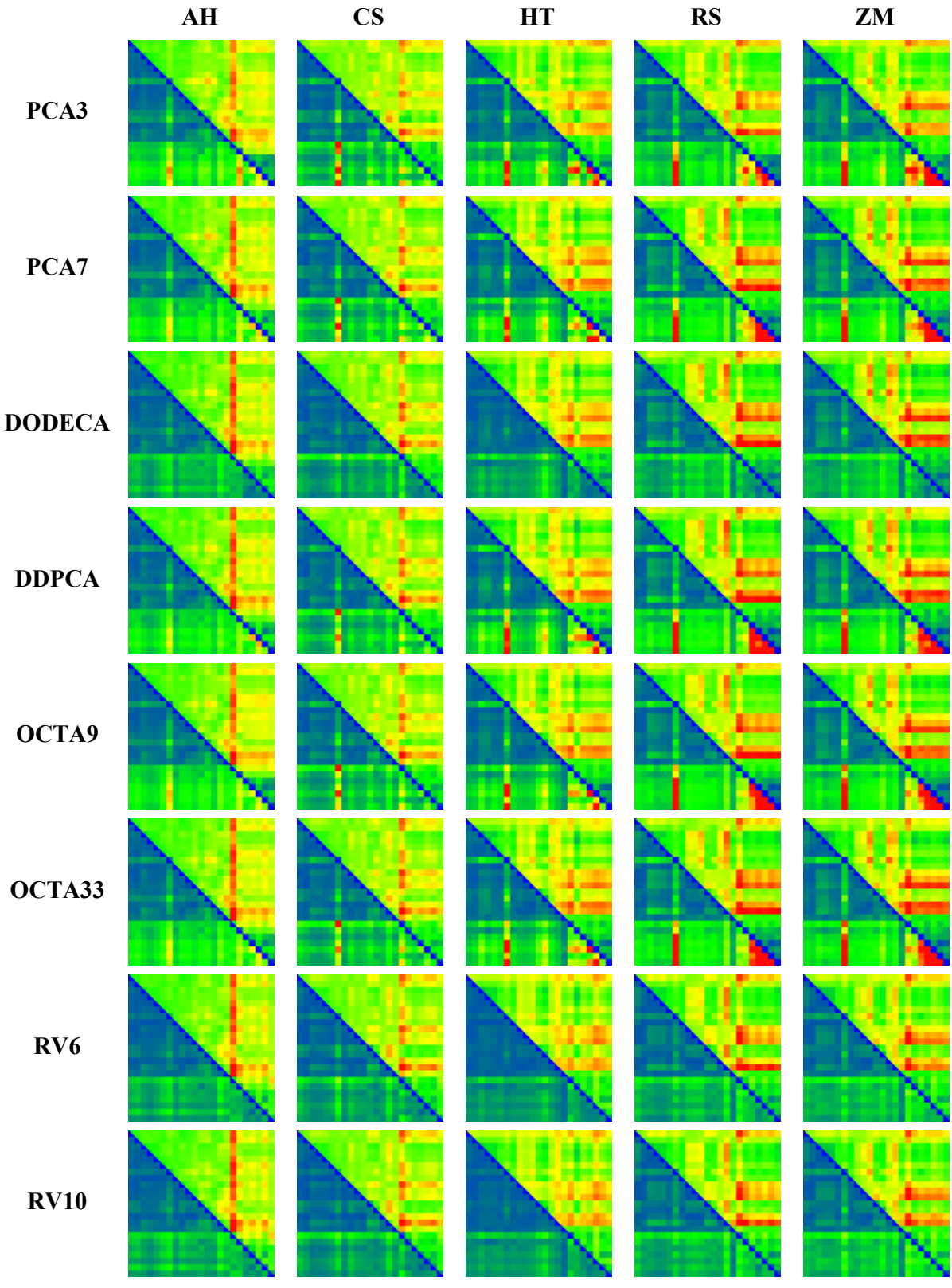



*Figure III.24 Separability / inter-class variability – MPEG7_23 database.*

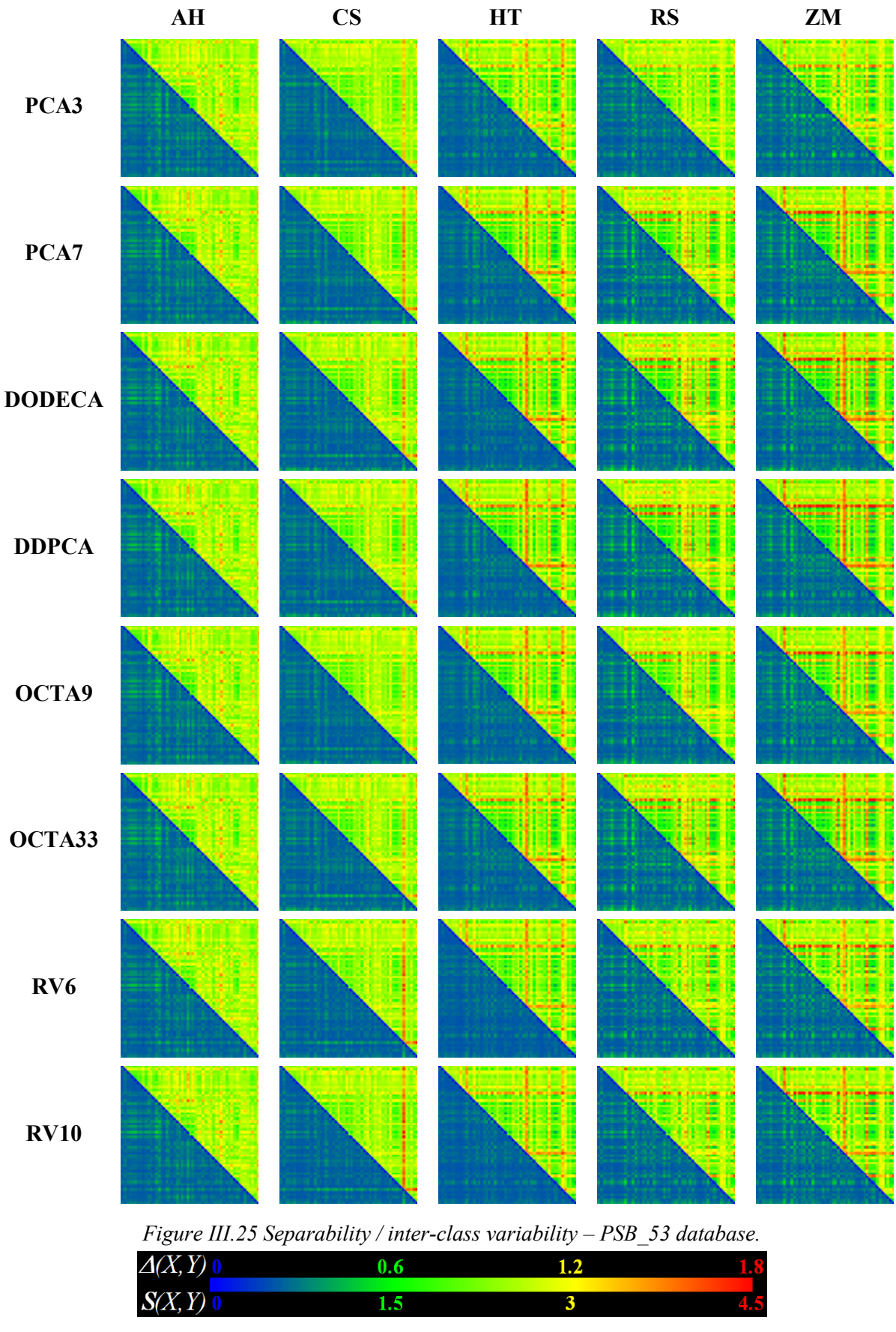



*Figure III.25 Separability / inter-class variability – PSB_53 database.*

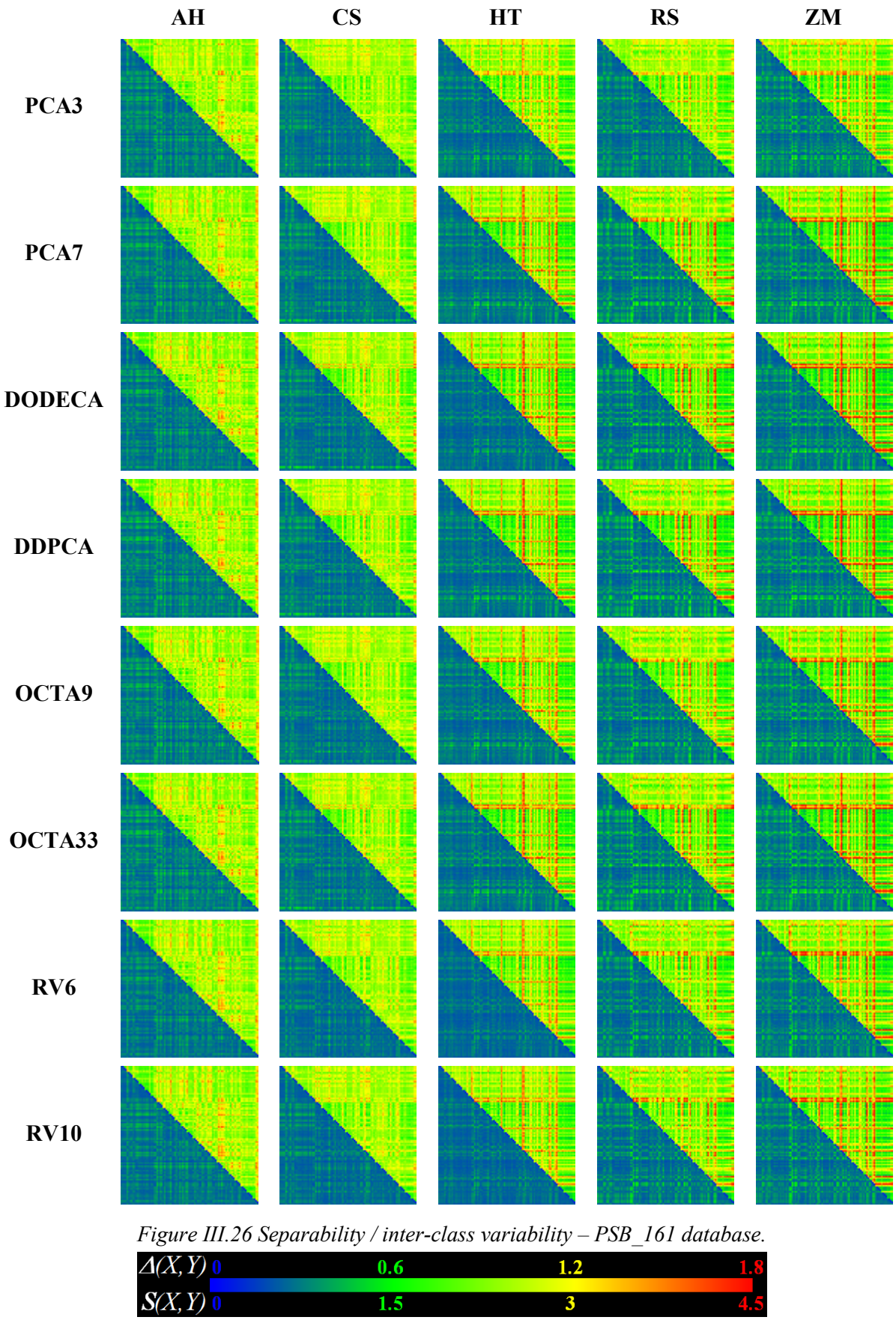



*Figure III.26 Separability / inter-class variability – PSB_161 database.*

## III.6. EXPERIMENTAL EVALUATION

In this section we propose an experimental evaluation of the view-based 3D model retrieval framework and analyse the influence of each element involved within the 2D/3D indexing framework (*i.e.,* projection strategy, 2D shape descriptor, matching approach) on the retrieval results.

Let us first present the evaluation protocol adopted in our work.

### III.6.1. Evaluation protocol

The retrieval performances are separately measured for each database (*i.e.,* MPEG7_23, PSB_53 and PSB_161) and each object is used to formulate a query on the corresponding database. A retrieved model is estimated as correct if it belongs to the same category as the query object, with respect to the classification of the considered database.

As objective evaluation measure we have retained the *First Tier* (*FT*), the *Second Tier* (*ST*) [Shilane04, Zaharia04] and the *Precision-Recall* (*PR*) curve [Chen03].

The *FT* score is defined as the percentage of correctly retrieved models within the first $N_Q$ positions:

$$FT = \frac{N_{C|Q}}{N_Q}, \qquad (III.19)$$

where:
- $N_Q$ represents the total number of correct objects that can be retrieved;
- $N_{C|Q}$ represents the number of correct objects within the first $N_Q$ positions.

The *ST* score, also known as *Bull Eye* score, is defined as the percentage of correct retrieved models within the first $2N_Q$ positions:

$$ST = \frac{N_{C|2Q}}{N_Q}, \qquad (III.20)$$

where:
- $N_{C|2Q}$ represents the number of correct objects within the first $2N_Q$ positions.

In the case of an ideal retrieval system both *FT* and *ST* are equal to 1.

The *Precision* represents the percentage of retrieved objects that are relevant:

$$P = \frac{N_{C|R}}{N_R}, \qquad (III.21)$$

where:
- $N_R$ represents the number of retrieved objects;
- $N_{C|R}$ represents the number of correct objects among the retrieved ones.

The *Recall* represents the fraction of relevant models that are retrieved:

$$R = \frac{N_{C|R}}{N_Q}. \qquad (III.22)$$

The last two metrics are represented as a curve, namely the Precision-Recall curve.

In order to illustrate the various evaluation measures retained, let us consider the airplane retrieval example in Figure III.27. If the total number of airplanes in the database is $N_Q$=8, then we obtain *FT* = 5/8 = 0.625, *ST* = 7/8 = 0.875 and *PR* curve as illustrated in Figure III.28.

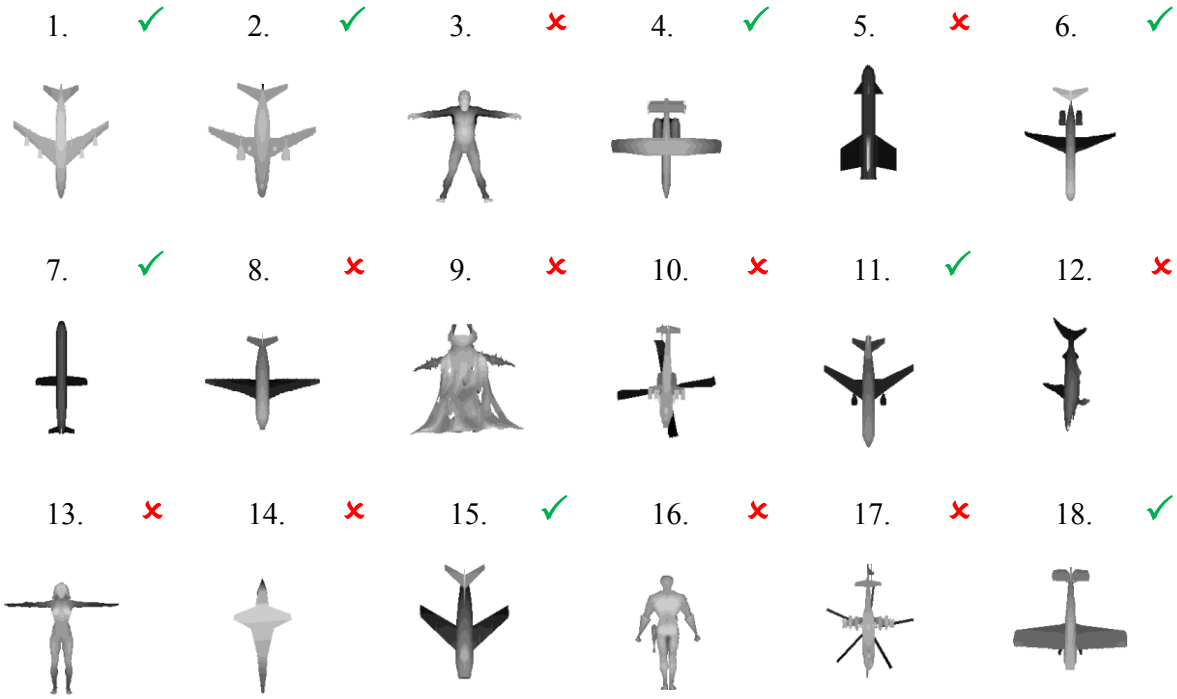



*Figure III.27 Example of airplane retrieval result.*

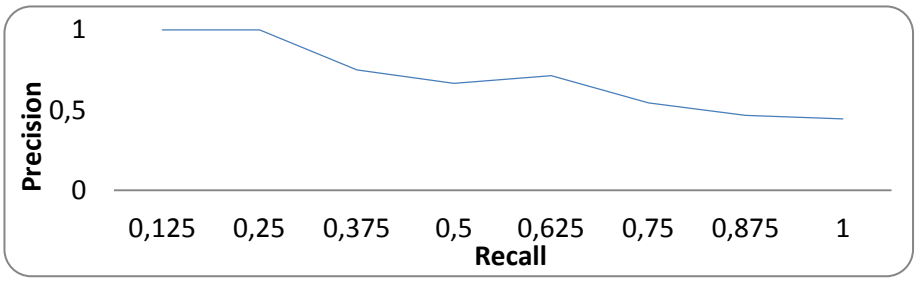



*Figure III.28 The Precision-Recall curve associated with the example in Figure III.27.*

### III.6.2. 3D model retrieval results and discussion

Figure III.29 and Figure III.30 illustrate the *FT* and *ST* scores obtained on the MPEG7_23 database with *minimum*, respectively *diagonal* matching strategy. The *FT* and *ST* scores obtained in the case of PSB_53 database are presented in Figure III.31 and Figure III.32 for the two

matching strategies. Finally, Figure III.33 and Figure III.34 present the *FT* and *ST* score for the PSB_161 database. The precise values of the FT and ST scores are provided in Table III.9 to Table III.14.

Concerning the case of DODECA, RV6 and RV10 strategies, the 3D model has an either random or object-dependent orientation in the virtual space. In such cases, the underlying hypothesis of a one-to-one correspondence between views, which represents the basis of the *diagonal* matching approach, does not hold. Thus, in the case of the *diagonal* matching strategy, solely the scores obtained for the projection methods which employ a canonical representation of the 3D model (*i.e.,* PCA3, PCA7, DDPCA, OCTA9 and OCTA33) are pertinent and reported in our results.

The *Precision-Recall* curves obtained for the three databases when using the *minimum* matching strategy are presented in Figure III.35, Figure III.37 and Figure III.39. Figure III.36, Figure III.38 and Figure III.40 present the *Precision-Recall* curves in the case of the *diagonal* matching approach.

As expected from the database analysis, the highest retrieval performances are obtained on the MPEG7_23 repository, with *FT* and *ST* scores up to 72.3% and 82.7%, respectively. In the case of PSB_53 dataset the best scores attain 35.4% in terms of *FT* and 48.2% in terms of *ST*. The retrieval performances on the PSB_161 (which presents lower intra-class variability and higher global separability) are slightly superior, while including three times more categories. Thus, we obtain here an *FT* score of up to 38.2% and an *ST* up to 50.2%. Such results are in accordance with the database analysis proposed in Section III.5.

When analysing the scores, the first observation is that the retrieval performances do not improve significantly when increasing the number of views. Thus, the results obtained with PCA3 are very close to those obtained with PCA7.

Globally, the variation in terms of FT score between PCA3 and OCTA33 (*i.e*., between 3 and 33 views) is of maximum 3.6%, whatever the database and the various shape descriptors considered.

The OCTA9 strategy ensures the highest retrieval performances in a large majority of cases. When passing from OCTA9 to OCTA33, the retrieval scores can degrade.

Such a behaviour can be explained by the following three facts:

- The shape characteristics are well captured by the first, PCA-based 9 views;
- When increasing the number of views, the inherent shape characterization ambiguities related to the various descriptors involved can lead to erroneous matches. This behaviour is stronger for the CS descriptor (5% decrease in FT score), which involves a filtering/early rejection procedure based on global shape characteristics.
- In the case of 3D models presenting symmetries, the set of views contains redundancies, which are reducing *de facto* the number of useful views.

Concerning the representative view selection strategies (RV6 and RV10), the related performances are inferior to PCA-based approaches. Such approaches generate object-dependent sets of view. Since the angles of views are different from one object to another, the corresponding

2D shapes to be matched can present significant variations, which make it difficult to derive pertinent similarity measures. Moreover, the RV6 and RV10 retrieval performances are quite equivalent with those corresponding to the DODECA uniform repartition strategy.

When comparing the *minimum* and *diagonal* matching strategies, we can observe that the *diagonal* approach leads to slightly superior results (about 2% of gain in both FT and ST scores). Actually, the *minimum* strategy aims to be more generic and able to match arbitrary views, such as those obtained by the DODECA, RV6 and RV10 strategies. The price to pay is the slight overall decrease in retrieval performances. In addition let us note that no global information concerning the relative position of the matched views is considered in this case. Determining a global match instead of adopting a greedy, *per view* matching strategy would be an interesting axis of research to explore in our future work.

The results obtained also show the superiority of contour-based shape descriptors with respect to region-based approaches. This observation appears clearly from the analysis of both FT/ST scores and *Precision-Recall* curves. Globally, the contour based-descriptors outperform the region-based techniques by up to 13% in terms of *FT* score and up to 15% in terms of *ST* score and with a mean difference between contour and region based descriptors of about 5% for both *FT* and *ST* scores (when comparing the results obtained with the same projection strategy and on the same database).

A finer analysis of results shows that the maximum difference between the performance of RS and ZM descriptors (for a given database and same projection strategy) is always less than 3% in terms of both *FT* and *ST* scores, with a slight superiority of the ZM descriptor. The HT descriptor leads to poorest overall retrieval on the three databases and for almost all 2D/3D indexing techniques. The difference between HT and the other descriptors is more significant in the case of DODECA, RV6 and RV10 with the *minimum* matching strategy, as show the low HT *Precision-Recall* curve in the corresponding figures. This behaviour leads to the conclusion that HT descriptor is more sensitive to small change of the shape, like those generated by the variation of the camera position. Such modifications seem to produce more ambiguity in the case of HT than when other descriptors are employed.

Concerning the two contour-based representations considered, the results demonstrate the superiority of the AH descriptor over the CS approach, with a global gain in performances of 2-3% over all databases and viewing angle selection strategies. This behaviour is also confirmed by the *Precision-Recall* curves presented.

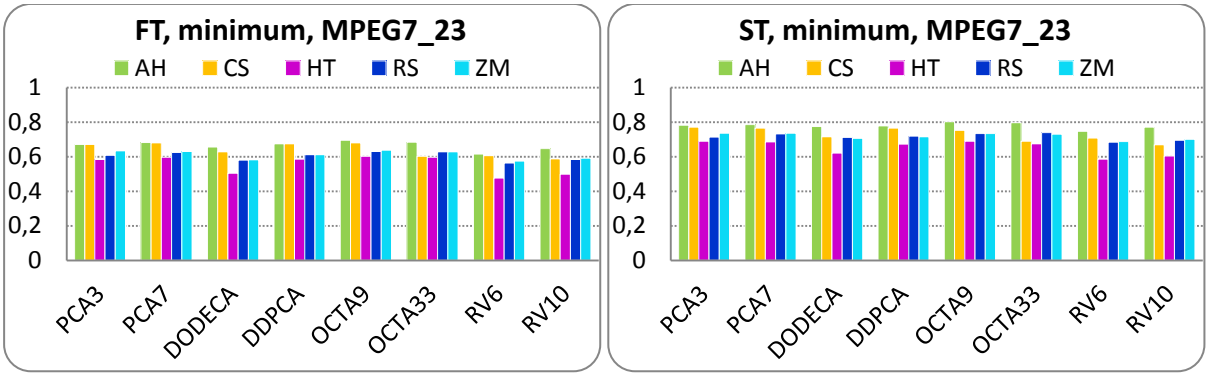



*Figure III.29 MPEG7_23 database: FT and ST score,* minimum *matching strategy.*

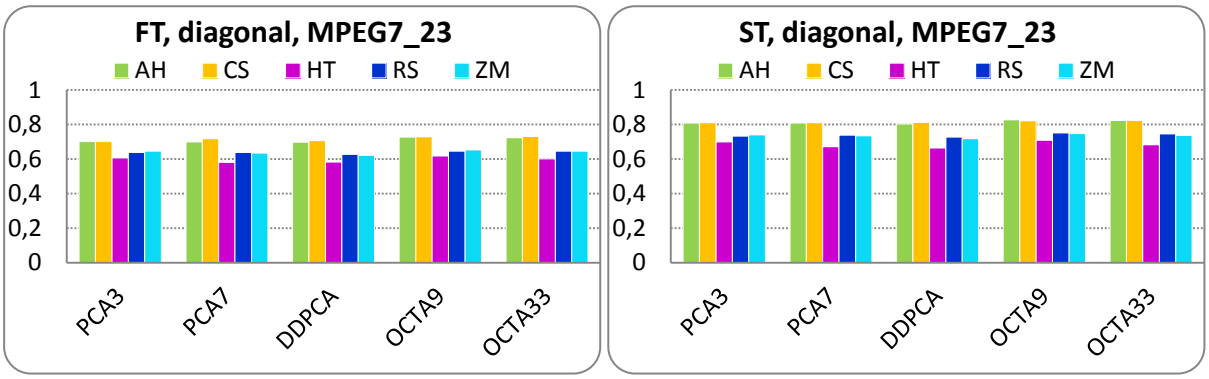



*Figure III.30 MPEG7_23 database: FT and ST score,* diagonal *matching strategy.*

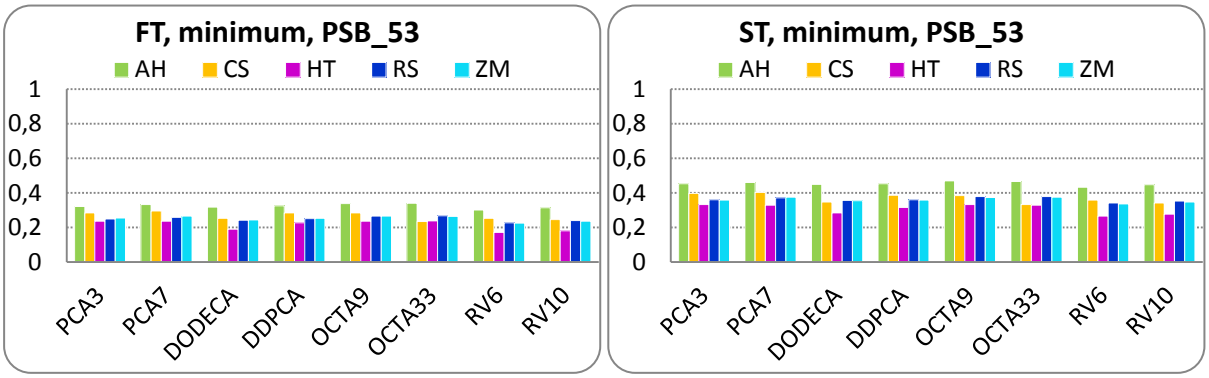



*Figure III.31 PSB_53 database: FT and ST score,* minimum *matching strategy.*

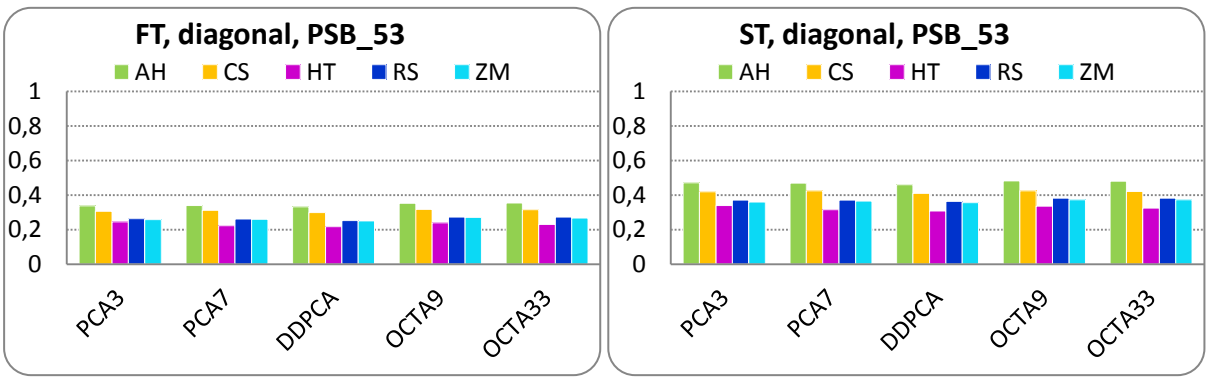



*Figure III.32 PSB_53 database: FT and ST score,* diagonal *matching strategy.*

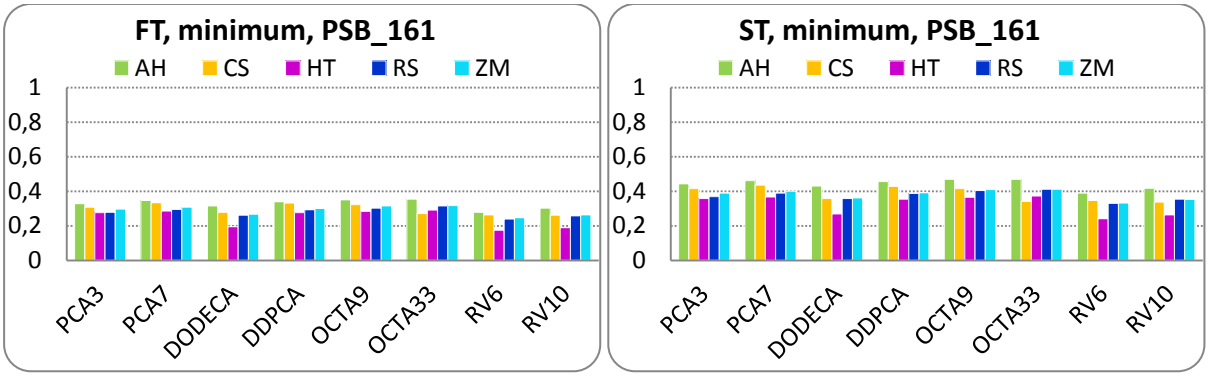



*Figure III.33 PSB_161 database: FT and ST score,* minimum *matching strategy.*

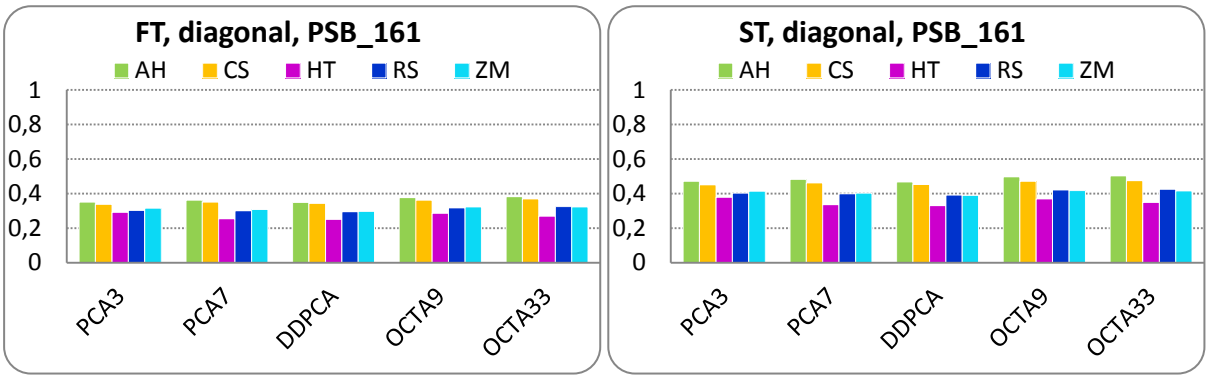



*Figure III.34 PSB_161 database: FT and ST score,* diagonal *matching strategy.*

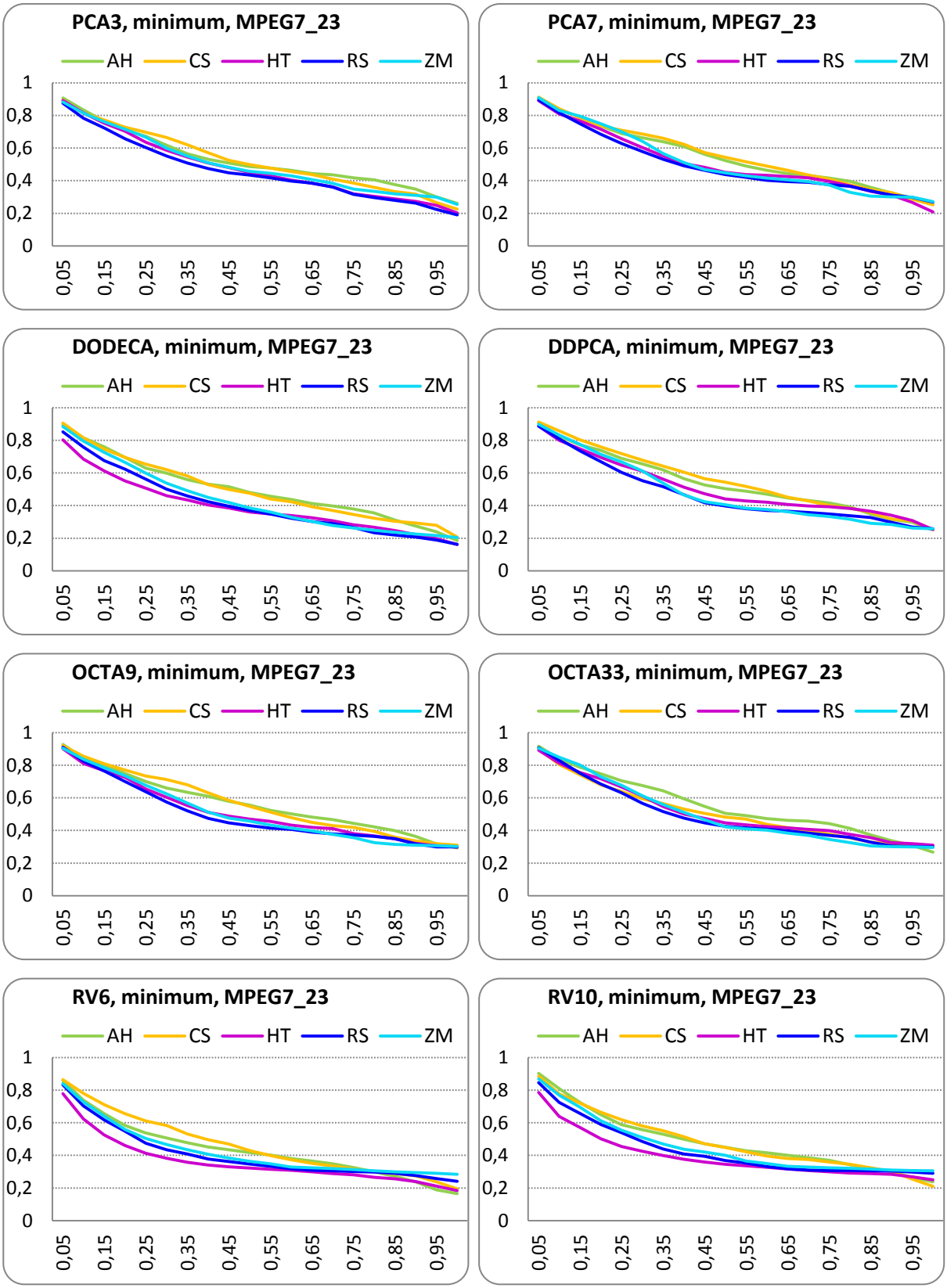



*Figure III.35 MPEG7_23 database: Precision-Recall curves,* minimum *matching strategy.*

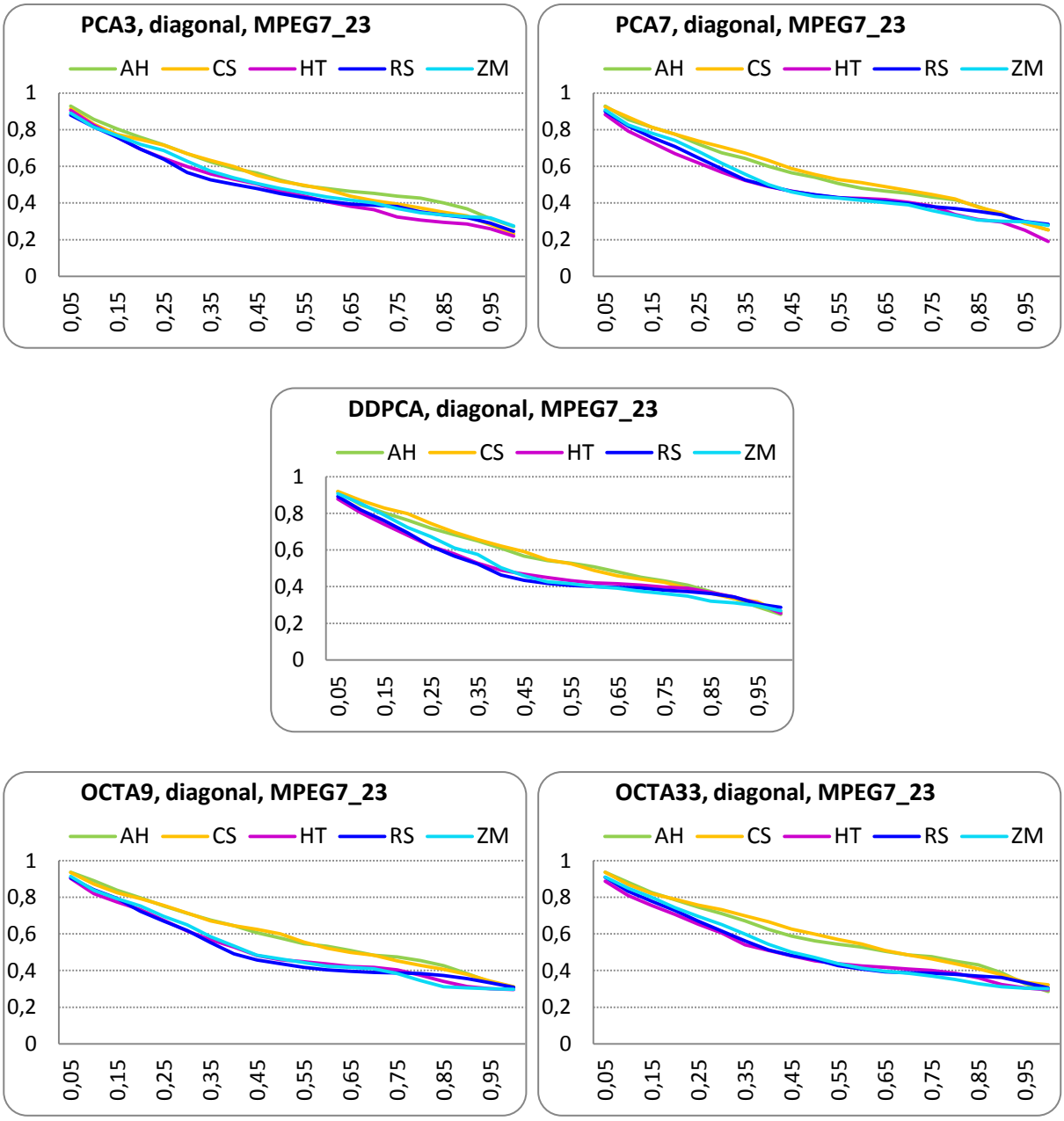



*Figure III.36 MPEG7_23 database: Precision-Recall curves,* diagonal *matching strategy.*

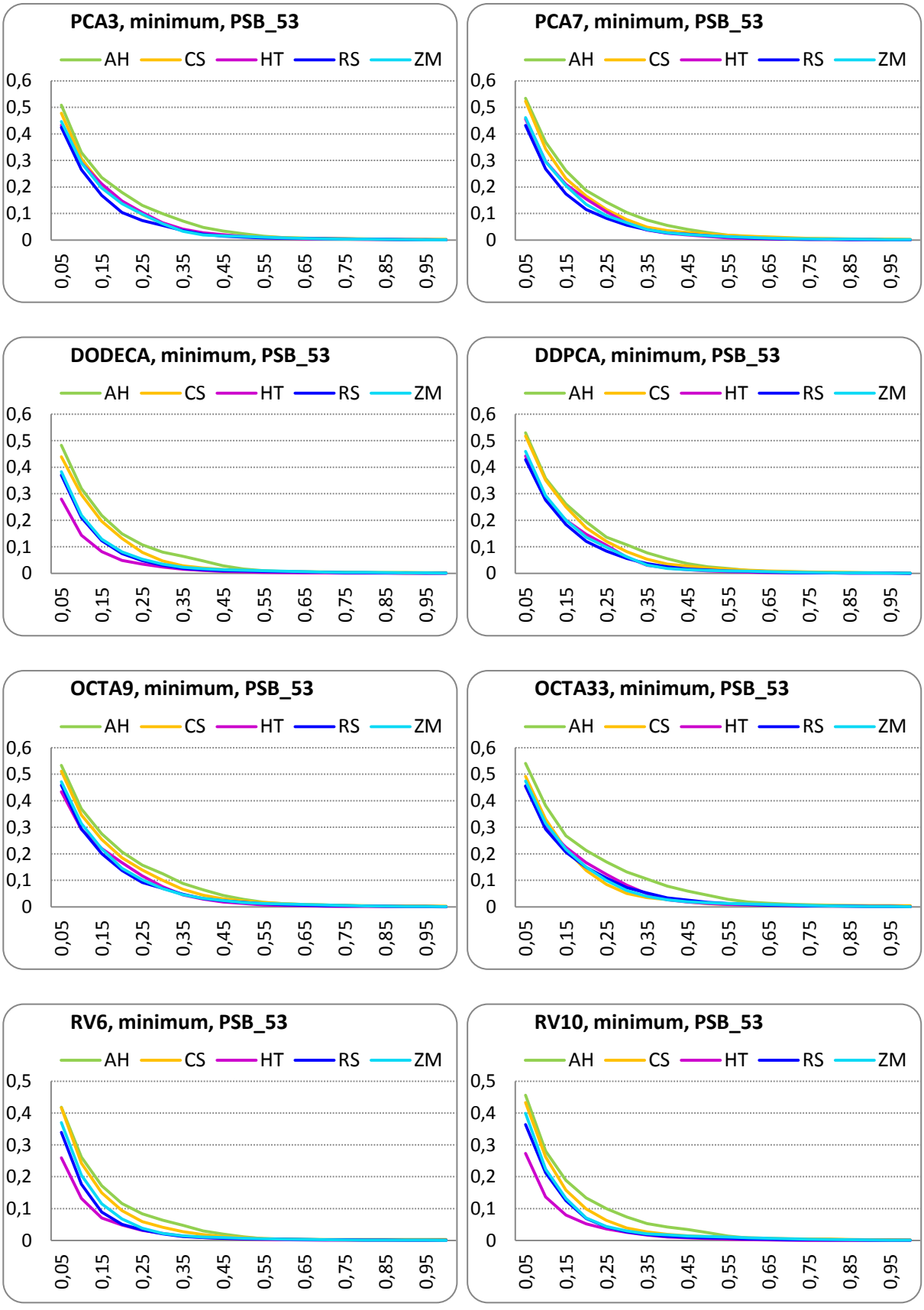



*Figure III.37 PSB_53 database: Precision-Recall curves,* minimum *matching strategy.*

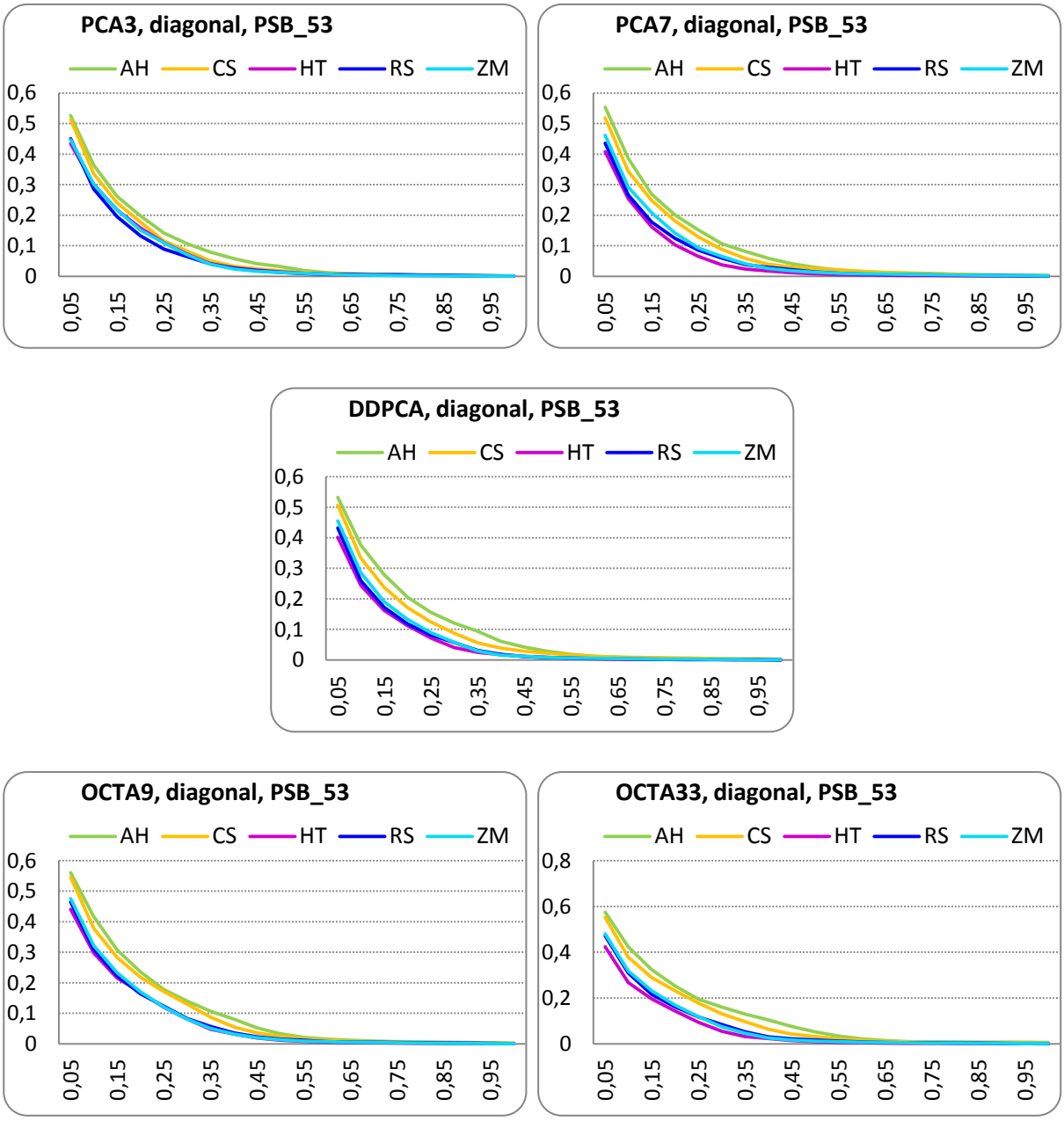



*Figure III.38 PSB_53 database: Precision-Recall curves,* diagonal *matching strategy.*

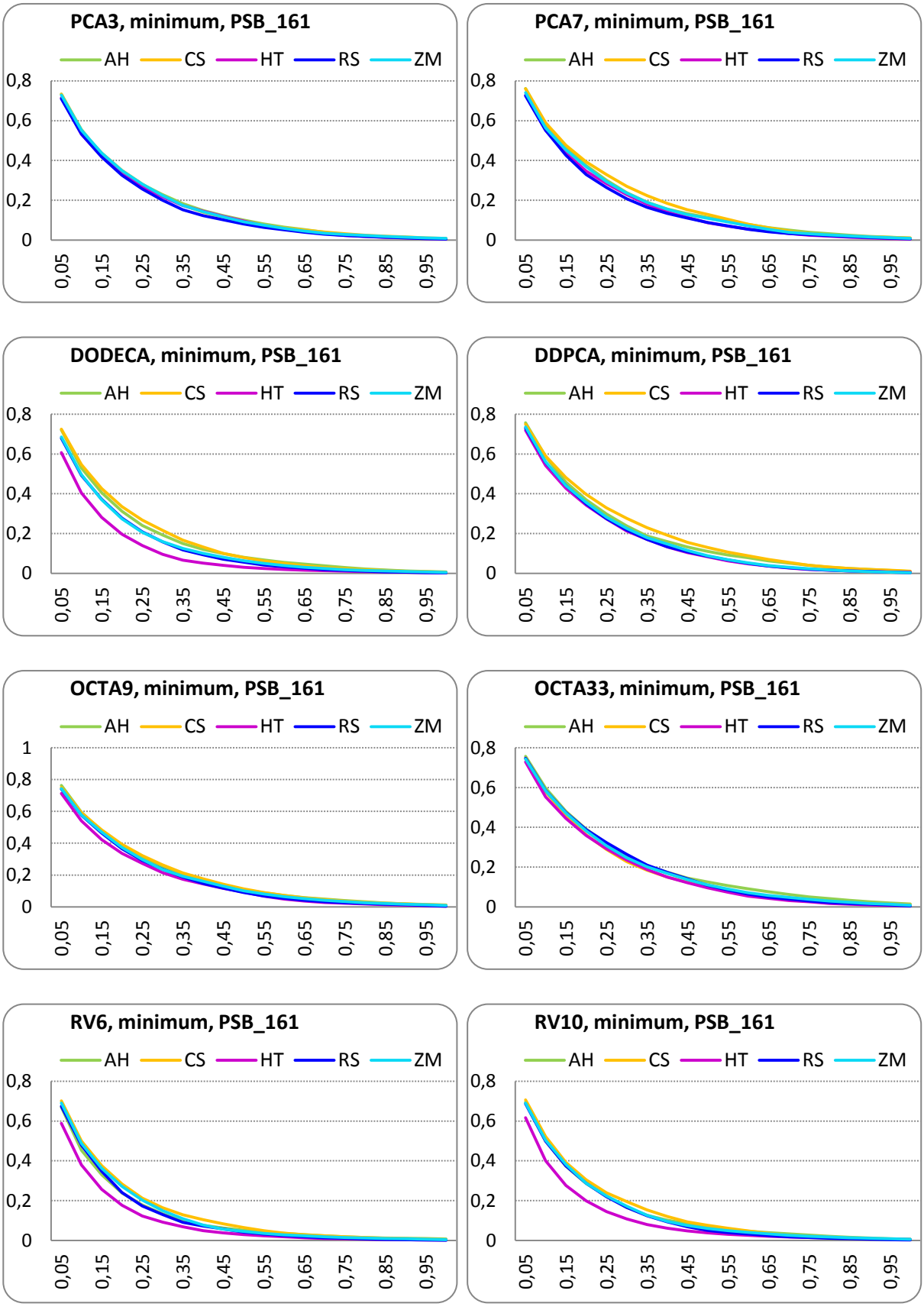



*Figure III.39 PSB_161 database: Precision-Recall curves,* minimum *matching strategy.*

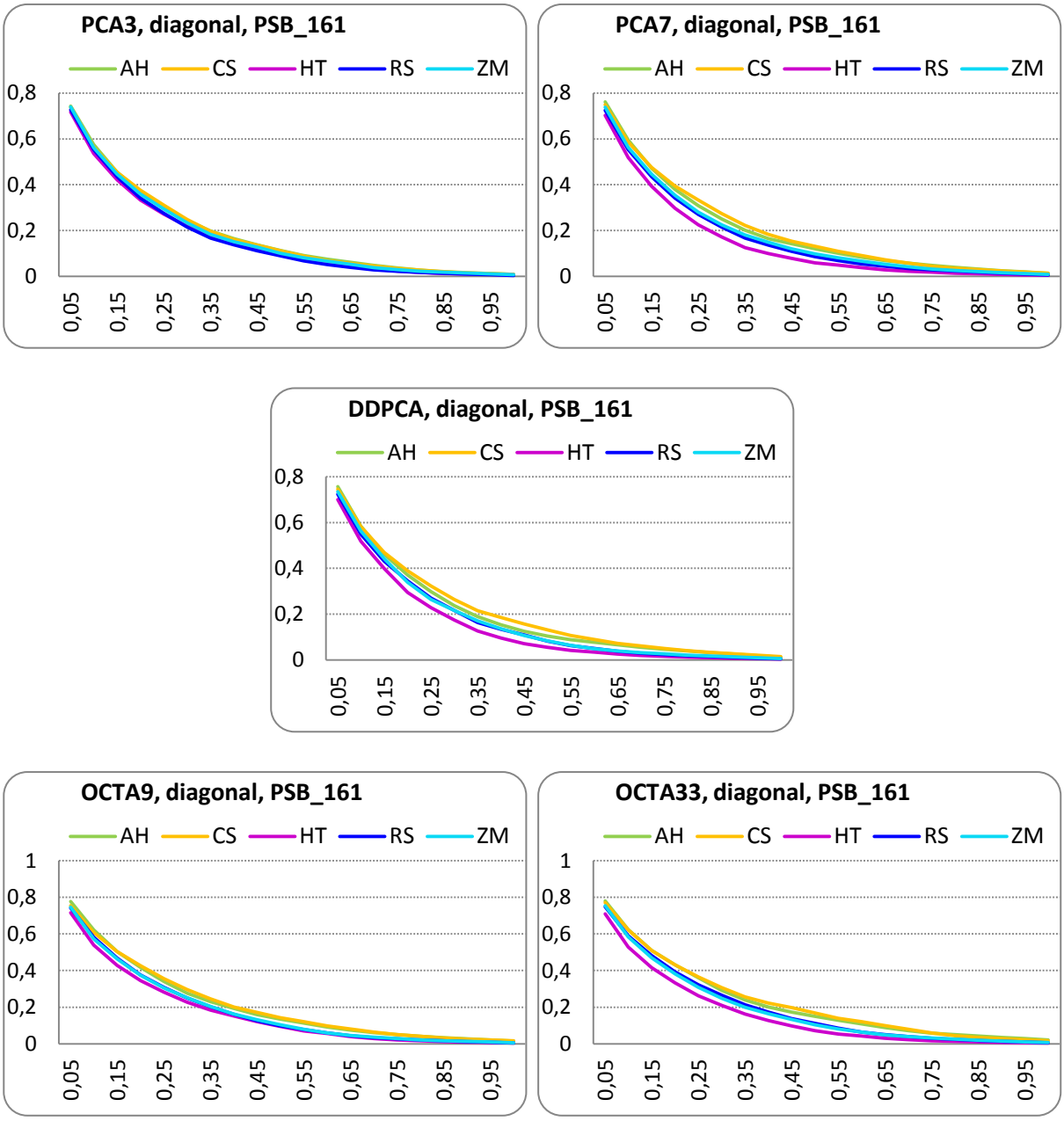



*Figure III.40 PSB_161 database: Precision-Recall curves,* diagonal *matching strategy.*

*Table III.9 MPEG7_23 database: FT and ST score with* minimum *matching strategy*

| FT | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,672** | **0,672** | 0,585 | 0,608 | 0,634 |
| PCA7 | **0,682** | 0,680 | 0,598 | 0,626 | 0,631 |
| DODECA | **0,656** | 0,628 | 0,504 | 0,580 | 0,582 |
| DDPCA | **0,675** | **0,675** | 0,586 | 0,612 | 0,612 |
| OCTA9 | **0,696** | 0,680 | 0,603 | 0,630 | 0,639 |
| OCTA33 | **0,685** | 0,602 | 0,596 | 0,628 | 0,629 |
| RV6 | **0,616** | 0,606 | 0,476 | 0,564 | 0,575 |
| RV10 | **0,648** | 0,588 | 0,499 | 0,584 | 0,592 |

| ST | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,782** | 0,771 | 0,691 | 0,713 | 0,737 |
| PCA7 | **0,785** | 0,766 | 0,686 | 0,733 | 0,736 |
| DODECA | **0,776** | 0,716 | 0,620 | 0,713 | 0,706 |
| DDPCA | **0,779** | 0,767 | 0,673 | 0,719 | 0,716 |
| OCTA9 | **0,802** | 0,752 | 0,689 | 0,734 | 0,734 |
| OCTA33 | **0,796** | 0,689 | 0,675 | 0,740 | 0,731 |
| RV6 | **0,748** | 0,707 | 0,586 | 0,684 | 0,689 |
| RV10 | **0,772** | 0,669 | 0,604 | 0,695 | 0,701 |

*Table III.10 MPEG7_23 database: FT and ST score with* diagonal *matching strategy*

| FT | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,701** | **0,701** | 0,606 | 0,638 | 0,646 |
| PCA7 | 0,699 | **0,718** | 0,580 | 0,638 | 0,634 |
| DDPCA | 0,698 | **0,706** | 0,582 | 0,627 | 0,621 |
| OCTA9 | 0,727 | **0,729** | 0,618 | 0,646 | 0,652 |
| OCTA33 | 0,723 | **0,731** | 0,600 | 0,646 | 0,645 |

| ST | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | 0,808 | **0,811** | 0,700 | 0,733 | 0,739 |
| PCA7 | 0,808 | **0,810** | 0,670 | 0,739 | 0,734 |
| DDPCA | 0,802 | **0,813** | 0,664 | 0,726 | 0,718 |
| OCTA9 | **0,827** | 0,821 | 0,709 | 0,751 | 0,748 |
| OCTA33 | **0,824** | 0,822 | 0,683 | 0,746 | 0,737 |

*Table III.11 PSB_53 database: FT and ST score with* minimum *matching strategy*

| FT | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,321** | 0,284 | 0,235 | 0,249 | 0,254 |
| PCA7 | **0,332** | 0,295 | 0,236 | 0,257 | 0,265 |
| DODECA | **0,317** | 0,251 | 0,189 | 0,242 | 0,242 |
| DDPCA | **0,325** | 0,284 | 0,226 | 0,250 | 0,252 |
| OCTA9 | **0,338** | 0,284 | 0,236 | 0,265 | 0,265 |
| OCTA33 | **0,338** | 0,234 | 0,237 | 0,267 | 0,263 |
| RV6 | **0,300** | 0,252 | 0,170 | 0,227 | 0,225 |
| RV10 | **0,314** | 0,244 | 0,179 | 0,240 | 0,235 |

| ST | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,452** | 0,397 | 0,332 | 0,359 | 0,358 |
| PCA7 | **0,460** | 0,403 | 0,329 | 0,371 | 0,374 |
| DODECA | **0,448** | 0,347 | 0,284 | 0,356 | 0,355 |
| DDPCA | **0,452** | 0,385 | 0,315 | 0,359 | 0,358 |
| OCTA9 | **0,469** | 0,383 | 0,333 | 0,379 | 0,373 |
| OCTA33 | **0,466** | 0,333 | 0,329 | 0,379 | 0,374 |
| RV6 | **0,433** | 0,357 | 0,265 | 0,340 | 0,335 |
| RV10 | **0,447** | 0,342 | 0,277 | 0,351 | 0,346 |

*Table III.12 PSB_53 database: FT and ST score with* diagonal *matching strategy*

| FT | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,337** | 0,305 | 0,244 | 0,263 | 0,259 |
| PCA7 | **0,340** | 0,312 | 0,223 | 0,262 | 0,260 |
| DDPCA | **0,332** | 0,299 | 0,218 | 0,252 | 0,251 |
| OCTA9 | **0,353** | 0,317 | 0,240 | 0,272 | 0,270 |
| OCTA33 | **0,354** | 0,315 | 0,229 | 0,272 | 0,267 |

| ST | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,470** | 0,420 | 0,340 | 0,371 | 0,360 |
| PCA7 | **0,469** | 0,425 | 0,316 | 0,372 | 0,365 |
| DDPCA | **0,460** | 0,409 | 0,308 | 0,363 | 0,355 |
| OCTA9 | **0,482** | 0,425 | 0,336 | 0,383 | 0,373 |
| OCTA33 | **0,481** | 0,420 | 0,324 | 0,381 | 0,373 |

*Table III.13 PSB_161 database: FT and ST score with* minimum *matching strategy*

| FT | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,327** | 0,308 | 0,277 | 0,279 | 0,296 |
| PCA7 | **0,346** | 0,335 | 0,286 | 0,294 | 0,308 |
| DODECA | **0,314** | 0,279 | 0,194 | 0,262 | 0,267 |
| DDPCA | **0,340** | 0,331 | 0,275 | 0,293 | 0,300 |
| OCTA9 | **0,350** | 0,323 | 0,284 | 0,303 | 0,316 |
| OCTA33 | **0,354** | 0,271 | 0,292 | 0,315 | 0,316 |
| RV6 | **0,279** | 0,264 | 0,174 | 0,239 | 0,246 |
| RV10 | **0,302** | 0,262 | 0,190 | 0,258 | 0,264 |

| ST | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,444** | 0,415 | 0,358 | 0,369 | 0,389 |
| PCA7 | **0,462** | 0,436 | 0,368 | 0,389 | 0,401 |
| DODECA | **0,429** | 0,358 | 0,269 | 0,358 | 0,361 |
| DDPCA | **0,456** | 0,429 | 0,355 | 0,387 | 0,391 |
| OCTA9 | **0,469** | 0,415 | 0,365 | 0,404 | 0,409 |
| OCTA33 | **0,469** | 0,340 | 0,372 | 0,411 | 0,412 |
| RV6 | **0,389** | 0,346 | 0,241 | 0,330 | 0,333 |
| RV10 | **0,417** | 0,337 | 0,264 | 0,353 | 0,352 |

*Table III.14 PSB_161 database: FT and ST score with* diagonal *matching strategy*

| FT | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,350** | 0,338 | 0,290 | 0,302 | 0,314 |
| PCA7 | **0,361** | 0,351 | 0,254 | 0,300 | 0,308 |
| DDPCA | **0,349** | 0,343 | 0,251 | 0,295 | 0,297 |
| OCTA9 | **0,376** | 0,362 | 0,285 | 0,318 | 0,322 |
| OCTA33 | **0,382** | 0,368 | 0,269 | 0,325 | 0,322 |

| ST | AH | CS | HT | RS | ZM |
|---|---|---|---|---|---|
| PCA3 | **0,472** | 0,451 | 0,378 | 0,403 | 0,413 |
| PCA7 | **0,482** | 0,463 | 0,335 | 0,399 | 0,402 |
| DDPCA | **0,467** | 0,452 | 0,330 | 0,391 | 0,389 |
| OCTA9 | **0,497** | 0,471 | 0,369 | 0,421 | 0,417 |
| OCTA33 | **0,502** | 0,474 | 0,349 | 0,425 | 0,415 |

## III.7. CONCLUSIONS

In this chapter we have tackled the issue of view-based 3D model retrieval. First, we have described the 2D/3D retrieval framework adopted and detailed both the various 2D shape descriptors and viewing angle selection strategies retained. A representative viewing angle selection approach employing a modified k-means clustering algorithm was also proposed. In addition, a novel contour-based 2D shape descriptor, so-called angular histogram (AH), able to capture both local and global shape characteristics has been introduced.

The first part of our experiments concerns an analysis of existing, categorized 3D model repositories in terms of intra and inter-class variability/separability properties. Both MPEG-7 (with 23 categories) and Princeton (with two levels of detail, corresponding to 53 and 161 categories) databases have been retained. An evaluation protocol, relying on a set of objective criteria has been introduced. The experimental analysis shows, without surprise, that the categories of MPEG7_23 database are better defined in the 2D/3D indexing space, and therefore easier to separate. We have also observed that PSB_53 presents classes which are more generic than those of PSB_161, which leads to more ambiguity and less separability. The database analysis has also shown that contour-based descriptors present similar behaviours for all classes, while the region-based descriptors are particularly appropriate for a sub-set of object categories.

An objective evaluation of the various 2D shape descriptors and viewing angle selection strategies has then been carried out on the same 3D repositories. The obtained results confirm the

database analysis. Thus, the best retrieval scores were obtained on the MPEG7_23 database, with *FT* up to 72.3% and *ST* up to 82.7%. On PSB_53 and PSB_161 databases we have achieved respectively 35.4% and 38.2% in terms of *FT* and 48.2% and 50.2% in terms of *ST*.

Concerning the viewing angle selection strategies, the experimental evaluation shows that they have limited impact (within a range of 3.6% of FT score variation from PCA3 to OCTA33) on the 3D model retrieval performances. Moreover, when passing from 9 views (OCTA9 strategy) to 33 (OCTA 33 strategy), the performances can even degrade.

A final major conclusion is related to the performances of the descriptors retained for evaluation. Thus, the contour-based approaches clearly outperform the region-based techniques, whatever the database and viewing angle selection strategies considered. The highest retrieval scores have been obtained for the proposed AH descriptor, for all the databases considered.

The experimental evaluation of view-based indexing performance in 3D model retrieval allows a first comparative analysis of the different elements that compose the 2D/3D indexing. In the following chapter we present how the view-based indexing methods can be exploited for 2D object recognition purposes.

# IV. 2D OBJECT CLASSIFICATION

**Abstract.** *In this chapter, we tackle the issue of object classification. The view-based indexing methods presented previously are here employed to allow semantic inference between 3D and 2D content. The underlying principle consists in exploiting the a priori knowledge contained in classified 3D models and to transfer it, with the help of view-based indexing, to unknown 2D objects. Such methods can be applied to both still objects (SO, i.e., objects extracted from still images) and video objects (VO, i.e., objects extracted from videos and composed of several instances).*

*After presenting an overview of the state of the art, we introduce the proposed 2D object classification framework. The proposed approach ensures fast categorization and also allows the effective combination of several indexing techniques.*

*In order to experimentally evaluate the proposed method, we have created several test datasets, including objects extracted from images and videos, but also synthetic views generated by 3D model projection. The experiments prove that the proposed framework leads, in the case of still objects, to recognition rates of up to 74%. Superior recognition results are obtained for video objects, where the same scores are up to 87.5%.*

**Keywords:** *object classification; 2D/3D inference; 2D/3D indexing; similarity measure; recognition framework;*

## IV.1. INTRODUCTION

In this chapter, we address the issue of object classification. The objective here is to automatically determine the semantic meaning of objects present in images and/or videos.

Until recently, semantic labelling approaches were exclusively based on keywords. However, the linguistic barriers represent an important drawback of such approaches. Also, a prior, manual annotation is required, which is a tedious and highly subjective process.

The early automatic methods employed correlation-based template matching [Lewis95] for recognition purposes. Such approaches have rapidly shown their limitations when dealing with strong changes of the object's appearance, such as those generated by 2D or 3D transforms, scaling and variations of the illumination.

Recent research on automatic object classification is mainly based on machine learning (ML) techniques [Mithchell97, Xue09]. The goal of ML algorithms is to automatically learn to recognize complex structures using a set of examples included in a so-called training set. Two families of approaches are available: supervised and unsupervised methods. In the first case, the required training set is labelled with both positive and negative examples, while unsupervised ML methods allow unlabelled training data.

In the case of supervised ML, a training set of labelled objects divided into $N$ categories is supposed to be available. Based on such a training set, the objective is to determine a function which best discriminates between the $N$ classes. Once the function is defined, it can be applied for each unknown object in order to determine to which category the object belongs. The supervised ML techniques may be highly accurate [Deselaers10]. However, it may happen that the function is too appropriate for the training set and thus inadequate for new objects [Pados94]. This phenomenon is known as over-fitting and represents one of the main drawbacks of supervised ML approaches. A second limitation is the requirement of sufficiently large training sets containing labelled data.

Concerning the unsupervised approaches, some popular methods are K-means, mixture methods and K-nearest neighbours… For some examples, the reader is invited to refer to [Weber00, Fergus03, Torralba08, Makadia08]. However, in terms of performances, the unsupervised methods are less accurate than the supervised machine learning methods.

Generally, when a large number of classes has to be considered, the ML approaches need to exploit a large set of features. Thus, in such cases the computational complexity becomes intractable [Li06]. Also, if we take into consideration that an object may change its appearance due to the pose variation, then the training set should include not merely different examples of objects from each class, but also different instances of each object, corresponding to various poses.

In our work, we consider a different approach, consisting of introducing in the recognition process some *a priori* 3D information, with the help of existing 3D models. The goal here is to

overcome the sensitivity to the object's pose of ML methods by exploiting a set of examples composed of 3D models. In order to allow the matching between 2D and 3D content, the 2D/3D shape indexing methods presented in Section III.2 are used.

This chapter first presents the state of the art methods in the field of 3D model-based 2D object classification. The proposed SO and VO classification frameworks are further described and experimentally evaluated.

## IV.2. RELATED WORK

Within the context of various 2D object recognition applications, there exist two main families of approaches, corresponding to different meaning of the term *recognition*. In a first case *recognition* refers to *classification* (*categorization*). In this case, the goal is to assign semantic labels, corresponding to a set of pre-defined categories, to the various objects that may appear in images or videos. The second family of methods aims to detect if some classes of objects are appearing in the image/video content, and to determine their position (*e.g.*, detecting pedestrians for video surveillance purpose). We speak in this case about *object detection*.

The two applications (*i.e.,* classification and detection) are in a certain sense complementary, since the first one searches for the semantic class of a given 2D object, while the second one searches for 2D objects knowing their semantic class. Even if the detection methods can be used for classification purposes, they are not specifically designed for such a purpose. In addition, classification methods aim at dealing with a large variety of semantic categories, while detection approaches are more particularly conceived to recognize objects from certain classes (*e.g.,* cars, pedestrians, guns...).

Both detection and classification techniques can exploit 3D representations. Here again there exists a certain ambiguity, because the term *3D model* can have different meanings:

- An object represented in the virtual space with the help of 3D graphical elements (*e.g.*, 3D mesh) [Toshev09, Liebelt08, Gupta08]. Such models can be obtained by scanning real objects or by modelling them with the help of 3D computer graphics software.
- A compact representation of the visual features (*e.g.*, shape, appearance...) extracted from several images representing different perspectives/views of a 3D object (real or synthetic) or several objects from the same semantic category [Ferrari04a, Hoeim07, Thomas06, Savarese07].

The large majority of 3D model-based detection methods is designed for video surveillance. Thus, most applications are dedicated to the identification of cars [Gupta08, Patterson08, Hoeim07, Kushal07, Leotta11, Hodlmoser12...], motorcycles [Thomas06], bicycles, pedestrians or combinations of them [Liebelt08, Schels11].

In [Ferrari04], authors integrate 3D information in the recognition process. The 3D representation of an object, generated from a set of different views, is composed of so-called *region-tracks*. Each track is defined by a circular region together with all regions from other views that have been matched with it. The region matching process is achieved with the help of an affine invariant interest point matcher [Matas02, Mikolajczyk02]. This makes it possible to put into correspondence pairs of regions with the help of a 2D affine transformation. Moreover, such a transformation induces a measure of similarity between the two matched regions.

During the recognition process, the test image is compared to each view of the object. Each comparison generates a set of matches which are further partitioned into *groups of aggregated matches* (GAMs). Each GAM includes regions which can be mapped onto each other by a geometric affine transformation. As the views are related through the region-tracks, the geometric consistency of the GAMs configurations can be evaluated and thus it becomes possible to eliminate potential mismatches. A genetic algorithm is further employed to determine the most consistent GAM configuration, whose matches cover the 2D object as completely as possible.

The region-tracks are further exploited by [Thomas06] in order to improve the Implicit Shape Model (ISM) proposed in [Liebe04]. The ISM builds, for each semantic class, a codebook composed of clusters of local features with similar appearance as well as their spatial distribution across several examples. Each codebook corresponds to a semantic class and to a viewpoint. In order to allow multi-view recognition, the codebooks are connected with the help of region tracks [Ferrari04], trained separately on the same dataset. The recognition process employs a voting procedure. The use of connected codebooks allows transferring votes across views with different poses, which improves the recognition performances, notably in the case of articulated objects.

The principle of linking together object parts from different viewing angles has also been exploited in [Savarese07]. First, the saliency detector [Kadir01] and the SIFT descriptor [Lowe99] are applied in order to characterize local appearance features. The local features are further grouped into regions (parts) that are consistent across images in both appearance and geometry. The 3D representation is finally obtained by linking together parts from a so-called canonical subset containing for each region its most frontal view. Two canonical parts are connected only if they are both visible in the corresponding frontal views. Each link is also characterized by an explicit homographic relationship between the connected regions.

In the recognition stage, the occurrences of each canonical part are searched across various pixel locations, scales and orientations. Homographic relationships are computed between pairs of matched regions and, by applying an optimization algorithm, a dissimilarity score can be determined. The object's category is then determined as the one of the model that yields to the lowest dissimilarity score. The experimental evaluation presented in [Savarese07] proves that the proposed algorithm outperforms the one introduced in [Thomas06].

An important contribution to the field of object recognition has been recently made by Liebelt *et al.* [Liebelt08, Liebelt10, Schels11, Schels12]. The objective of such methods is twofold: (1) viewpoint-independent object detection and (2) 3D pose estimation. In contrast to the

previously presented approaches [Ferrari04, Thomas06, Savarese07], synthetic, textured 3D models are here exploited.

In [Liebelt08], authors introduce the term of *3D feature map*, defined as a set of annotated local features extracted from several views of a given 3D model. In order to obtain such a representation, the models are first normalized in size and position. Further, 324 views are rendered in 2D by varying the camera's parameters (*i.e.*, azimuth, elevation and distance). For each image, the local features are obtained by computing the SURF descriptor [Bay06] of each point of interest (POI) identified with the Fast Hessian detector [Bay06]. The annotated features are obtained by associating to each SURF descriptor the corresponding 3D model position. A discriminative filtering is performed in order to select feature that are discriminant enough for the object category and robust to viewpoint and background changes. Finally, a visual codebook with $k$=2000 elements is obtained by clustering the retained features with the k-means algorithm [Silverman02].

During the detection phase, local features are extracted from the image and matched with the 5 closest codebook entries. The desired object is detected by a voting procedure involving the features included in the 5 selected clusters. The experimental results obtained are promising but somehow limited, since solely two object categories (corresponding to cars and motorbikes) are considered.

Liebelt *et al*. propose further in [Liebelt10] to keep only the geometry information of the 3D models and to exploit the appearance information contained in real-life images. The proposed method requires that, for each training image, the 2D bounding box and the viewpoint (*i.e*., 3D camera parameters) of the object are available. The synthetic 3D models are rendered in 2D by varying the camera parameters, resulting in a set of 240 synthetic projection images. The bounding boxes of both real and synthetic images are subdivided into a regular grid, where each grid block corresponds to a part of the object. For each grid block associated to a real image, the geometry information is extracted from the block with the same position on the corresponding synthetic image (*i.e.*, the synthetic image presenting the most similar viewpoint). The 3D geometry associated to each grid block is obtained as follows. Each pixel inside a part region corresponds to a 3D position on the surface of the model. The set of 3D positions associated to a grid block is modelled by a mixture of Gaussians [Reynolds07]. The parameters of the Gaussian Mixture Model (GMM) are used to encode the geometry features of the grid block. The local appearance features are described with the DAISY descriptor [Tola10]. In the training phase, a codebook of DAISY descriptors is generated. The codebook is further used to identify regions which present a high likelihood of containing the object of interest. A SVM classifier [Vapnik95] is learned for each object region, under each considered viewpoint.

The recognition process starts by a pre-detection step. A sliding-window detection and a subsequent mean-shift mode estimation are employed to localize potential regions of interest in the image. Further, the most likely viewpoints are determined through a region voting procedure. The object's position and pose are finally determined with the help of an evaluation score which measures the consistency between the detected regions and the learnt 3D geometry model. The experimental evaluation proposed by the authors proves that the average precision of the object

detection process is significantly increased (by about 6-7%) when object detection and pose estimation are simultaneously performed.

Part detectors are also exploited in [Schels11]. In contrast with [Liebelt10], here solely synthetic images are considered for training. The parts (*i.e.*, regions of the object) are detected as local regions presenting high gradients across training images. The algorithm uses two sets of images for learning, including part examples and viewpoint examples, both representing rendered views of 3D models. The appearance of each part is described with the help of the histogram of gradients (HOG) descriptor [Dalal05]. A separate linear SVM classifier [Vapnik95] is trained for each viewpoint and for each part. Based on the trained part detectors, a spatial layout model is generated for each viewpoint. The images from the second set (*i.e.*, the viewpoint examples) are described through a spatial pyramid which accumulates responses for each part detector. A second, more powerful classifier (a non-linear SVM) is trained on the set of spatial pyramids.

For each image, the detection is performed in two steps. First, possible regions of interest (hypotheses) are determined with the help of the spatial layout model. Further, the spatial pyramid representation is generated and the second classifier is employed to refine the previously detected hypotheses. The experimental evaluation proposed shows that the method outperforms the [Liebelt10] approach.

The above-presented methods lead to interesting results, which are achieved at the end of a complex process, involving a high number of SVM classifiers. In order to reduce the computational complexity, the same authors propose in a more recent work [Schels12] to reduce the number of parts considered. First, the parts of synthetic images, encoded by HOG features, are grouped with the help of an unsupervised clustering procedure. Further, the parts are ranked as more or less informative, based on the number of models and viewpoints where they occur. The most suited parts for detection are finally determined through an entropy-based selection process. The rest of the approach is similar to the one presented in [Schels11], except that only the most informative parts are employed for the spatial representation.

The experimental evaluation presented in [Schels12] demonstrates that the detection is improved by eliminating non-informative parts. Moreover, authors also compared their approach with state-of-the-art detection methods [Glasner11, Su09, Sun09] trained on images of real objects. The comparison shows that for a similar amount of training information (*i.e.*, number of objects and views per object), synthetically-trained approaches lead to similar results.

In [Heisele09], solely the shape information is exploited, from textureless synthetic 3D models. Only five 3D objects, each one representing a different category, are here considered. The models are rendered in 60,000 images by randomly varying the camera location and orientation and the lighting conditions.

The first set of experiments aims to discriminate between two 3D objects, whatever their orientation. Histograms of gradients [Dalal05, Lowe04] are used to encode the appearance features and two classifiers are tested: Support Vector Machine (SVM) [Vapnik95] and nearest neighbour (NN) [Cover67]. The number of images employed to learn each category (represented by only one 3D model) varies from 2,000 to 40,000.

The second set of experiments deals with pose-invariant object detection. Here, the test images are obtained from 642 viewing positions corresponding to the vertices of a geodesic grid placed around the 3D object.

Even if the proposed analysis is minimalistic (only five 3D models are exploited), some interesting conclusions are presented. Thus, the experiments show that, even for a simple task such as distinguishing between two objects, very large training sets have to be used in order to achieve high accuracy. Since finding a large amount of training images is a tedious task, employing synthetic 3D models can be a suitable solution.

[Toshev09] extrapolates the 3D model-based recognition problem to video objects. The first stage of their algorithm consists in detecting moving objects. Feature tracking [Shi94] and motion-based clustering are combined in order to separate the objects and the background. A set of several instances of the object (corresponding to different frames) is obtained at the end of this stage.

The 3D models used in the recognition process are described by a so-called *view graph*, determined as follows. First, 500 silhouettes are generated for each model from approximately uniformly distributed viewing angles. Each silhouette boundary is described with the help of shape contexts [Belongie02]. Further, their number is reduced to 20 views per model with the help of the k-medoids clustering technique. Finally, the selected silhouettes are organized as a graph covering the entire viewing sphere.

In the recognition process, each video object is compared to all 3D models. The most similar 3D model gives the semantic class of the object and the pose estimation. The 3D/2D matching procedure takes into account the shape similarity (through shape contexts features) while maintaining motion coherence over time (*i.e.*, the silhouettes corresponding to successive frames should present limited changes of the viewing angle). The proposed algorithm was tested on a database of 42 videos, containing 3 different classes: cars, airplanes and helicopters. The recognition process involves 260 PSB 3D models (*cf.* Section III.5.2, 52 categories but only 5 models per class). The authors reported achieving an accuracy of 83% (80% for cars, 91.7% for airplanes and 80% for helicopters).

Another video object recognition approach was proposed very recently in [Hodlmoser12]. The algorithm is designed to detect vehicles in videos, to determine their model and their pose. The recognition process exploits the information contained in textureless 3D vehicle objects. The images representing the rendered 3D model contain not only the external contour, but also internal edges. The unknown objects and their bounding box are first detected in each frame with a state-of-the-art approach [Felzenszwalb10]. Further, the object is described by its contours, identified with the Canny edge detector [Canny86]. The object is further compared to several views of each model, obtained from viewpoints close to the initial pose estimation, and the most similar models are identified. The fast directional chamfer matching [Liu10] was here employed to compare the contours of two objects. Further, the best matching model through the entire video sequence is determined using a Markov random field (MRF) [Li95]. The experimental evaluation proposed in [Hodlmoser12] is performed on a database with 8 videos, representing 6 different car models

(Chevrolet Silverado 2500HD, Chrysler PT Cruiser, VW Beetle, Toyota RAV4, Chevrolet Blazer and Skoda Fabia). The reported results proved that the above presented approach outperforms the one proposed in [Toshev09].

The analysis of the literature shows that a large majority of object recognition approaches are designed for object detection and pose estimation, frequently exploited for video surveillance purposes. However, for such applications the number of categories to be detected is relatively low. The methods in [Thomas06, Liebelt08, Liebelt10, Schels11, Schels12, Toshev09] are trained for only 2 or 3 categories. In [Heisele09], their number increases to 5; however, their analysis remains minimalistic, as each class contains only one object whose projections are employed for both training and tests. In [Savarese07], authors deal with a larger number of categories (*i.e.*, 10), representing everyday objects: cars, staplers, irons, shoes, monitors, computer mice, heads, bicycles, toasters and cellphones. However, their approach requires that each training object to be captured from 7 different viewpoints, which can be a difficult constraint in creating the learning database. The number of views associated to each training object is even higher in [Thomas06], where the target is to acquire 16 images per object. However, in some cases it is impossible to capture it from all the necessary perspectives (*e.g.*, capturing an airplane from many different perspectives). Such drawbacks can be overcome by using synthetic 3D models, which allows automatically generating as many views as needed, from arbitrary viewing angles.

In order to overcome the limitations of existing approaches, in our work we have considered the inclusion of synthetic 3D models in the recognition process. The proposed method, detailed in the next section relies exclusively on the shape information, as the geometry is, in general, the most discriminant feature of real objects.

In our work we address the classification issue, *i.e.*, the objective is to determine the semantic category of a given object. Therefore, from now on throughout this chapter, the term *recognition* will refer to *classification*.

The main contributions of our work can be summarized as follows:

- The proposed approach is able to deal with a large number of categories. In our current implementation, the system disposes of up to 161 categories of objects and was tested on up to 23 classes.
- The classification method is able to deal with both still and video objects.
- We provide a comparative study of different 2D/3D indexing methods that can be involved in the recognition process. The influence on the recognition performance of the viewing angle selection strategy, 2D shape descriptors and number of instances per video object is analyzed here.
- The proposed method allows real-time category recognition.

The proposed classification method is presented in the following section.

## IV.3. PROPOSED METHOD

The proposed 2D object classification method exploits the inference between categorized 3D models and 2D content. Such an inference is possible through the use of view-based indexing techniques, as those presented in Section III.2.

The principle consists in exploiting the *a priori* information contained in the categorized 3D model data set in order to transfer the semantic labels from such models to unknown 2D content.

The objects to be classified can be present in images or in videos. In the first case, the unknown content represents a still object (SO), while in the second case a video object (VO).

The two classification processes are very similar and exploits the same 2D/3D semantic inference techniques. Let us start by presenting the still object classification approach.

### IV.3.1. Still object classification

The still object recognition process is illustrated in Figure IV.1. In order to classify an unknown 2D shape, the system needs to utilise a categorized 3D model repository. Each 3D model is further described using view-based shape indexing, which allows comparison between 3D and 2D content.

In the 2D/3D indexing stage, a set of 2D views is generated for each model and a 2D shape descriptor is associated to each view. This stage is performed only once, in an offline phase, and the resulting descriptors are further stored and used for recognition purposes.

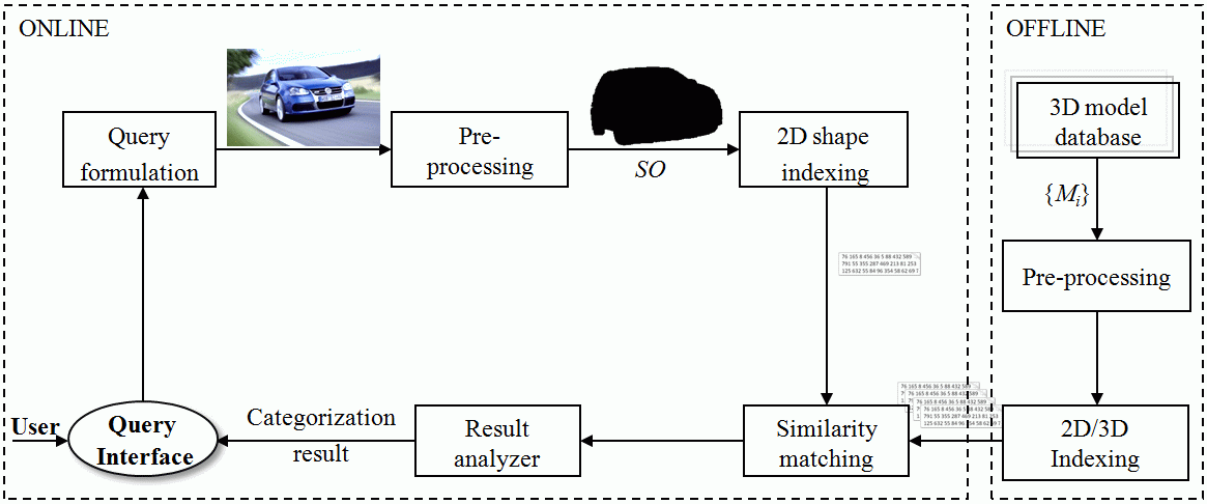



*Figure IV.1 Still object recognition framework.*

The system input is an unknown 2D object that needs to be identified. If the object of interest is not extracted from the image, a pre-processing step is required in order to obtain a binary image representing the shape of the unknown object. The binary image may be obtained with the help of some semi-automatic segmentation methods, as those presented in Chapter V.

Furthermore, the 2D shape descriptor is extracted for the current unknown image. In order to allow the comparison between 3D and 2D content, the same descriptor needs to be used in the 2D shape indexing as in the 2D/3D indexing process.

Next, in the similarity matching stage, the distance $d(SO,M)$ between a still object $SO$ and a 3D model $M$ is computed. It is defined as the best match between the still object and the projections of the 3D model:

$$d(SO, M) = \min_i d\big(SO, P_i(M)\big), \qquad (IV.1)$$

where:

- $P_i(M)$ represents the $i^{th}$ projections of the model $M$;
- $d\big(SO, P_i(M)\big)$ represents the similarity metric associated with the considered 2D shape descriptor; it is computed between the feature vector associated with the shape of the 2D object $SO$ and the one corresponding to the $i^{th}$ silhouette of the 3D model $M$.

The last module of the recognition framework is represented by the result analyzer. It examines the similarity between the query and the categorized 3D models in order to determine which are the most probable categories that may fit the unknown image.

In this analysis, we make the underlying assumption that among the top retrieved 3D models a large number should belong to the semantic category of the query. In order to determine the query's category based on the similarity between 3D and 2D content, the result analysis includes the following steps:

- Sort the models by decreasing order of similarity;
- Keep the $N_{TRM}$ top retrieved models;
- Determine the $N_{MRC}$ most represented categories among the first $N_{TRM}$ models;
- Present the $N_{MRC}$ categories as response to that query.

The still object recognition framework is extended to video objects, as described in the following section.

### IV.3.2. Video object classification

The term *video object* (VO) refers to the set of various instances $\{I_i(VO)\}$ of an object within a video. The goal of the video object recognition framework is to associate semantic labels to the objects present in a video.

Figure IV.2 represents an overview of the video object recognition framework. As in the case of still images, a categorized 3D model database is supposed to be available and described with the help of 2D/3D indexing techniques.

The input is the unknown object present in the given video $V$, represented through a subset of $N_F$ frames $\{F_i(V)\}$, including the object of interest in different poses. The frame selection may be performed by considering a frame-clustering algorithm such as those introduced in [Tapu11], [Rasheed05].

The object of interest is further segmented from each selected frame, resulting in a set of $N_I = N_F$ binary images, representing the instances $I_i(VO)$ of the video object. Each instance of the video object is described using a 2D shape descriptor (according to the 2D/3D indexing methods employed for describing the 3D model repository).

The similarity measure $d(VO, M)$ between a video object and a 3D model is defined as the sum of individual distances between each instance $I_i(VO)$ of the video object and the 3D model $M$:

$$d(VO, M) = \sum_{i=1}^{N_I} d(I_i(VO), M), \qquad (IV.2)$$

with

$$d(I_i(VO), M) = \min_j d\left(I_i(VO), P_j(M)\right). \qquad (IV.3)$$

Finally, the result analyzer is applied to the distances determined in Equation IV.3 in order to establish which are the most probable categories that fit the input video object.

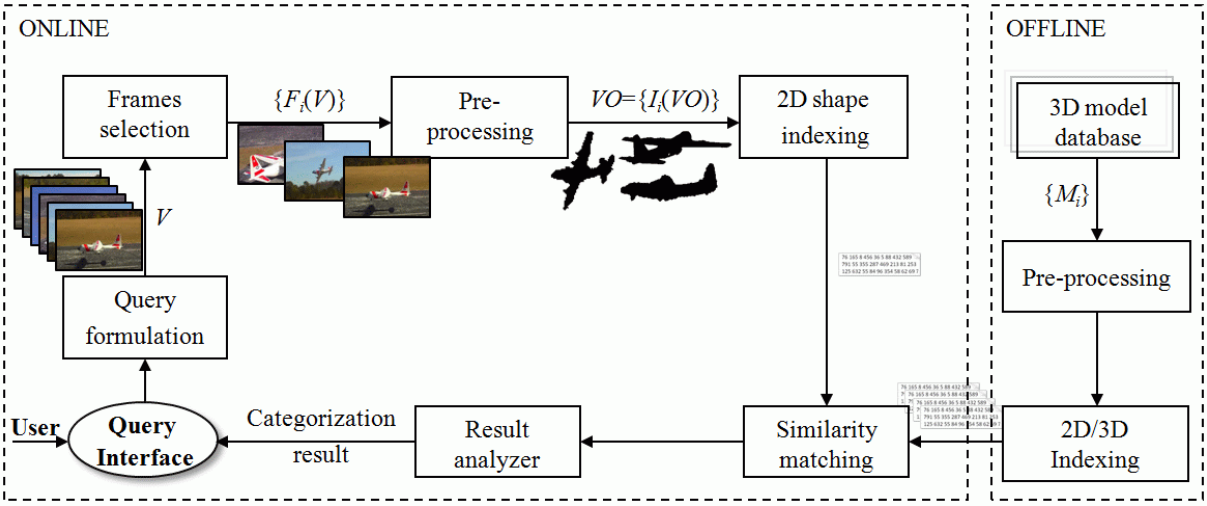



*Figure IV.2 Video object recognition framework.*

Both still and video object recognition systems allow combining several 2D/3D indexing methods. The framework is the same as in the case of individual 2D/3D indexing methods, excepting the result analyzer step. The query is successively compared with the 3D models using each considered method. Next, for each 2D/3D indexing approach $Ai$, the $N_{TRM,Ai}$ top retrieved models are determined. Finally, the models found in the top retrieved positions for each method $Ai$ are merged, resulting a global set which contains $N_{TRM}$ elements, with:

$$N_{TRM} = \sum_{i=1}^{N_A} N_{TRM,Ai}. \qquad (IV.4)$$

The most represented categories are determined by analyzing the number of occurrences of each class within the global set.

In order to validate the above-presented 2D object recognition framework, we propose an experimental evaluation, described in the following section.

## IV.4. EXPERIMENTAL EVALUATION

The aim of the experimental evaluation proposed is to analyze the classification power of the 2D object recognition framework proposed previously. In addition, we also aim at analyzing the influence of each 2D/3D indexing element (*i.e.,* the projection strategy and the 2D shape descriptor) involved in the classification process.

The 3D model repositories exploited in the recognition process are those presented in Section III.5:

- The MPEG7 3D model database (MPEG7_23), including $N_M$=362 models divided into 23 semantic categories (Section III.5.1);
- The Princeton shape benchmark (PSB), composed of $N_M$=1814 models; two classifications are associated with PSB, one including 53 categories (PSB_53) and the other one presenting 161 semantic classes (PSB_161 – *cf*. Section III.5.2).

Each 3D model is described using the view-based indexing approaches presented in Section III.2. The same viewing angle selection strategies (PCA3, PCA7, DODECA, DDPCA, OCTA9, OCTA33, RV6 and RV10 – *cf*. Section III.2.1) are here employed.

The retained shape descriptors are those described in Section III.2.2: MPEG-7 Region Shape (RS) and Contour Shape (CS), Hough Transform (HT), Zernike Moments (ZM), and Angular Histogram (AH).

In order to experimentally determine the classification power of the proposed framework, we have created several test datasets, described in the following section.

### IV.4.1. 2D object test datasets

Several test datasets were created in order to evaluate the object classification framework for both still and video objects.

#### IV.4.1.1. Still objects

First, we have created a dataset by selecting Still Objects from Images (SOI). The SOI set consists of $N_{SOI}$=115 images randomly acquired from the Web, by performing textual queries with existing image search engines (*e.g.*, Google Images, Flickr). The SOI dataset includes 23 object classes corresponding to the MPEG7_23 categories. For each category, 5 images have been retained. The objects of interest were manually segmented. However, we have experimentally

determined that similar recognition results are obtained when employing the interactive segmentation proposed in Chapter V. Figure IV.3 illustrates the still image dataset obtained.

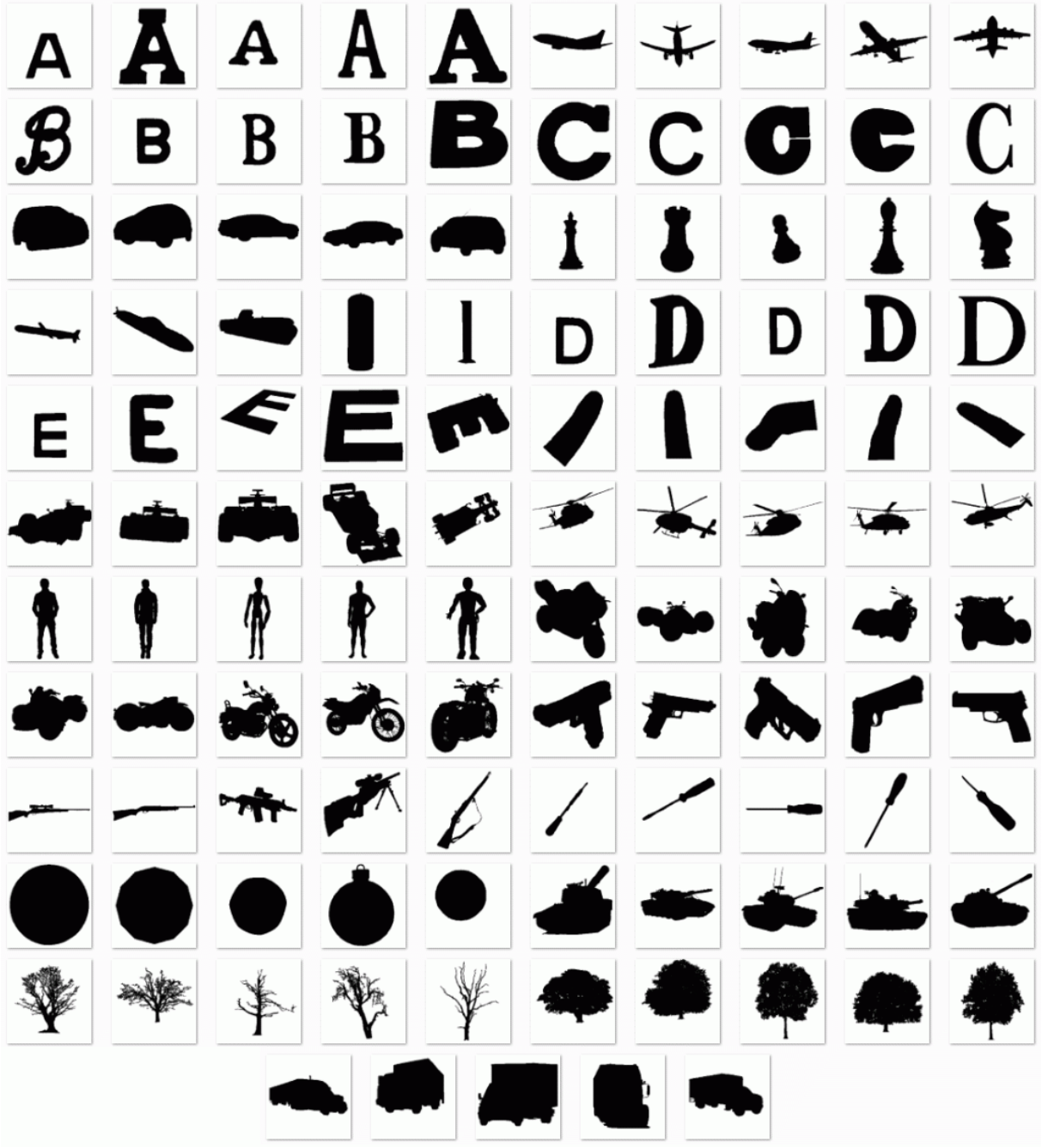


*Figure IV.3 Still object dataset.*

As solely 13 classes of models are common for all MPEG-7 and Princeton databases, only 65 objects are considered when performing experiments with the PSB_53 and PSB_161 repositories. They correspond to commercial airplanes, humans, sedan cars, tanks, race cars, motorcycles, helicopters, handguns, rifles, chess pieces, screwdrivers, trees and barren trees. When using the PSB_53 dataset for recognition, sedan cars, race cars and tanks are merged in the

cars class, trees and barren trees are included in the plants category and handguns combined with rifles become the guns class.

Annexe A3 contains the list of SOI classes. The list of 3D model databases used to test each 2D object class is listed in the same annexe.

### IV.4.1.2. Video objects

Next, we have created a video object database which includes $N_V$=40 videos selected from the Internet. Eight categories of objects have been considered, including airplanes, cars, chess pieces, helicopters, humanoids, motorcycles, pistols and tanks. Each one of these categories is present in 5 videos and for each video $N_F$=3 representative frames have been considered. Therefore, each video object consists of $N_I$=3 instances (Figure IV.4).

*Figure IV.4 Sample frames from VOV test set and the corresponding segmented objects.*

The considered frames have been segmented with the algorithm presented in Chapter V.3. From now on, the set of Video Objects extracted from Videos will be referred to as VOV.

The 2D objects segmented from each frame were also considered independently as still objects. Thus, an extended still object database, so-called SOV (*Still Objects extracted from Videos*) has been obtained, consisting in $N_{SOV}$=120 shapes.

### IV.4.1.3. Synthetic images

Finally, in order to further enlarge the 2D object test sets, we have created a set of synthetic images, representing 3D-to-2D projections of MPEG7 3D models. Only the 13 categories which are common for all databases have been considered. For each of them, three models were randomly selected, resulting in a subset of 39 3D models (Figure IV.5).

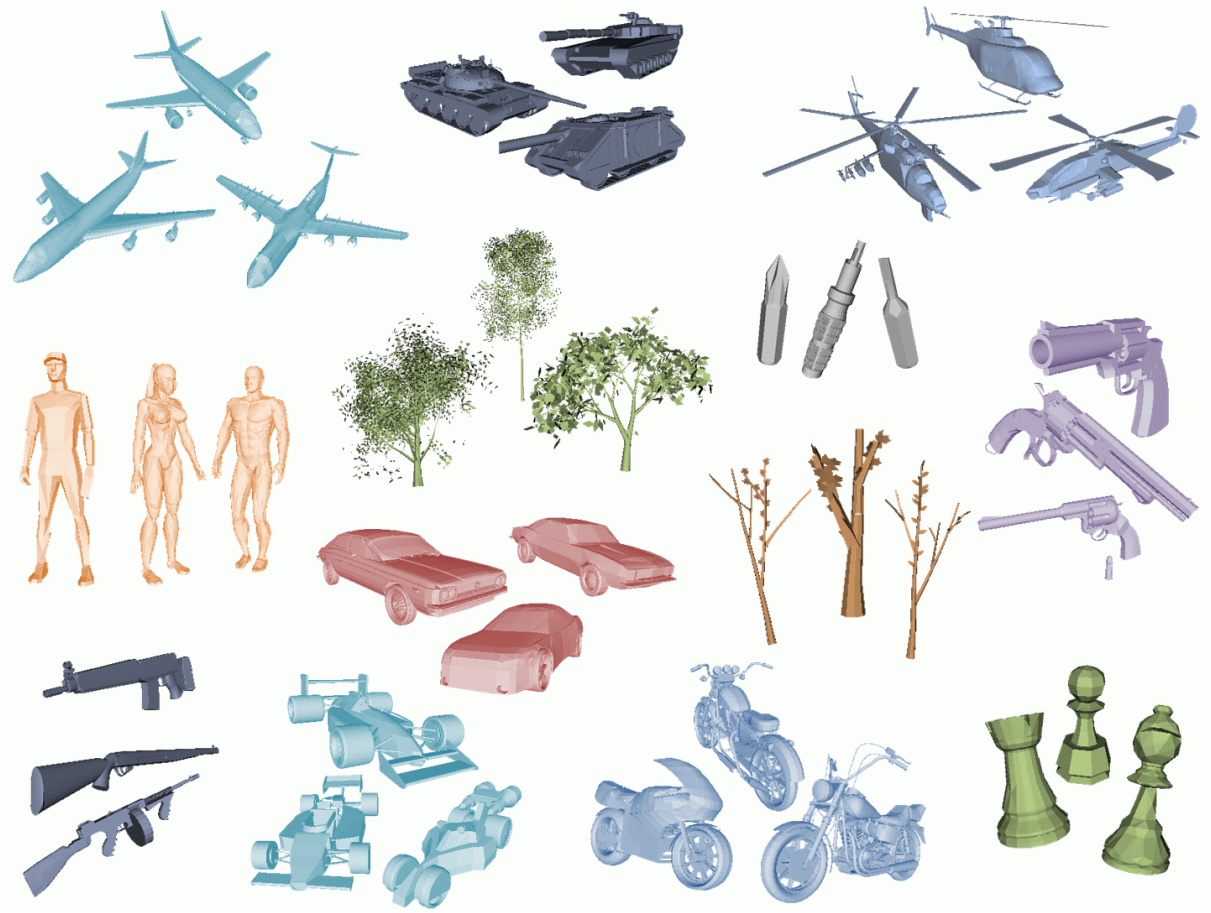


*Figure IV.5 The 3D models selected to generate the synthetic image dataset.*

Further, a number of 10 views, corresponding to the DODECA projection strategy, were generated for each of the 39 models, resulting in a synthetic dataset of $N_{Sy}$=390 classified images.

The 390 views were used to test the classification frameworks for both still and video object. In the first case, the images were considered independently, resulting in a set of ***S**till **O**bjects from **Sy**nthetic images* (SOSy) of $N_{SOSy}$=390 objects.

In the second case, the views of the same 3D model were considered as a video object. Therefore, the resulting set consists of $N_{VOSy}$=39 *Video Objects from Synthetic images* (VOSy), each one being composed of $N_I$=10 different instances of the object.

### IV.4.2. Evaluation protocol

The experiments have been carried out separately on each one of the five databases previously described (SOI, SOV and SOSy for still objects and VOV and VOSy for video objects).

All the 8 projection strategies and 5 shape descriptors have been considered individually for evaluation. In addition, in order to exploit the potential complementarities between different descriptors or projection strategies, we have also considered a combination strategy between various descriptors/projection approaches.

For each object $O$ to be recognized, a binary recognition label, denoted by $R(O, N_{MRC})$, is defined as follows:

$$R(O, N_{MRC}) = \begin{cases} 1; & if\ C(O) \in \{C_1 \dots C_{N_{MRC}}\} \\ 0; & otherwise \end{cases}, \qquad (IV.5)$$

where:

- $C(O)$ represents the category of the query object;
- $C_i$ denotes the $i^{th}$ category the most represented within the top retrieved models.

The recognition rate, denoted by $RR(N_{MRC})$, is defined as the mean value of the recognition for all the objects in the test dataset:

$$RR(N_{MRC}) = \frac{\sum_{i=1}^{N_O} R(O_i, N_{MRC})}{N_O}, \qquad (IV.6)$$

where:

- $N_O$ denotes the number of still/video objects in the test dataset.

In the ideal case, $RR(1) = 1$, which means that the correct category was the most represented (within the top retrieved models) for all the objects.

The recognition rate was computed by taking into account one, two or three most probable categories ($N_{MRC}$=1,2,3). For the case of PSB_161 database, we have also considered the $RR(10)$ score because of the higher number of available categories (161 classes).

When considering only the most represented category ($N_{MRC}$=1), the parameter $N_{TRM,Ai}$ has been set to 10 (*i.e.,* only the 10 top retrieved models have been taken into account for each indexing approach). When more than one class is considered ($N_{MRC}$>1) the set of retained 3D models is enlarged to 20.

In order to illustrate the recognition process, let us consider the example presented in Figure IV.6. The query consists of a still object representing a humanoid ($C(O)$=*humanoids* ). Two different indexing approaches, denoted by $A_1$ and $A_2$ (which may correspond to different shape

descriptors and/or viewing angle selection strategies), are here considered. For each one, the eight top retrieved models have been retained (represented by the 3D model's view the most similar to the query).

When the first approach was used (*i.e.*, $A_1$), *cars* was determined as the category the most represented (4 occurrences) within the first $N_{TRM} = 8$ top retrieved models. The second category is *humanoids*, with 3 occurrences, while *pistols* class is represented by only 1 model ($C_1$=*cars*, $C_2$=*humanoids* and $C_3$=*pistols*). Therefore, when considering only the most represented category, $R(O,1) = 0$; however, $R(O, N_{MRC}) = 1$ for $N_{MRC} \geq 2$.

In the case of the second approach (*i.e.*, $A_2$), $C_1$=*pistols* and $C_2$=*humanoids*, are the most represented categories, both with 3 occurrences; $C_3$=*helicopters* and $C_4$=*cars* are represented by only one model each. When two classes have the same number of occurrences, the one having the best placed model is considered first (*e.g., pistols* and *humanoids* are represented by the same number of models but the best retrieved pistol is in the first position, while the best retrieved humanoid is in the fourth position).

Finally, when considering the combination of approaches $A_1$ and $A_2$, the *humanoids* class is determined in the first position, with 6 retrieved models, followed by *cars*, *pistols* and *helicopters* with 5, 4, respectively 1 occurrences.

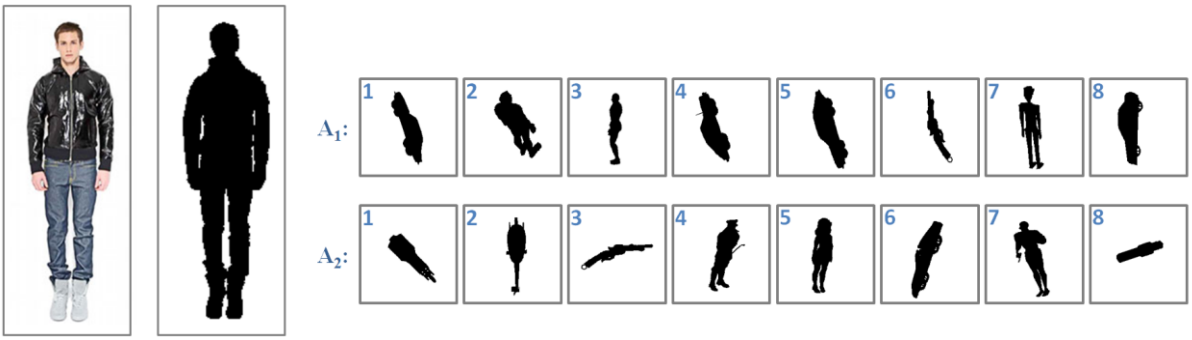



*Figure IV.6 Example of recognition rate computation.*

### IV.4.3. Results and discussion

Let us start by presenting the results obtained for still objects.

#### IV.4.3.1. Still objects

Figure IV.7, Figure IV.8 and Figure IV.9 respectively present the recognition rates obtained on the SOI test set with the MPEG7_23, PSB_53 and PSB_161 3D model databases. Similarly, the recognition rates obtained on the three databases are illustrated in Figure IV.10, Figure IV.11 and Figure IV.12 for the SOSy test set and in Figure IV.13, Figure IV.14 and Figure IV.15 for the SOV dataset. The corresponding values of the recognition rates obtained are detailed in Annexe A4.

In the case of SOI test set, the highest recognition rates *RR*(1) are about 51%, 74% and 54% when MPEG7_23, PSB_53 respectively PSB_161 models were employed. The *RR*(3) scores are up to 71%, 86% and 66%, respectively.

Similar recognition rates are obtained when testing the SOV objects: 54%, 64% and 53% in terms of *RR*(1) and 79%, 83% and 69% in terms of *RR*(3) with each one of the 3D model databases.

Finally, in the case of synthetic objects (SOSy database), the best *RR*(1) rates are about 58%, 63% and 56% while *RR*(3) are up to 84%, 82% and 75% with MPEG7_23, PSB_53 and PSB_161 respectively.

The highest recognition rates are obtained in all cases with the contour-based descriptors (*i.e.,* AH and CS). As in the case of 3D model retrieval, we observe that contour-based descriptors lead to higher recognition rates than those exploiting the region (*i.e.,* HT, RS and ZM). The difference between the two families of descriptors is about 10% to 20%. We observed that this difference is less important in the case of SOV test objects (4% to 13%). This behaviour may be explained by the fact that SOV objects present poorer resolution (45×85 – 500×350) than the SOI objects. The poorest results are obtained with HT, which leads to a mean difference of 10%-15% compared to other descriptors. On the contrary, the highest classification scores are generally obtained with AH on SOI and SOSy test sets and with CS on SOV objects.

Concerning the viewing angle selection, we observed here again that the repartition strategy is more important than the number of views. Thus, DODECA, RV6 or RV10 globally outperform OCTA9 and lead to rates similar to OCTA33, for 3 to 5 times less views. This shows the pertinence of employing representative views, such as those obtained by the RV6 and RV10 strategies, which both avoid redundancies and ensure a discriminative shape representation. Also, when comparing DODECA and DDPCA we observed that the first strategy presents better overall scores (up to 20%). As explained when discussing the 3D model retrieval framework results, the difference between DODECA and DDPCA may be induced by the fact that numerous 3D models present symmetries. As DDPCA cameras are places symmetrically with respect to the principal planes of the model, the resulting silhouettes represent mirror-reflected versions of the same image. Thus, DDPCA set may include redundancy, which reduces the amount of relevant information. The same phenomenon appears in the case of OCTA9 and OCTA33 and leads to relatively poor results with respect to the number of views employed.

When comparing the results obtained with PSB_53 and with PSB_161, we observe that the first database leads to better results, which is explained by the lower number of classes. However, if we consider that selecting 3 possibilities among 53 is similar to selecting 10 classes among 161, than we should compare *RR*(3, PSB_53) to *RR*(10, PSB_161). It can be observed in the corresponding figures and tables that the two scores present very similar values.

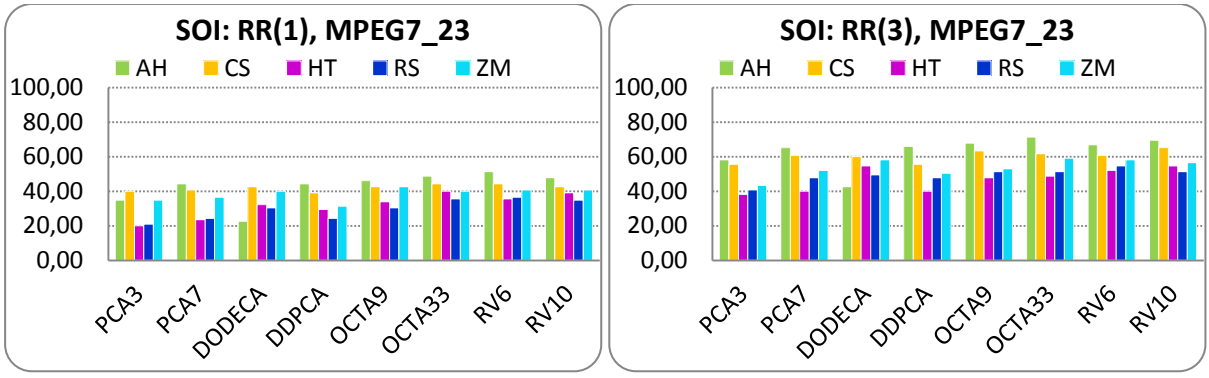



*Figure IV.7 SOI database: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

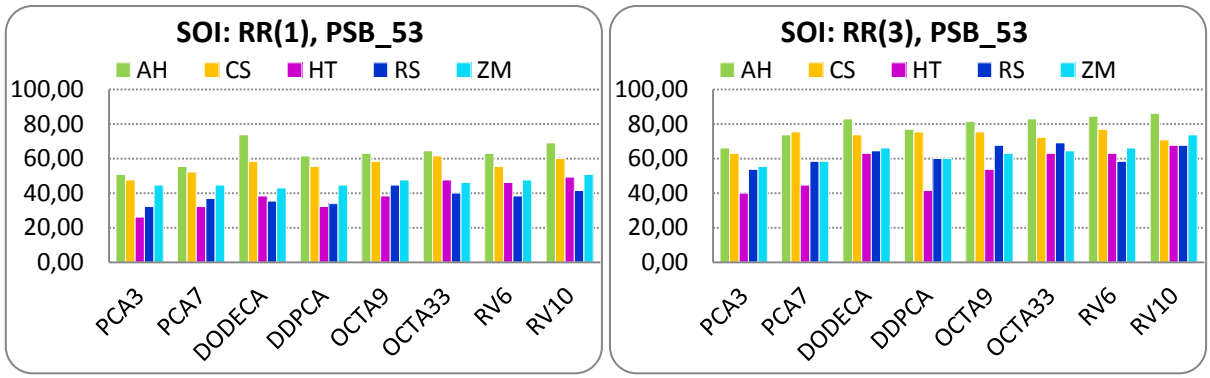



*Figure IV.8 SOI database: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

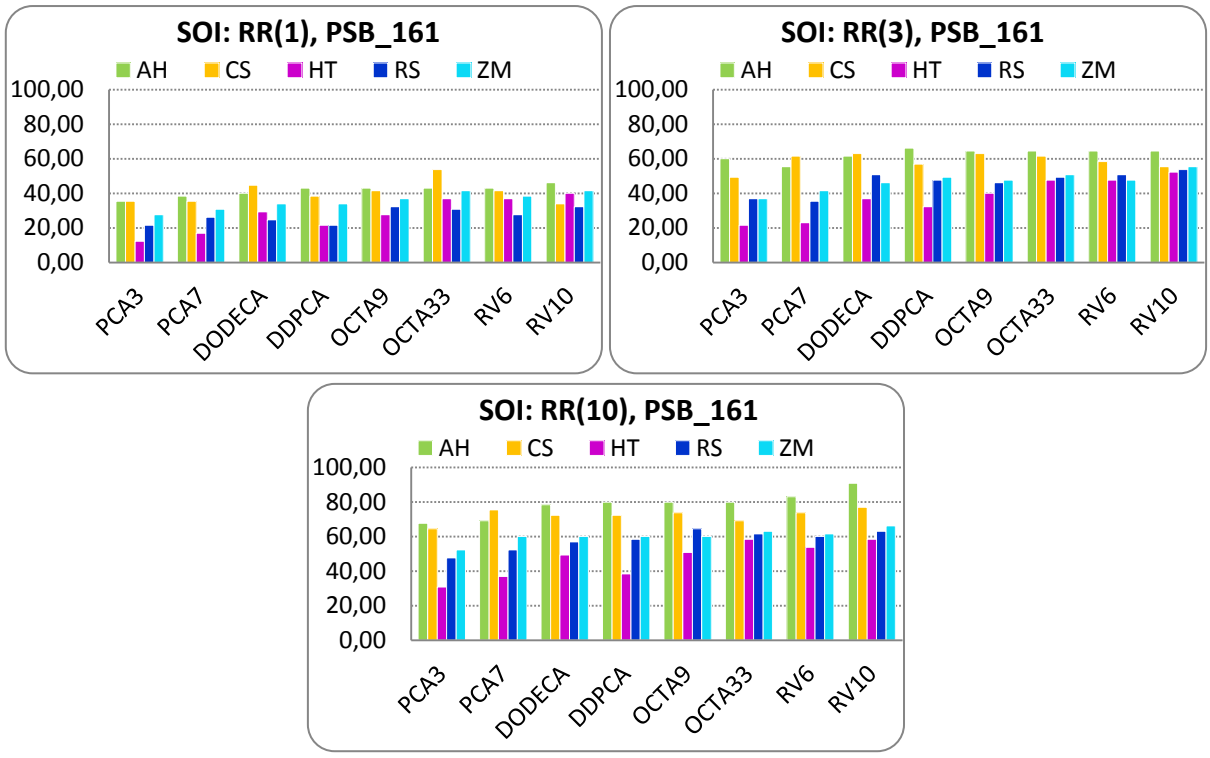



*Figure IV.9 SOI database: RR(1), RR(3) and RR(10) scores obtained with the help of PSB_161 3D models.*

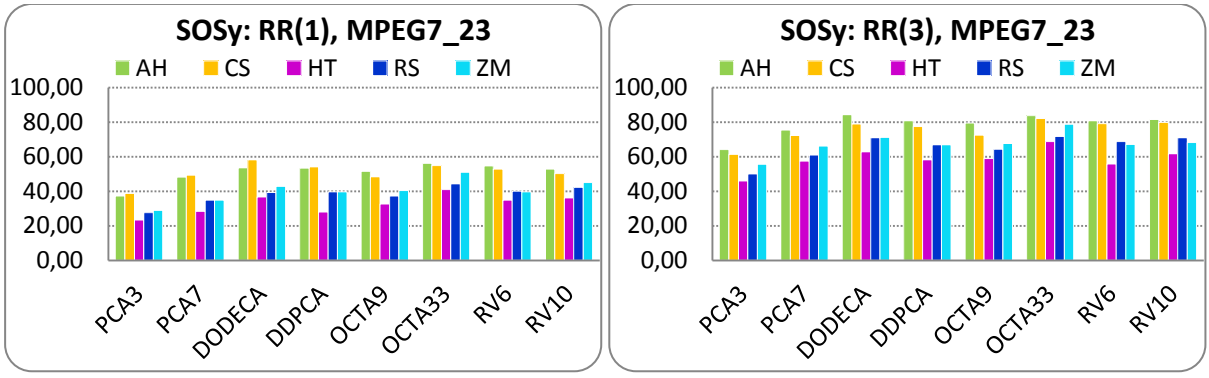



*Figure IV.10 SOSy database: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

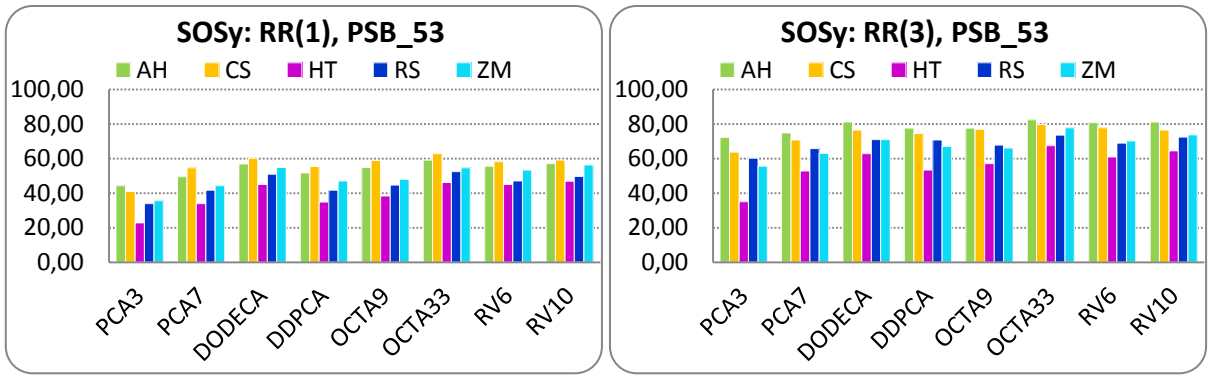



*Figure IV.11 SOSy database: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

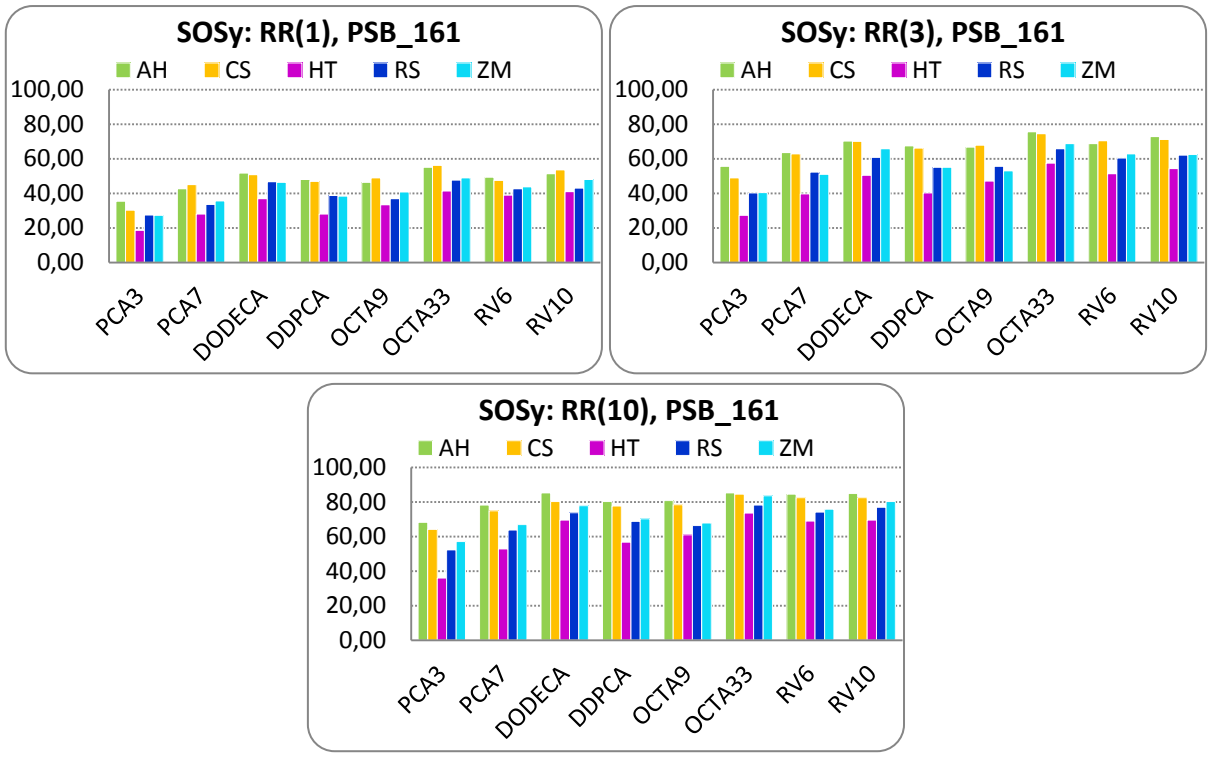



*Figure IV.12 SOSy database: RR(1), RR(3) and RR(10) scores obtained with the help of PSB_161 3D models.*

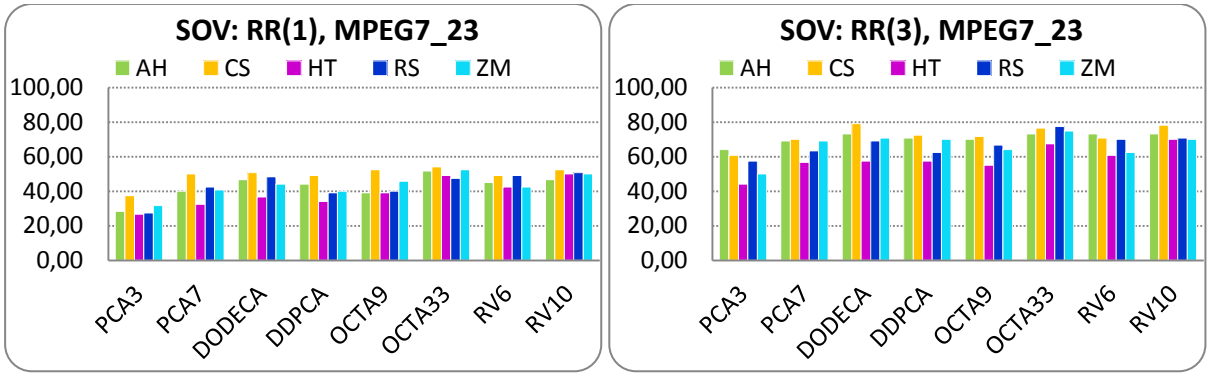



*Figure IV.13 SOV database: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

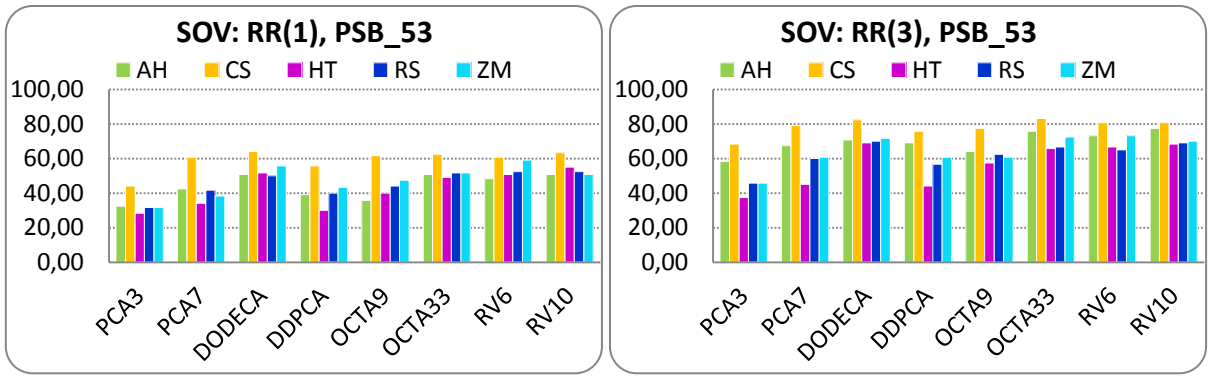



*Figure IV.14 SOV database: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

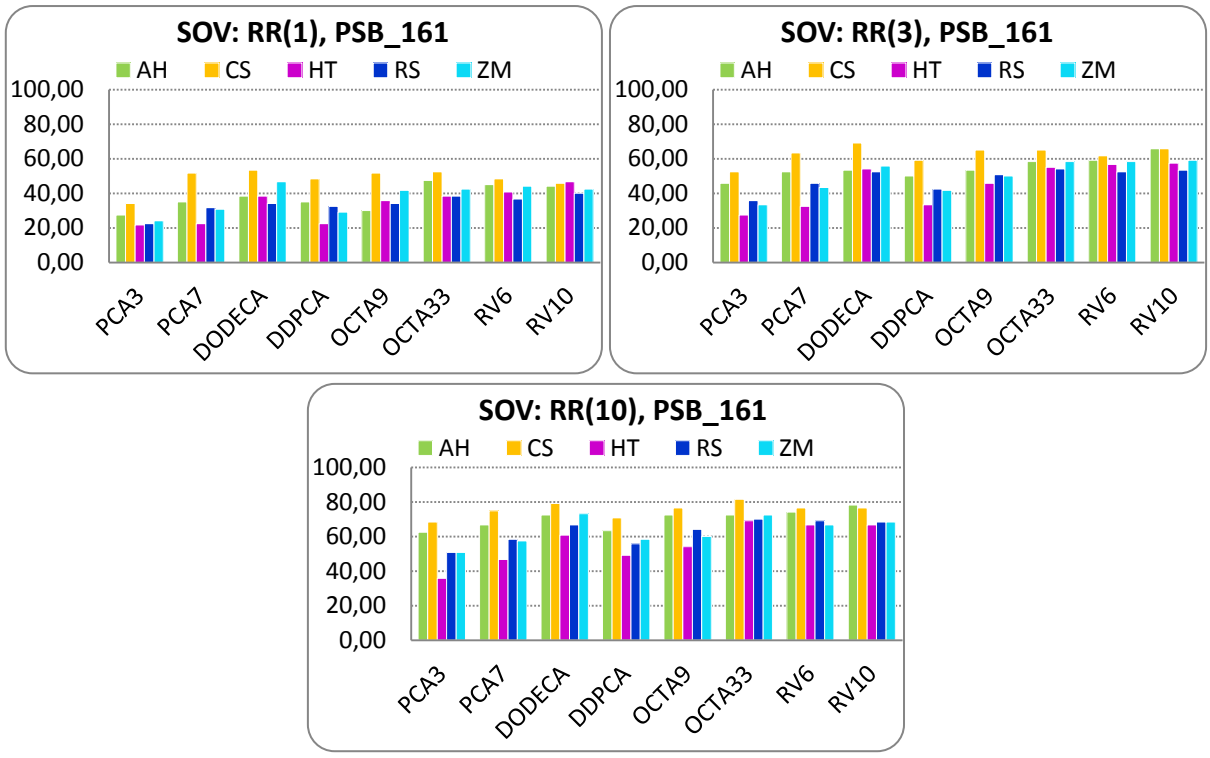



*Figure IV.15 SOV database: RR(1),RR(3)and RR(10) scores obtained with the help of PSB_161 3D models.*

Further, we have analyzed the individual behaviour of each category of models. Here, we have considered only the DODECA projection strategy, as it is one of the best performing. Figure IV.16 to Figure IV.24 illustrate the *RR*(3) scores obtained for each category of the still object test sets. The correspondence between the number and the name of the categories can be found in Annexe A3.

We observe that some categories, such as *airplanes*, *humanoids* and *cars*, are globally easier to classify and lead to high recognition rates in most of the cases and with all descriptors. Some other categories, as *motorcycles* and *tanks*, present poorer scores. We observed that *tanks*, *trucks* and *Formula 1* are often classified as *cars*. When using the PSB_161 3D model database, many *commercial airplanes* are detected as *fighter jet aircrafts*. Such mistakes can be explained by the similarity between the true class and the one mis-assigned.

By analysing the results for each class of objects, we also observe that the relative performance of the retained descriptors depends on the semantic class. Therefore, we tested our system when two descriptors were combined (*cf.* the combination strategy described in Section IV.3.2) in order to exploit the possible complementarities between them.

We have chosen to combine the two descriptors with the best performances (*i.e.,* CS and AH). We have also retained ZM, the best region-based descriptor, and combined it in turn with AH and CS.

Figure IV.25 to Figure IV.33 illustrate the recognition rates obtained individually with the AH and CS descriptors, but also with combined descriptors: AH&CS, AH&ZM and CS&ZM. The corresponding values are provided in Annexe A4.

We observed that the combined AH&CS method provide recognition rates $RR_{AH\&CS}$ slightly superior to the maximum scores between CS and AH ($\max(RR_{AH}, RR_{CS})$): 2%-4% higher with the MPEG7_23 models, a gain of 2%-6% when PSB_53 DB is employed and up to 10% higher when PSB_161 models are involved. However, when the recognition rates obtained with each one of the two descriptors are very different ($|RR_{AH}-RR_{CS}|>15\%$), the rate obtained with the combined method is inferior to the maximum between $RR_{AH}$ and $RR_{CS}$. In these cases, the poorest method has a negative influence on the result of the combined approach.

When comparing the rate obtained with the combined method $RR_{AH\&CS}$ with the mean $RR_m=(RR_{AH}+RR_{CS})/2$ of the rates obtained with AH and CS separately, we observe that the combined approach leads to scores superior to the mean rate $RR_m$. The gain obtained is up to 10% (with a mean of 3%-4%) when MPEG7_23 database is used, up to 13% (with a mean of 6%-7%) when employing PSB_53 models and up to 16% (with a mean of 7%-8%) when PSB_161 DB was involved.

When a contour-based descriptor (*i.e.,* AH or CS) is combined with ZM, the performance of the contour-based descriptor is sometimes slightly improved; however, in some other cases the recognition rate is inferior to the one obtained only with the contour-based descriptor. In conclusion, a better improvement is obtained when a contour-based descriptor is combined with the other contour-based descriptor than when ZM is involved.

Globally, we observe that the combination of the two contour-based descriptors improves the results and guarantees more stability in the recognition process. Thus, we reached $RR_{AH\&CS}(3)$ up to 69% and 86% with MPEG7_23 respectively PSB_53 models, while with PSB_161 the $RR_{AH\&CS}(3)$ score was up to 77% and $RR_{AH\&CS}(10)$ up to 91%.

Let us underline that such recognition rates are highly promising, since in the case of still images, the recognition is achieved starting from a single image. Let us now analyze the case of video objects, where multiple object instances are available.

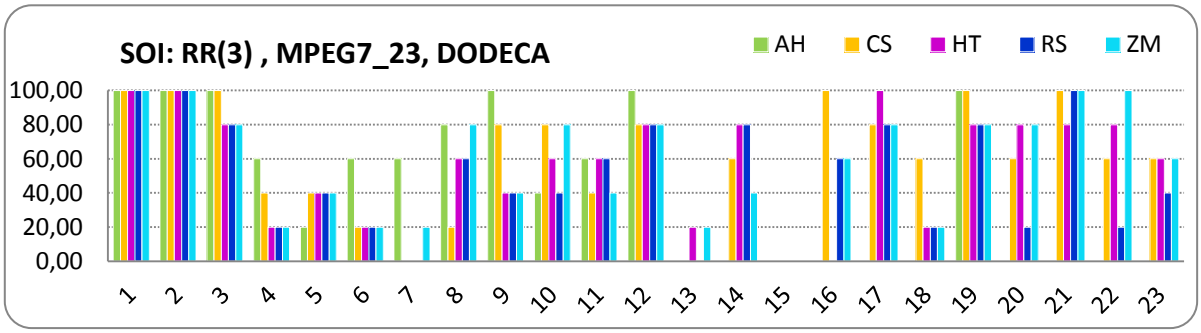



*Figure IV.16 SOI database: RR(3) scores per category obtained with the help of MPEG7_23 3D models.*

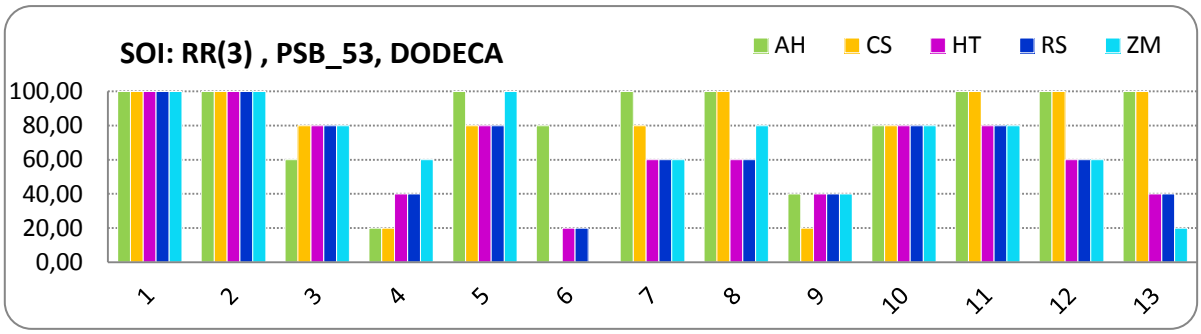



*Figure IV.17 SOI database: RR(3) scores per category obtained with the help of PSB_53 3D models.*

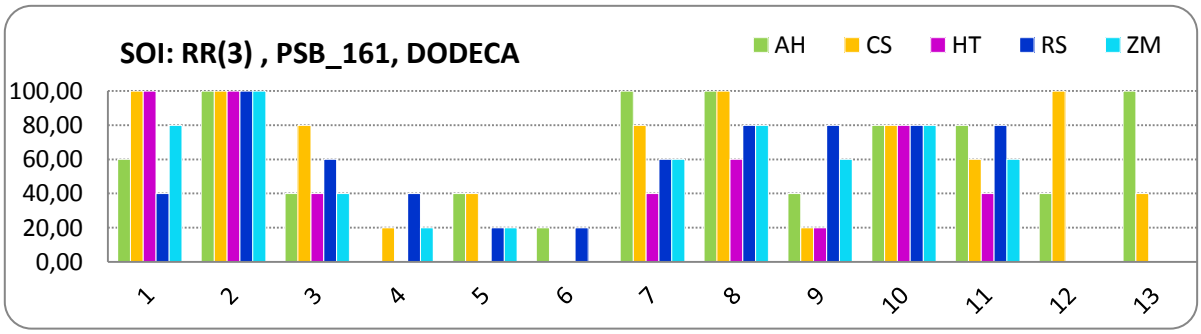



*Figure IV.18 SOI database: RR(3) scores per category obtained with the help of PSB_161 3D models.*

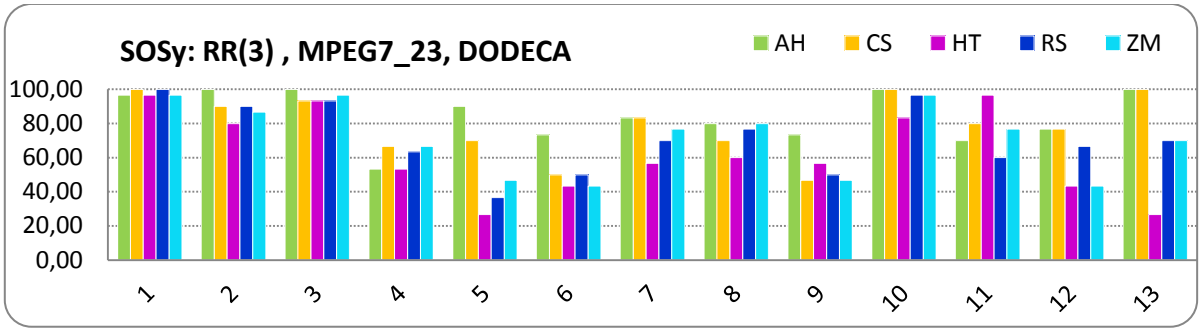



*Figure IV.19 SOSy database: RR(3) scores per category obtained with the help of MPEG7_23 3D models.*

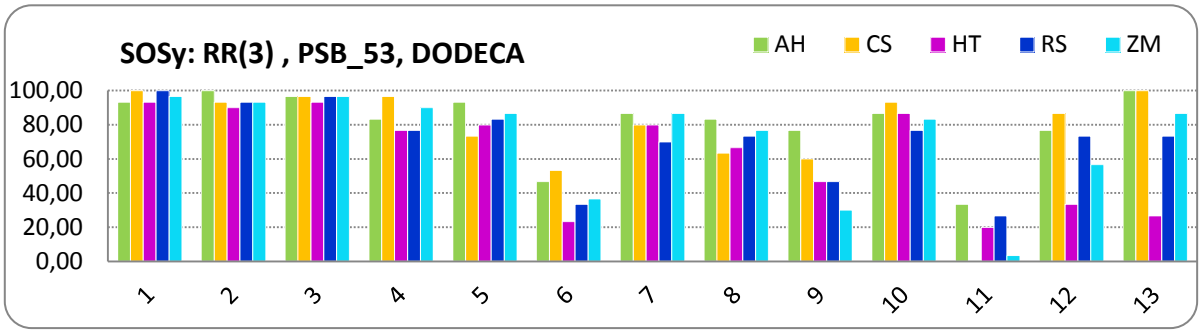



*Figure IV.20 SOSy database: RR(3) scores per category obtained with the help of PSB_53 3D models.*

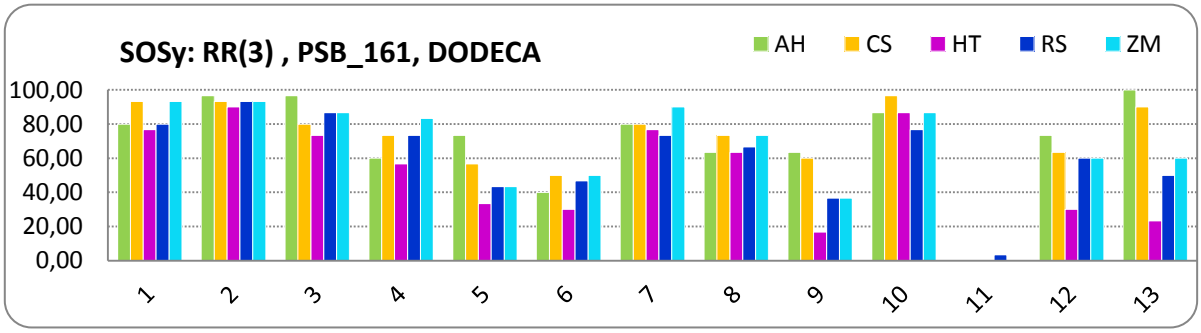



*Figure IV.21 SOSy database: RR(3) scores per category obtained with the help of PSB_161 3D models.*

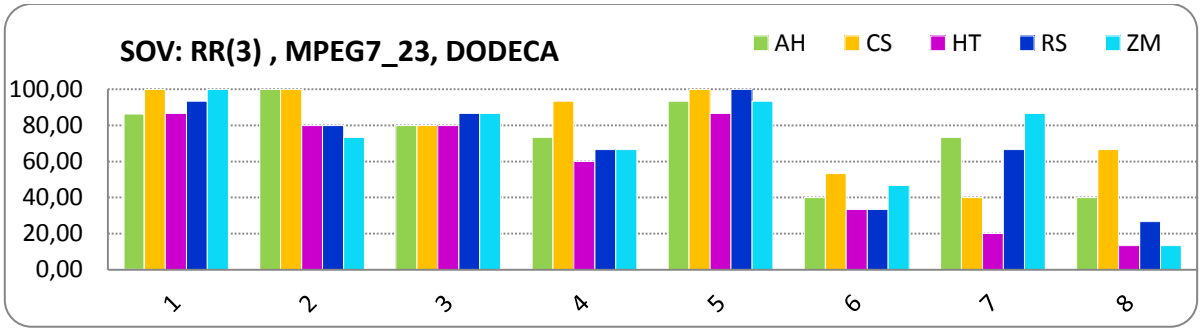



*Figure IV.22 SOV database: RR(3) scores per category obtained with the help of MPEG7_23 3D models.*

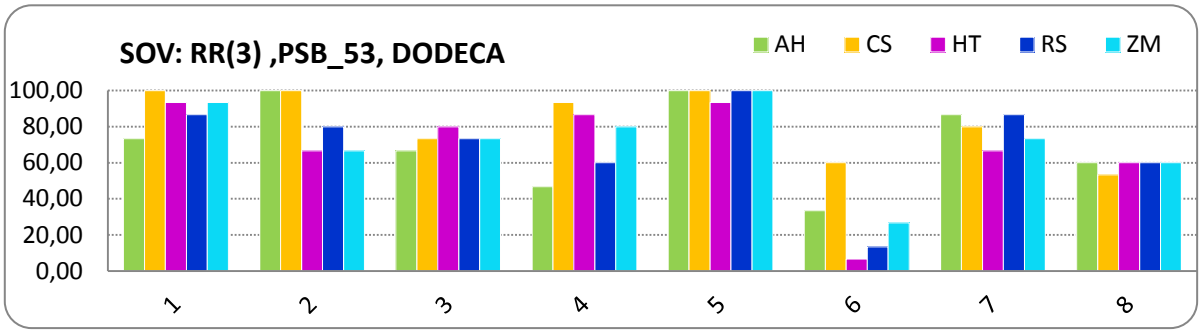



*Figure IV.23 SOV database: RR(3) scores per category obtained with the help of PSB_53 3D models.*

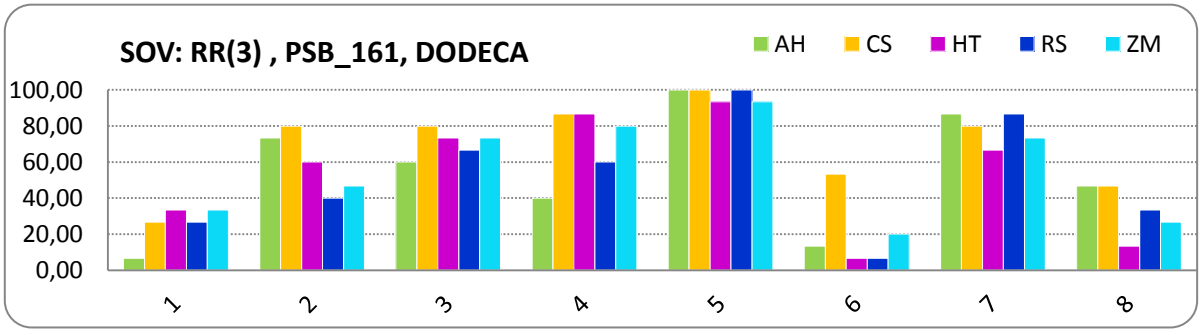



*Figure IV.24 SOV database: RR(3) scores per category obtained with the help of PSB_161 3D models.*

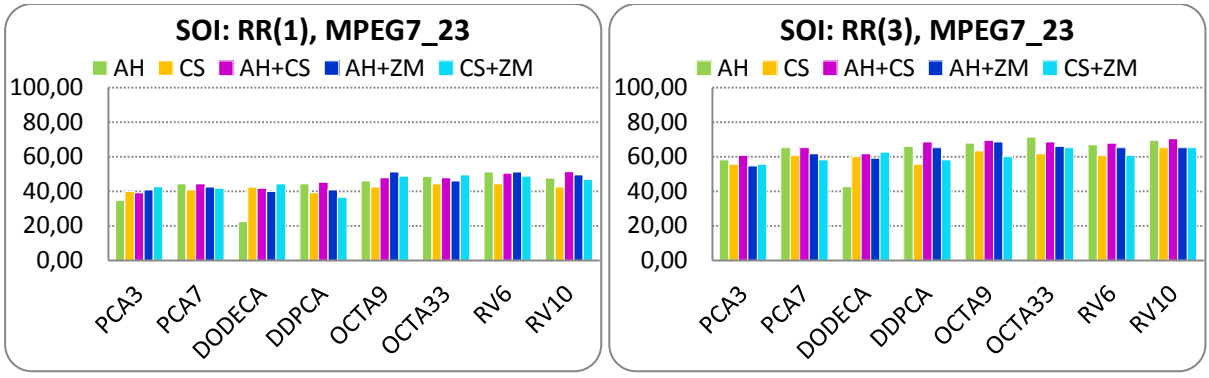



*Figure IV.25 SOI database – combined descriptors: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

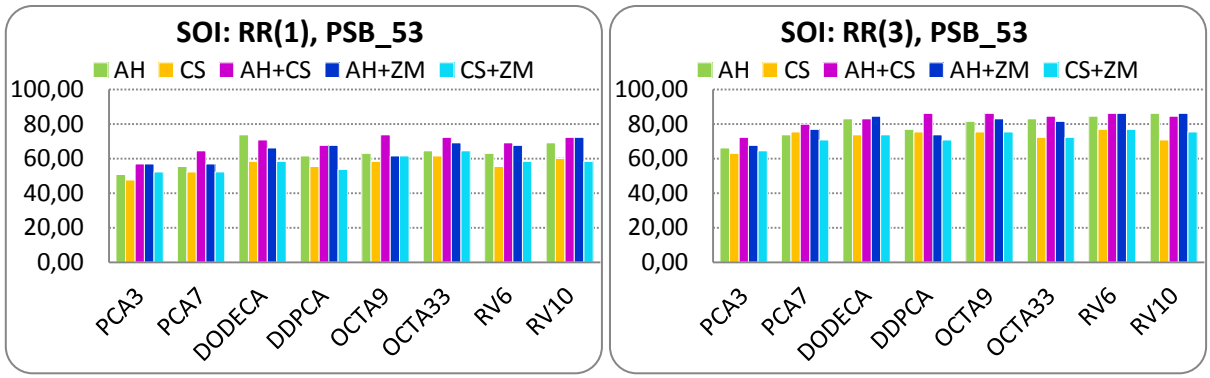



*Figure IV.26 SOI database – combined descriptors: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

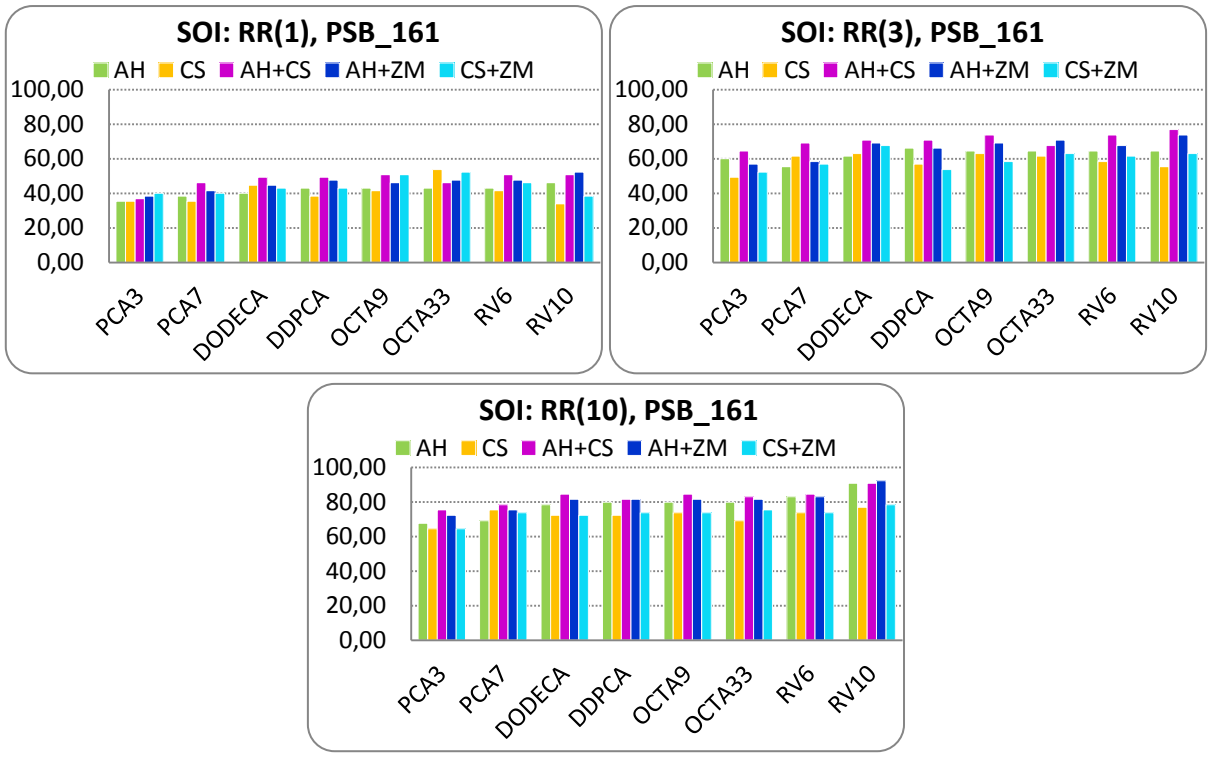



*Figure IV.27 SOI database – combined descriptors: RR(1), RR(3) and RR(10) scores obtained with the help of PSB_161 3D models.*

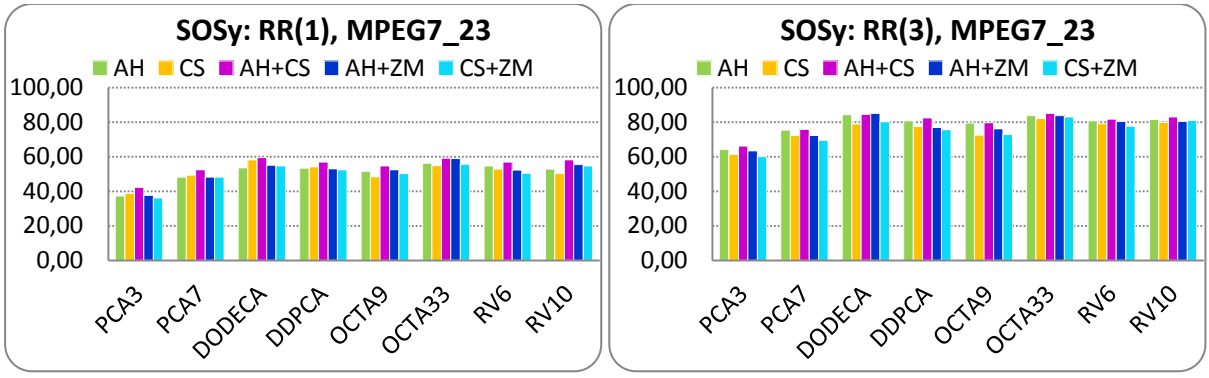



*Figure IV.28 SOSy database – combined descriptors: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

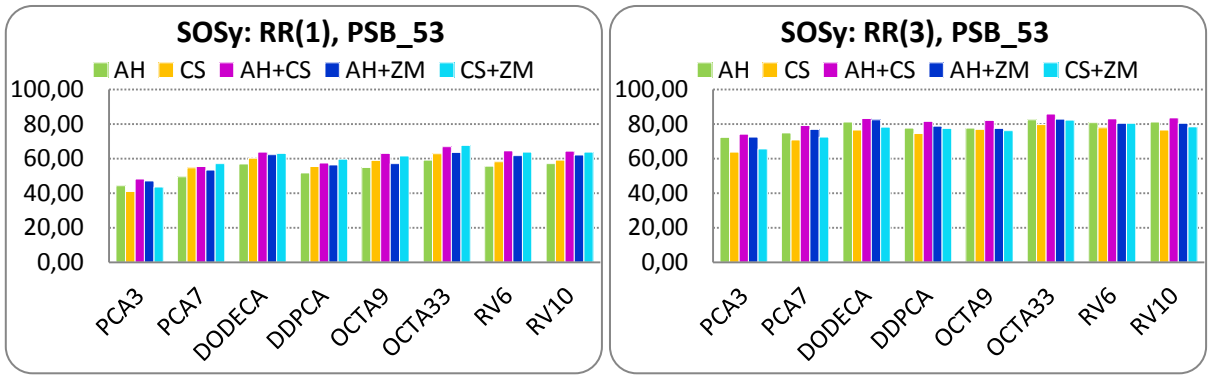



*Figure IV.29 SOSy database – combined descriptors: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

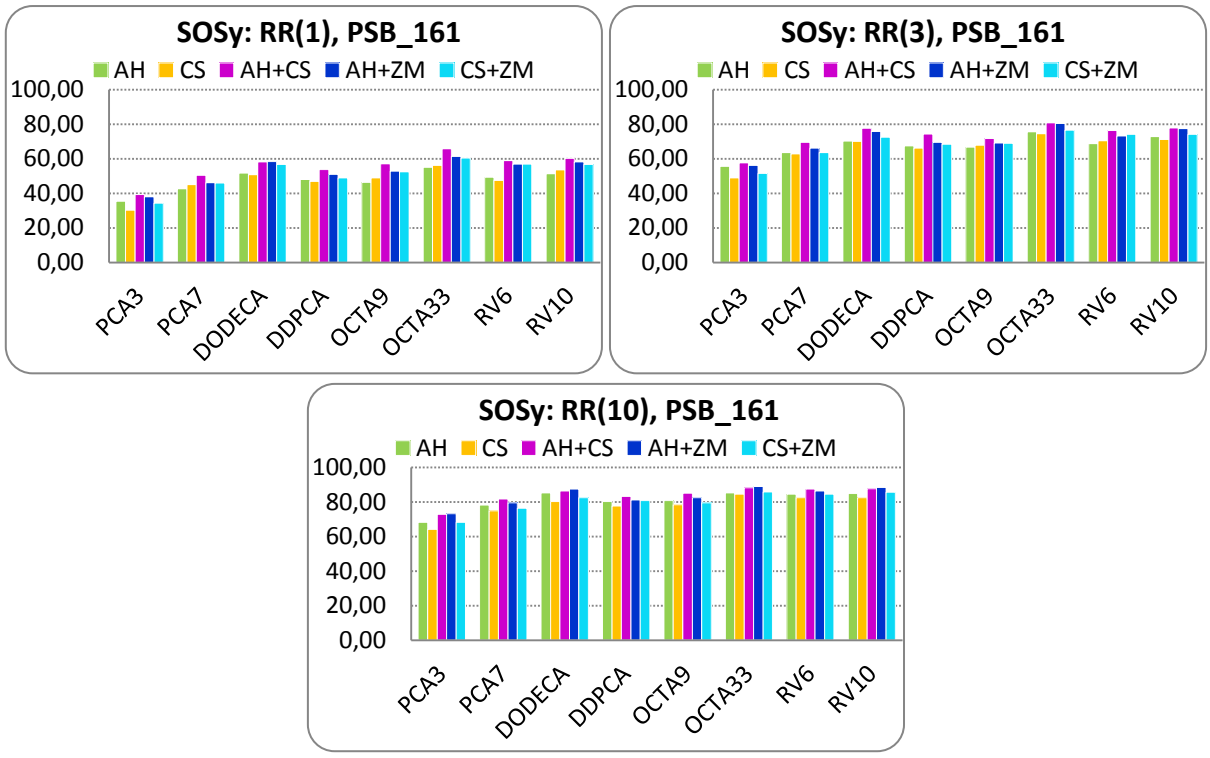



*Figure IV.30 SOSy database – combined descriptors: RR(1), RR(3) and RR(10) scores obtained with the help of PSB_161 3D models.*

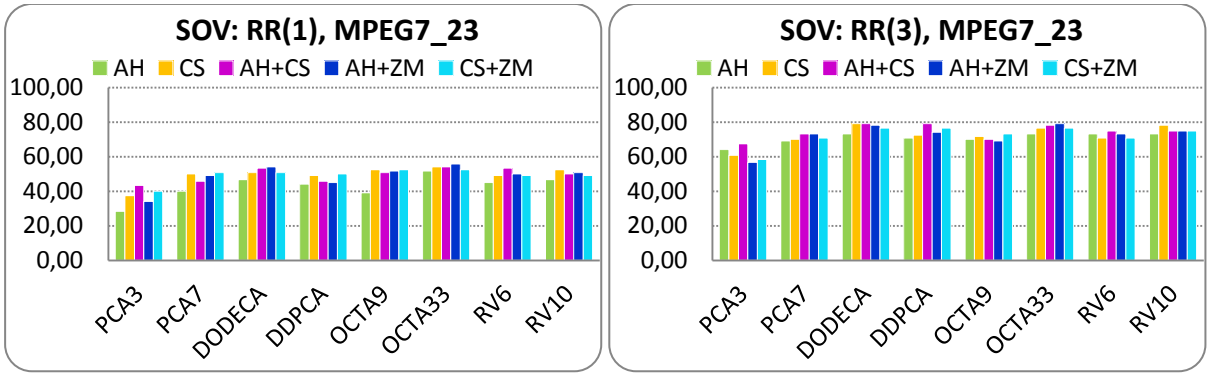



*Figure IV.31 SOV database – combined descriptors: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

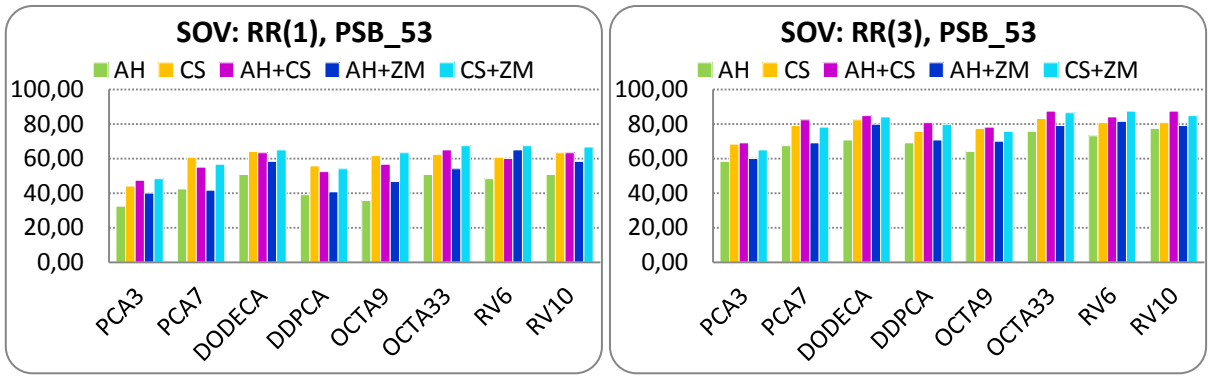



*Figure IV.32 SOV database – combined descriptors: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

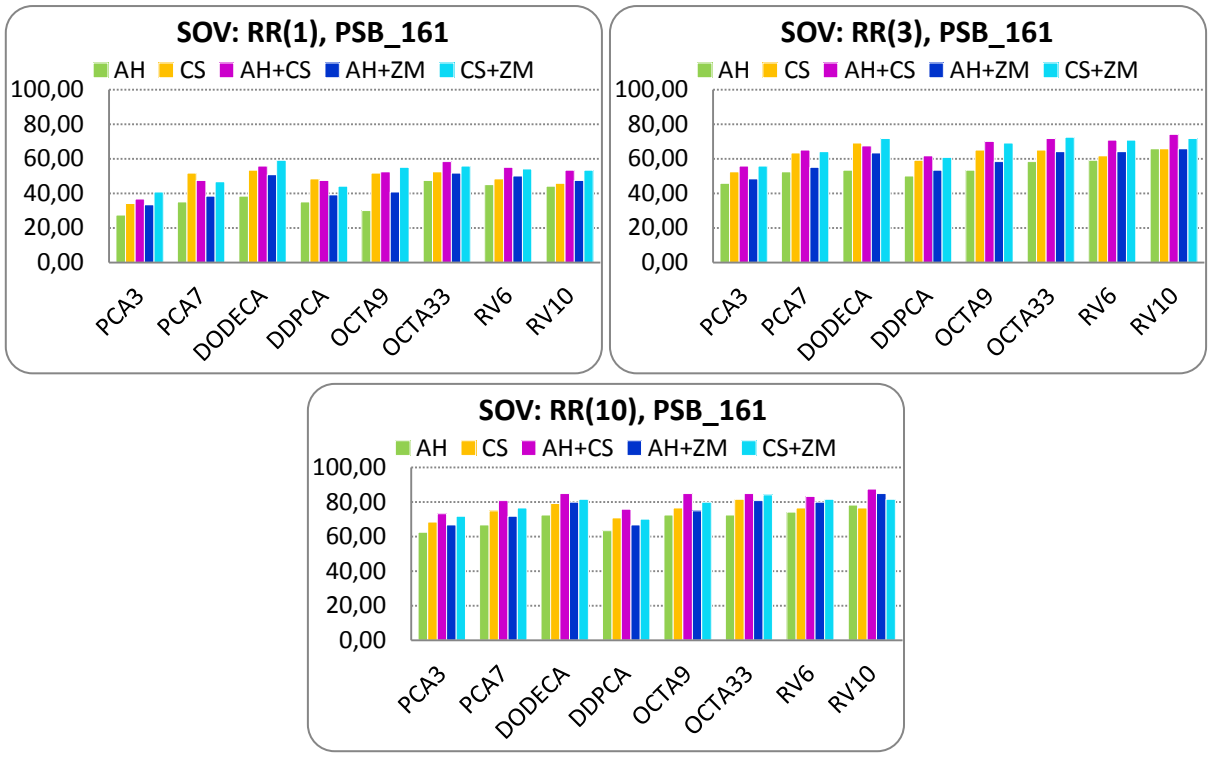



*Figure IV.33 SOV database – combined descriptors: RR(1), RR(3) and RR(10) scores obtained with the help of PSB_161 3D models.*

### IV.4.3.2. Video objects

The video object recognition rates obtained on the VOV test set with the help of MPEG7_23, PSB_53 and PSB_161 3D models are respectively presented in Figure IV.34, Figure IV.35 and Figure IV.36. The rates obtained on the same databases are illustrated in Figure IV.37, Figure IV.38 and Figure IV.39, for the case when combined descriptors are employed. The precise recognition values are provided in Annexe A4.

As in the case of SOV test set (where the same images were considered independently), here again we observe that the CS descriptor provides maximal performance, with:

- $RR(1) = 75\%$and $RR(3) = 85\%$, for the MPEG7_23 data set,
- $RR(1) = 85\%$ and $RR(3) = 92.5\%$ for the PSB_53 models,
- $RR(1) = 77.5\%$, $RR(3) = 87.5\%$ and $RR(10) = 92.5\%$, for the PSB_161 data set.

In terms of $RR(1)$ rate, the CS descriptor outperforms AH by about 15%. However, the AH descriptor presents more important gains when $N_{MRC}$ increases (*i.e.,* from $RR(1)$ to $RR(3)$ and $RR(10)$) then CS. Thus, in terms of $RR(3)$ and $RR(10)$ the AH and CS descriptors lead to similar results. The AH descriptor achieves $RR(3)$ scores up to 87.5% with the MPEG7_23 and PSB_53 models. For the PSB_161 models, the AH descriptor reaches $RR(3)$ of 77.5% and $RR(10)$ of 92.5%.

Concerning the combination of the AH and CS descriptors, the hybrid strategy is effective in the sense of the RR(3) and RR(10) scores. Thus, for the combined approach the recognition rates $RR(3)$ are up to 87.5% with MPEG7_23, 97.5% with PSB_53 and 90% with PSB_161 models.

Further, we have compared the recognition scores obtained on the set of images extracted from videos when they were tested independently as still objects (*i.e.,* the SOV test set) and when each query consists of several images (*i.e.,* the VOV test set). We observe that, by increasing the input information from one image (in the case of still objects) to three images (in the case of video objects), the recognition performance is improved by 10% to 20%. This shows the pertinence of considering in the recognition process multiple images as input.

In order to analyse how the recognition scores increases with $N_I$ (the number of instances used to formulate each video object query), we have considered the set of synthetic images. Here, we have generated up to $N_I$=10 instance for each query. Figure IV.40, Figure IV.41 and Figure IV.42 illustrate the recognition rates obtained with different $N_I$, for the case where DODECA projection strategy was employed. We observe that the considered recognition rates (*i.e., RR*(1) and $RR(3)$ for all 3D model databases and $RR(10)$ for PSB_161) are significantly improved (by about 10%) when the number of instances increases from one to two images per video object query. When the third instance is added to the query, the recognition improves by about 5%. Furthermore, each new instance leads to a very limited improvement of the scores (less than 1-2%), notably in the case of $RR(3)$ and $RR(10)$ rates.

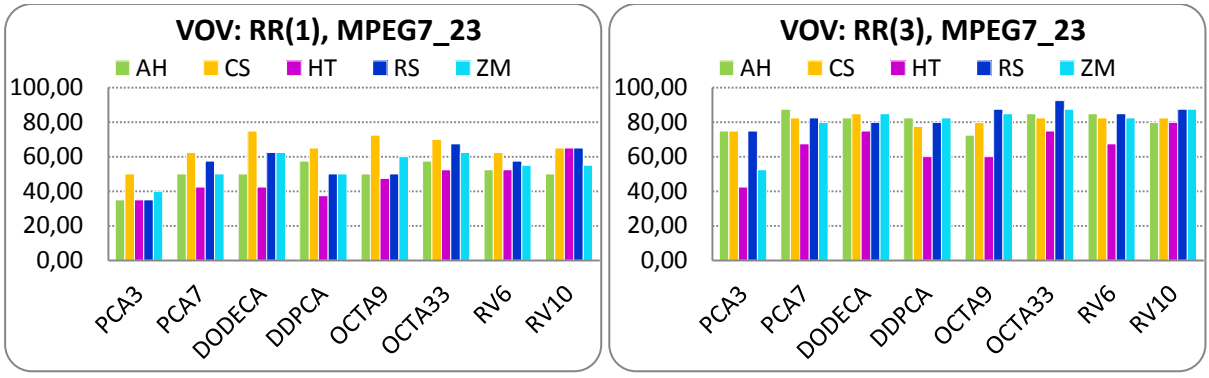



*Figure IV.34 VOV database: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

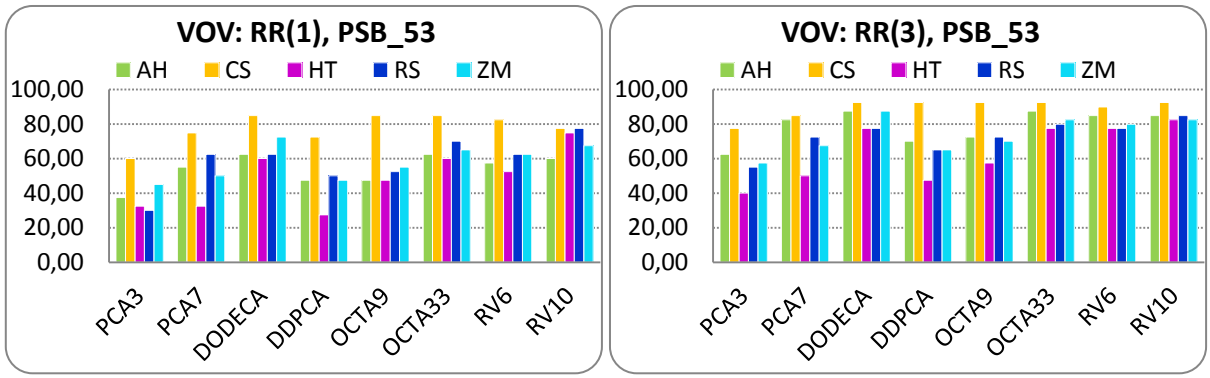



*Figure IV.35 VOV database: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

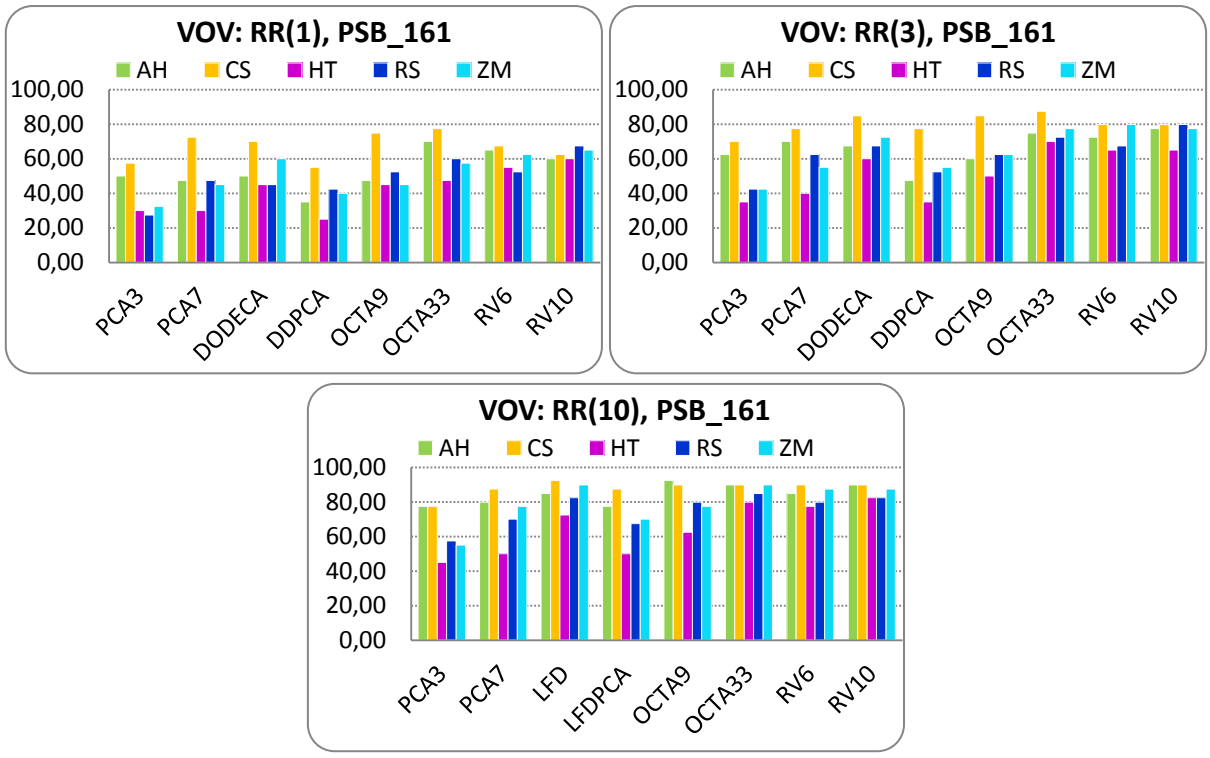



*Figure IV.36 VOV database: RR(1),RR(3)and RR(10) scores obtained with the help of PSB_161 3D models.*

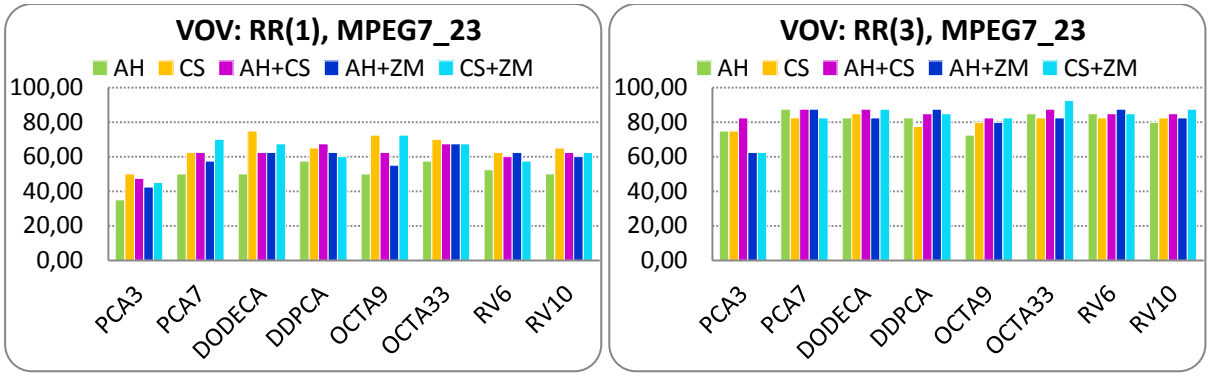



*Figure IV.37 VOV database – combined descriptors: RR(1) and RR(3) scores obtained with the help of MPEG7_23 3D models.*

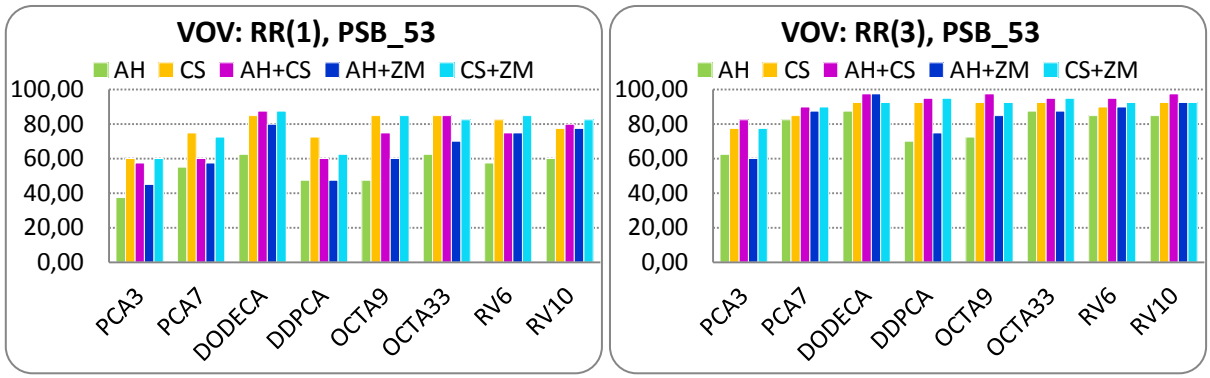



*Figure IV.38 VOV database – combined descriptors: RR(1) and RR(3) scores obtained with the help of PSB_53 3D models.*

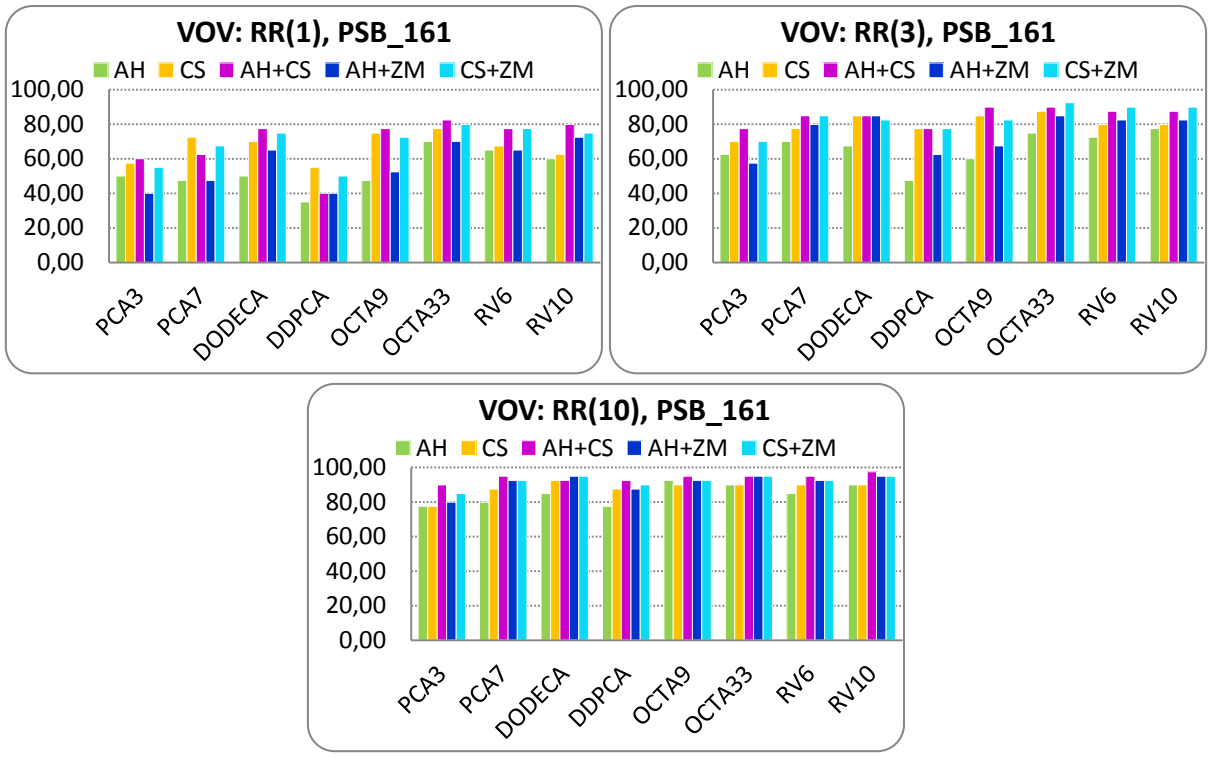



*Figure IV.39 VOV database – combined descriptors: RR(1), RR(3) and RR(10) scores obtained with the help of PSB_161 3D models.*

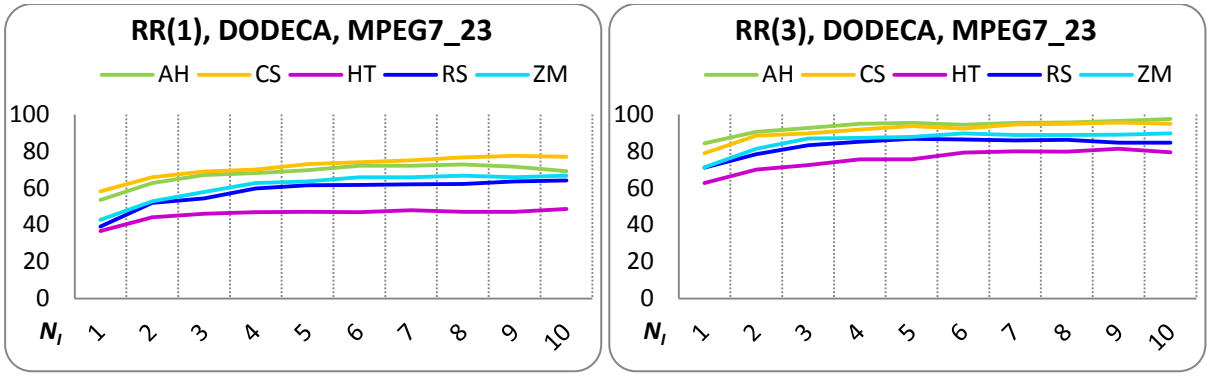



*Figure IV.40 VOSy database: RR(1) and RR(3) scores obtained with DODECA strategy and MPEG7_23 3D models for different $N_I$ per VO query.*

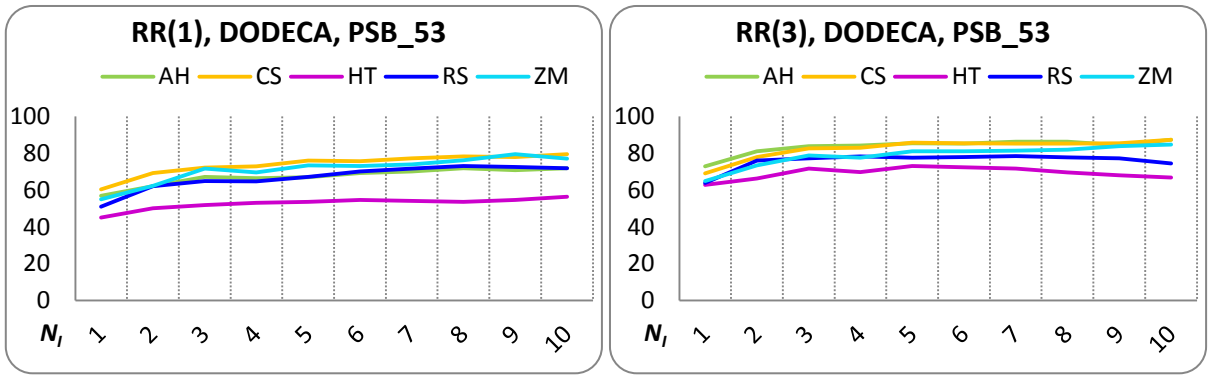



*Figure IV.41 VOSy database: RR(1) and RR(3) scores obtained with DODECA strategy and PSB_53 3D models for different $N_I$ per VO query.*

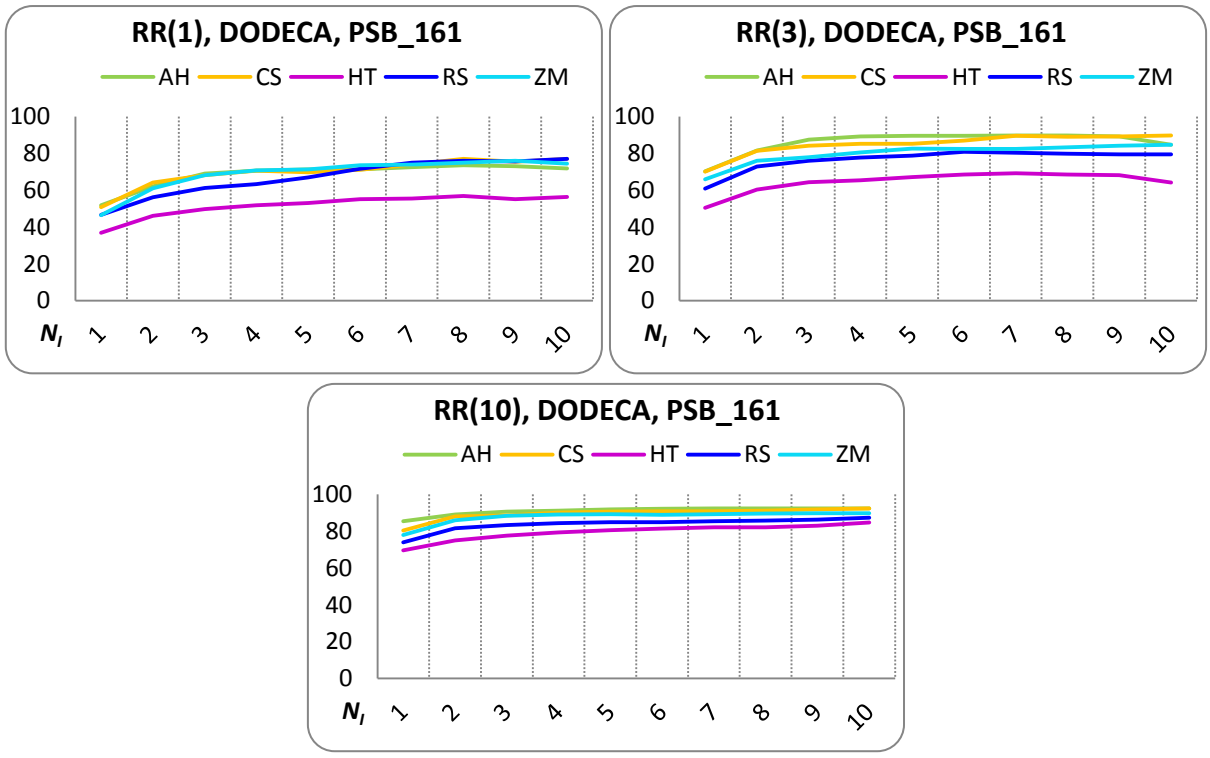



*Figure IV.42 VOSy database: RR(1),RR(3)and RR(10) scores obtained with DODECA strategy and PSB_161 3D models for different $N_I$ per VO query.*

## IV.5. CONCLUSIONS

In this chapter, we have considered the issue of 2D object classification by 2D/3D inference. The objective is to exploit the *a priori* knowledge contained in classified 3D models and, with the help of view-based indexing techniques, to transfer it to unknown 2D objects. Such methods can be applied both to still objects (*i.e.,* objects extracted from still images) and to video objects (*i.e.,* objects extracted from videos and composed of several instances).

A review of the state of the art 2D object classification techniques has been first proposed. Further, we have introduced a 3D-model based object recognition framework which integrates the 2D/3D indexing techniques presented in Section III.2. The result analyser module included in this framework makes it possible to combine several 2D/3D indexing methods. Thus, the possible complementarities between the retained descriptors can be exploited and the performance can be improved.

The experimental evaluation on still objects proved that, just by using simple and fast descriptors, we can reach *RR*(1) score up to 74% and *RR*(3) score up to 86%. The recognition scores can be further improved by 2% to 6% when CS and AH descriptors are combined.

The tests performed on video objects lead to *RR*(3) up to 87.5% when selecting among 23 semantic classes (MPEG7_23) and 97.5% when selecting among 53 categories (PSB_53). In the case of PSB_161, the highest *RR*(3) score was of 90%, while *RR*(10) reached up to 97.5%. Our analyses also proved that disposing of three different instances of a video objects is sufficient to allow correct classification.

In order to make possible the integration of the proposed 2D object recognition framework in real-life applications, an object segmentation module is required. In the next chapter, we propose a semi-automatic segmentation tool, designed to help the user to extract an object of interest from an image.

# V. INTERACTIVE OBJECT SEGMENTATION

**Abstract.** *In this chapter, we present an interactive segmentation method, designed to help the user to extract an object of interest from an image. The proposed approach adopts the scribble-based segmentation paradigm. The user interaction consists of specifying a set of lines, corresponding to both foreground and background scribbles. The segmentation process is based on colour distributions, estimated with Gaussian Mixture Models (GMM). We show that such a technique presents some limitations when dealing with compressed images, even for relatively high quality compression factors: in this case, blocking artefacts may degrade the segmentation results. In order to overcome such a drawback, a modified GMM model, which re-shapes the Gaussian mixture based on the eigenvalues of the GMM components, is proposed.*

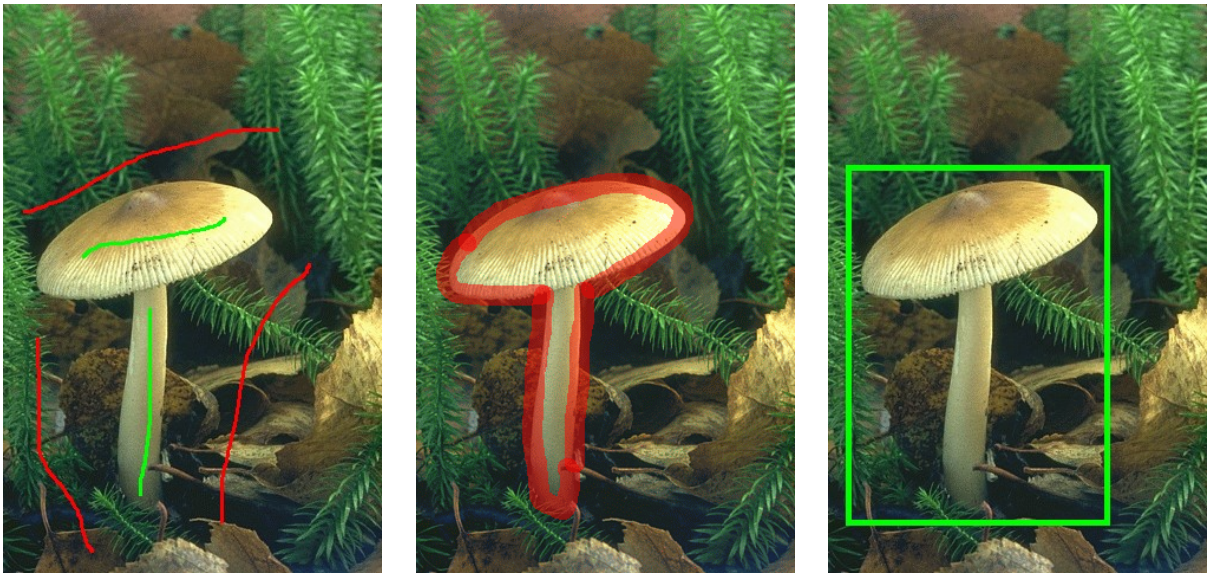
*The experimental evaluation, carried out on a corpus of various images with different characteristics and textures, demonstrates the superiority of the modified GMM model which is able to appropriately take into account compression artefacts.*

**Keywords:** *interactive image segmentation; foreground extraction; Gaussian mixture model;*

## V.1. INTRODUCTION

This chapter tackles the issue of interactive, semi-automatic image segmentation. The objective is to assist the user to extract meaningful objects of interest from a given image, whatever their complexity in terms of colour, texture and shape characteristics, while minimizing the required human interaction. Let us underline that, in order to be able to design a general public tool and to facilitate its adoption in industrial applications, the interaction process should not involve any specific knowledge or abilities from the user.

In recent years, a scribble-based interactive segmentation paradigm has emerged [Protiere07, Bai07, Boykov01, Gulshan10]. The principle consists of interactively specifying a set of scribbles, marking both the object of interest and the background. Such scribble may be arbitrary lines or free-form curves [Protiere07, Bai07, Boykov01, Gulshan10] (Figure V.1a), rough object boundaries [Blake04] (Figure V.1b) or bounding rectangles [Rother04] (Figure V.1c).



**a.** **b.** **c.**

*Figure V.1 Various scribbles encountered in the literature.*
*a. free-form curves; b. rough object boundaries; c. bounding rectangles.*

Nevertheless, when the interaction required from the user is too low, the lack of information has to be compensated by additional *a priori* knowledge, like in [Veksler08]. Here, the user is asked to specify the centre of the convex star shape (*cf.* Section V.2) which is then used to control the segmentation process. However, such a notion is often quite difficult to deal with for non-expert users.

In this chapter, we present a novel segmentation method suited for both expert and non-advised users. The human interaction is limited to sketching line segments over the desired object as well as over parts of the background. The main advantage of the proposed method is its ability

to perform well on both compressed and uncompressed images. This is particularly useful since most of the auto-created image content available today is represented in a compressed form (*e.g.*, JPEG/JPEG2000 for most of the existing commercial cameras).

The remainder of this chapter is organized as follows. In the next section, we present an overview of the state of the art scribble-based segmentation techniques. The proposed method is then detailed in the third section. The experimental evaluation, carried out on a corpus with ground truth and including various objects, is presented in the fourth section. Finally, we conclude the chapter and open some perspectives of future work.

## V.2. RELATED WORK

Among one of the first and in the same time one of the most popular interactive segmentation approaches let us mention the *Graph Cut* technique introduced in [Boykov01]. Here, the user marks some more or less thick scribbles on the image in order to specify the foreground $F$ and background $B$ components. The labelled pixels are used to approximate the image intensity distribution of the $F$ and $B$ components. In order to perform the segmentation, a weighted graph is generated (Figure V.2).

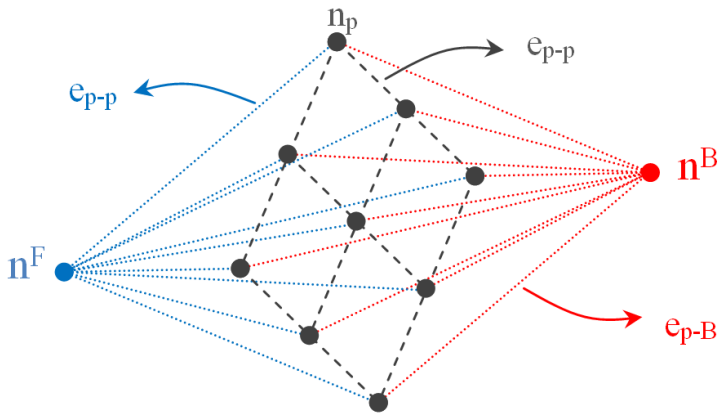



*Figure V.2 Graph representation exploited in the Graph Cut approach.*

The graph contains one node $n_p$ for each pixel $p$ of the image and two terminal nodes ($n^F$, $n^B$) that represent the two possible labels $l \in \{F, B\}$. Each node $n_p$ is connected by an edge to the set of its 8 neighbours and to the two terminals nodes ($e_{p\text{-}F}$ and $e_{p\text{-}B}$ edges). The weight of an edge that connects a pixel with a terminal node ($e_{p\text{-}F,B}$) denotes the probability that the considered pixel belongs to the corresponding $F$ or $B$ plane. The weight $w(p,q)$ of an edge that connects two neighbouring pixels $p$ and $q$ is given by an exponential function that takes into account both the variation of their intensities ($I_p$, $I_q$) and the Euclidian distance $dist(p,q)$ between the considered pixels, as defined in the following equation:

$$w(p,q) = exp\left(-\frac{\left(I_p - I_q\right)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}. \qquad (V.1)$$

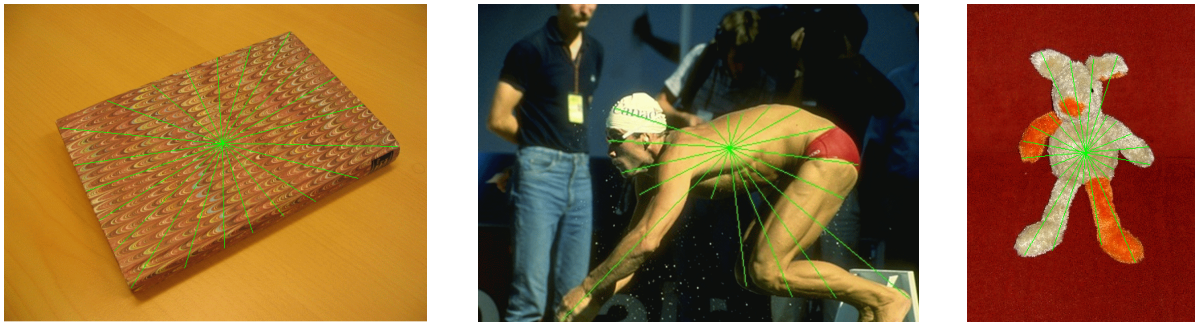
Thus, two similar pixels will have a strong connection, expressed by a high weight of the edge. Finally, the segmentation is performed by searching for the cut of minimal cost [Boykov04] that partitions the graph into two components, one containing the $n^F$ node and the other one the $n^B$ node.

The *Graph Cut* approach has been subject to numerous extensions. Among them, let us mention the *GrabCut* method, introduced in [Rother04]. The human interaction is here reduced to a rough specification of a bounding rectangle, placed around the object of interest. The pixels located inside the rectangle are considered as being part of the foreground, while the external pixels are labelled as background. Two Gaussian Mixture Models (GMM) [Reynolds07] are generated: one for the foreground (*F*) and one for the background (*B*) component. Further, an iterative process is applied. Each *F* and *B* pixel is assigned to the most probable cluster from the corresponding GMM. Then, each component of the two GMMs is re-estimated according to the updated pixel assignment. In the same time, the weights of the graph edges are re-calculated accordingly and the minimum cut segmentation is performed. The iterative process continues using the resulting partition of the *F* and *B* pixels until convergence.

Another extension of the *Graph Cut* approach is proposed in [Veksler08]. Here, some *a priori* information is introduced in the segmentation process. Thus, the authors propose to consider solely objects that can be represented by star-convex shapes [Smith68]. Therefore, the ambiguity is reduced by excluding all the shapes that violate this assumption.

Let us recall that a shape *S* is called star-convex if there exists a point *c* in S such that for any arbitrary point *x* within *S* the line segment from *c* to *x* is completely included within the shape. The star-convexity condition can be re-formulated in the discrete domain with respect to neighbouring pixel as follows. Let us consider *S* a star-convex shape with the centre *c* and *x* a pixel within *S*. If *y* is a neighbour of *x* that belongs to the (*x,c*) segment, than *y* should also belong to *S*. This observation makes it possible to introduce an additional, star-convex shape constraint in the *Graph Cut* process, which consists of forbidding cuts along the (*x,c*) segment.

Compared to *Graph Cut* and *GrabCut*, the *Star-Convexity* approach is significantly less demanding in terms of human interaction, which represents its main advantage. Here, solely the centre of the star-convex shape needs to be specified by the user. As no background pixels are marked, the border of the image is used as background label. However, the star-shape constraint is not appropriate for all the objects that can be encountered in practice, which are not star-shaped (Figure V.3).



**a.** **b.** **c.**

*Figure V.3 Star-shape condition: a. example of star-shaped object. b.&c. example of shapes which do not fulfill the star-shape condition.*

This drawback may be overcome using multiple stars and geodesic star-convexity, as proposed in[Gulshan10]. Here, the segment that connects a given pixel $x$ and the centre $c$ is replaced by the shortest geodesic path $\Gamma(x,c)$ between $x$ and $c$. A shape is geodesic star-convex (*GSC*) if the path $\Gamma(x,c)$ lies inside the shape. Authors define the elementary distance $d(p,q)$ between two neighbour pixels $p$ and $q$ as a function of the Euclidian distance $d_E(p,q)$ and of the image gradient between the two pixels, as described in the following equation:

$$d(p,q) = \sqrt{(1-\gamma_g)d_E(p,q)^2 + \gamma_g \|\nabla I(p)\|^2}, \qquad (V.2)$$

where the parameter $\gamma_g$ weights the Euclidian distance versus the geodesic term.

Moreover, in order to cover a larger range of shapes, the star-convexity is extended to multiple stars. Here, the centre $c$ is replaced by a set of centres $\{c_i\}$. For each pixel $x$, only the closest centre is considered $c_{i(x)}$. Thus, a shape $S$ is geodesic multiple-star-convex if for any $x$ within $S$, the path $\Gamma(x,c_{i(x)})$ lies entirely inside the shape. As the choice of the centres $\{c_i\}$ is a difficult task, authors propose to use the foreground labelled pixels as multiple-star centres. The experimental evaluation presented in [Gulshan10] shows that *Geodesic Star-Convexity* algorithm outperforms *Star-Convexity*, *GraphCuts* and *DistanceCut* methods.

The powerful geodesic distance principle is also adopted, in a different manner in [Protiere07]. Here, authors present a semi-automatic segmentation approach which uses the concept of *adaptive weighted distances*. The algorithm requires two sets of labelled pixels, one for the foreground *(F)* and the other for the background *(B)*, which are roughly scribbled (*i.e.*, free-form curves) by the user in an interactive manner. Each image is described by a combination of $N_{Ch} = 19$ channels consisting of the luminance/chrominance components in the $Y$, $C_b$, $C_r$ colour space, as well as the output of 16 Gabor filters [Kyrki04] (with 4 scales and 4 orientations) applied on the $Y$ channel and aiming at describing the texture information. Based on the values of the labelled pixels, the probability density functions ($PDF_i^l$) are approximated by a Gaussian distribution for each channel $i$ and for each foreground/background label $l \in \{F, B\}$. As the channels that better discriminate between planes are not the same in all cases, a weighting coefficient $\omega_{Ch}^i$ is computed for each channel. The value of this coefficient depends on the

probability to assign a wrong label to a given pixel, according to the PDF of the considered channel. Finally, the global likelihood $P^l(x)$ of each pixel $x$ to belong to the foreground ($l = F$) and to the background ($l = B$) is computed as described by the following equation:

$$\forall\, l \in \{F,\, B\}, \quad P^l(x) = \sum_{i=1}^{N_{Ch}} \omega_{Ch}^i \frac{\mathrm{PDF}_i^l(X_x^i)}{\mathrm{PDF}_i^F(X_x^i) + \mathrm{PDF}_i^B(X_x^i)}, \tag{V.3}$$

where:

- $\{X_x^i\}$ are the channel components of the pixel $x$;
- $N_{Ch}$ denotes the number of channels (19, in this case).

A weight $W^l(x)$ is associated to each pixel $x$ and for each label $l$:

$$\forall\, l \in \{F,\, B\}, \quad W^l(x) = 1 - P^l(x). \tag{V.4}$$

This leads to two complementary, foreground and background weight images. For each pixel $x$, the geodesic distances $d^F(x)$ and $d^B(x)$ to the $F$ and $B$ scribbles are then computed conditionally to the $W^F$ and $W^B$ weight maps, respectively, as described in the following equation:

$$\forall\, l \in \{F,\, B\}, \quad d^l(x) = \min_{s \in \{l\}} \min_{\gamma \in \Gamma(x,s)} \sum_{p \in \gamma} |\nabla W^l(p) \cdot \dot{\gamma}(p)|\,, \tag{V.5}$$

where:

- $\Gamma(x, s)$ denotes the set of all possible paths $\gamma$ between pixels $x$ and $s$;
- $\dot{\gamma}(p)$ represents the path's tangent vector in point $p$;
- $\{l\}$ denotes the set of scribble pixels $s$ associated to label $l$.
- $d^F$ and $d^B$ are the foreground and background geodesic distance maps.

The segmentation is finally achieved by assigning each pixel to the closest scribble pixel, in the sense of the geodesic distance in Equation V.5. Thus the label $l_x$ associated to pixel $x$ is obtained as:

$$l_x = \arg\min_{l \in \{F,\, B\}} d^l(x). \tag{V.6}$$

A similar method, called *DistanceCut*, is proposed in [Bai07]. Here, the representation of the image is reduced to the 3 channels corresponding to the Luv colour space [Tkalcic03]. The colour distribution of the labelled pixels, previously assimilated as a Gaussian, is now approximated in a finer manner, with the help of the kernel density estimation technique described in [Yang03]. Finally, the segmentation is performed in a similar manner to the method in [Protiere07], by assigning each pixel to the closest label via the foreground and background geodesic distances.

In this chapter, we propose a different extension of the scribble-based segmentation method proposed in [Protiere07]. Instead of using kernel density estimation as in [Bai07], we consider Gaussian mixture models for colour distribution estimation, which makes it possible to obtain an adaptive representation, well-suited for characterizing non-uniform regions. Let us note that this has already been suggested as a perspective by the authors in [Protiere07]. However, to our very best knowledge, such an extension has not yet been presented and validated by any of the methods

reported in the literature. We notably show that the direct extension from a single Gaussian model to a GMM leads, in the case of compressed images, to block artefacts that can significantly degrade the segmentation results. In order to overcome this limitation, a modified GMM model, able to deal with compressed images, is further proposed.

Let us first present the GMM-based segmentation approach.

## V.3. GMM-BASED SEGMENTATION

As in [Protiere07], the proposed method starts from two arbitrary sets of labels, marked by the user, indicating the foreground and the background parts. For the user's convenience, a straight line drawing tool is used in order to mark the labels (Figure V.4a). However, the algorithm performs as well when the labels are free-form strokes. The labelled pixels are used to estimate the colour distribution of the foreground and background planes.

The proposed method extends the reference method introduced in [Protiere07]. Instead of modelling the foreground/background PDFs by a single Gaussian function, which can solely approximate roughly the colour distributions, we have considered a GMM model, described in the following section.

### V.3.1. From Gaussian PDFs to GMMs

In order to accurately model the foreground and background PDFs, two dedicated GMMs [Reynolds07] are employed. In our work, we have considered the Luv colour space, because of its recognized perceptual uniformity features [Tkalcic03].

A GMM model, denoted by $g(X)$, is by definition a weighted sum of $N_{GMM}$ multivariate Gaussian functions:

$$g(X) = \sum_{i=1}^{N_{GMM}} \omega^i g_i(X), \qquad (V.7)$$

where:

- $X$ is a D-dimensional vector (in our case $D = 3$ and $X$ stores the Luv values of a given pixel $x$);
- $\omega^i$ is a positive weight associated to the $i^{th}$ component of the GMM;
- $g_i(X)$ is the $i^{th}$ multivariate Gaussian distribution, defined as:

$$g_i(X) = \frac{1}{\sqrt[2]{(2\pi)^D}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(X-\mu_i)^T \Sigma_i^{-1}(X-\mu_i)\right\}, \qquad (V.8)$$

where $\Sigma_i$ is its covariance matrix and $\mu_i$ the mean vector.

Starting from each of the two sets of foreground and background scribbles, the corresponding GMM's parameters (*i.e.*, $\omega^i$, $\Sigma_i$, $\mu_i$) are estimated using the iterative Expectation-Maximization (EM) algorithm [Dempster77, Moon96]. The number of GMM components ($N_{GMM}$)

is adaptively estimated using the Rissanen criterion [Rissanen83], as described in [Bouman97]. The resulting foreground and background probability densities are respectively denoted by $g^F$ and $g^B$.

Let us mention that straight lines scribbles are highly convenient in terms of simplicity of the required user interaction but may lead to a relatively low number of pixels, insufficient to obtain a reliable GMM estimation. In order to overcome such limitation, the user-specified sets of scribbles are enlarged as follows. A parameter $\alpha^l$ is computed for each label $l \in \{F, B\}$ based on the total length $L^l$ of their corresponding scribbles (*i.e.*, number of pixels included in the considered *F* and *B* sets of scribbles). Further, each set of *F* and *B* pixels is enlarged by including, for each considered pixel, a number $\alpha^l$ of its neighbours, which are randomly selected. The parameter $\alpha^l$ is computed as follows:

$$\alpha^l = \min\left(\left\lceil \frac{N_t^l}{L^l} \right\rceil, 8\right), \qquad (V.9)$$

where $N_t^l$ is a parameter that specifies the target number of pixels in each set. The maximal value of the parameter $\alpha^l$ is limited to 8, since a 8-connected neighbourhood is considered.

For each label $l \in \{F, B\}$, the likelihood maps $P^l(x)$ defined in Equation V.3 become:

$$\forall\, l \in \{F, B\}, \quad P^l(x) = \frac{g^l(X_x)}{g^F(X_x) + g^B(X_x)}. \qquad (V.10)$$

The weight maps $W^l(x)$ are further obtained by applying the Equation V.4. The segmentation is then achieved with the help of geodesic distances computed conditionally to the background and foreground weights, as described in Section V.2 (*cf.* Equations V.5 and V.6). Finally, if several connected components are obtained, only those adjacent to a foreground scribble are retained.

The sets of $N^l$ pixels used for foreground and background GMM estimation are illustrated in Figure V.4c and f, respectively. In order to allow the comparison between the input and the output of the GMM estimator, $N^l$ random variables (Figure V.4e. and h.) were generated using the parameters $\{\omega i, \Sigma_i, \mu_i\}^l$ associated to each label *l*. Here, only for visualization purposes, Figure V.4 c, e, f, and h are illustrated in the RGB colour space.

Figure V.5 presents another GMM-based segmentation result. Figure V.5c and e illustrate the distance between each pixel and the closest foreground, respectively background scribble, with respect to the corresponding likelihood maps (Figure V.5b and d). Low values (dark pixels) denote short distances to labelled pixels.

The result of the segmentation process is presented in Figure V.5f. We can observe that, in this case, despite of the richness of the texture information present in the image, the foreground flower has been correctly segmented with the help of solely 5 scribbles (2 for the foreground and 3 for the background).

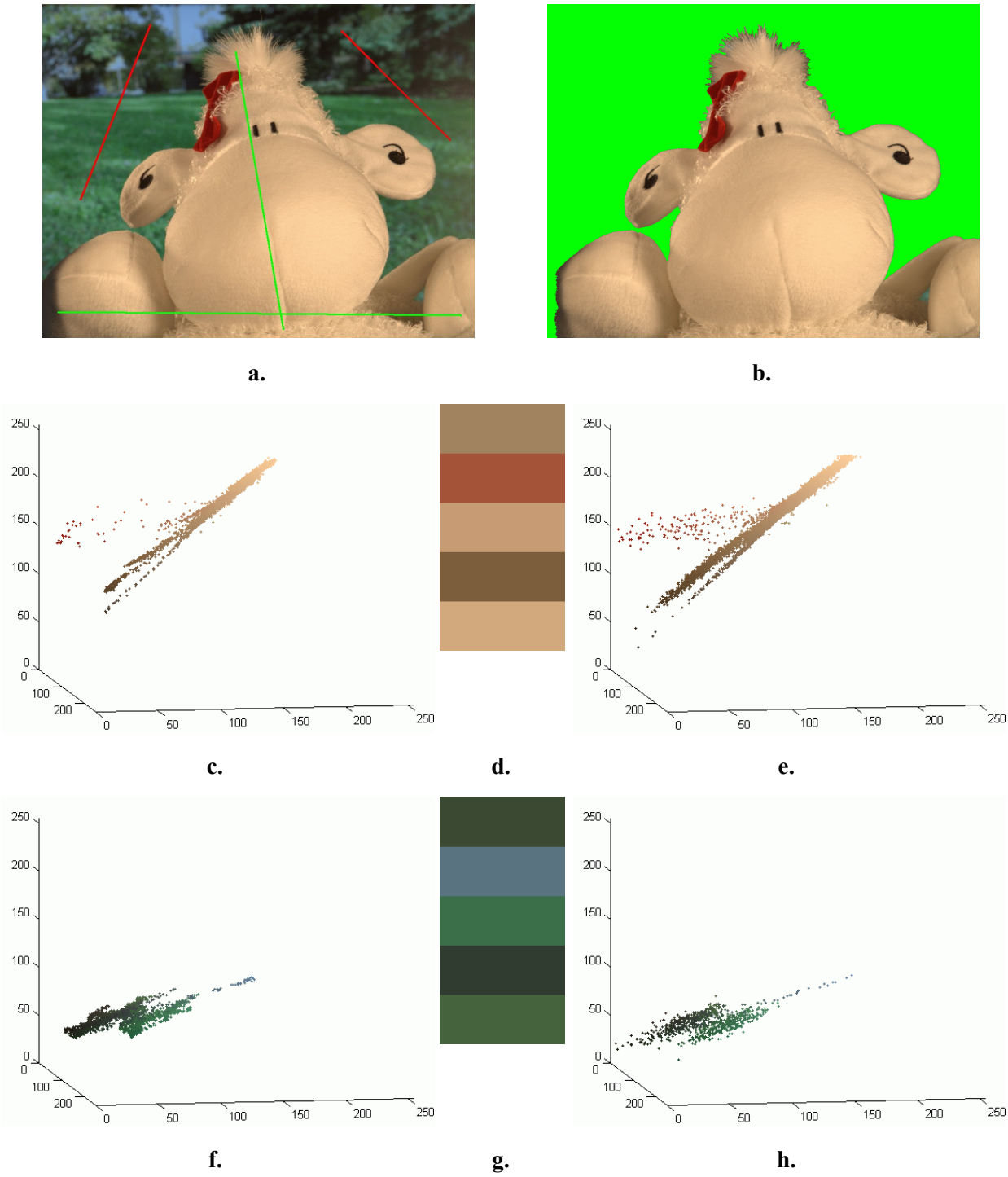



*Figure V.4 Example of GMM-based segmentation.*
*a. the user's scribbles (foreground in green and background in red). b. the segmentation result. c.&f. The set of $N^l$ pixels used for foreground, respectively background GMM estimation; b.&g. the mean values of the GMM components; e.&h. the estimated GMM. $N^l$ random variables were generated using the parameters $\{\omega i, \Sigma_i, \mu_i\}^l$ of the estimated GMM.*

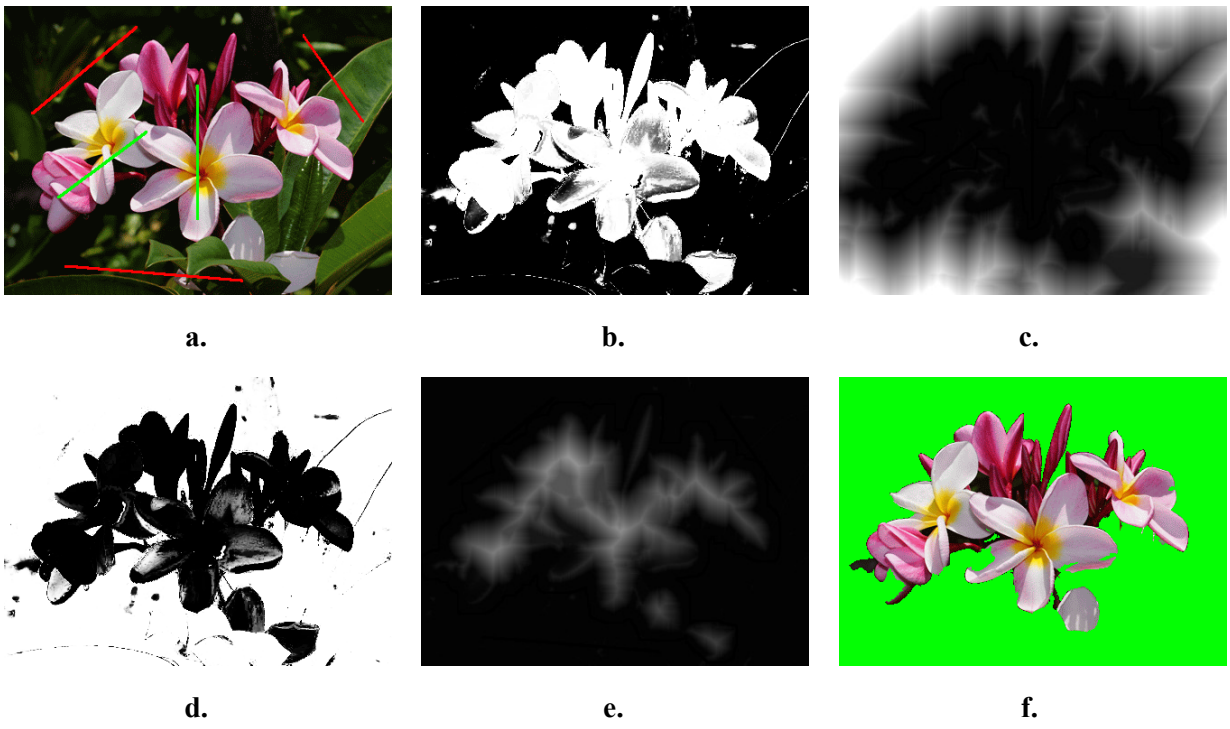



*Figure V.5 The segmentation process.*
*a. The input image and the labels; b. the foreground likelihood map; c. the foreground distance map; d. the background likelihood map; e. the background distance map; f. the obtained segmented object.*

Figure V.6 and Figure V.7 illustrate the segmentation result obtained with simple Gaussian PDFs, as in [Protiere07], and with GMM for two different examples. In the case of Gaussian PDFs, each colour component (*i.e.*, Y, $C_b$, $C_r$) of each label $l \in \{F, B\}$ is modeled by a single Gaussian PDF. The likelihood $P^l(x)$ of the pixel $x$ to belong to the label $l$ is computed as a weighted sum of Gaussian probabilities (Equation V.3).

Both methods provide good results for the first example (Figure V.6), representing a sheep on a relatively uniform grass texture. Here, the background is almost monochromatic (green) and the corresponding $GMM^B$ components (Figure V.6d, bottom row) are very close. Thus, the background distribution can be correctly modelled with the help of simple Gaussian PDFs. In addition, the background and the foreground do not contain similar colours and can be therefore easily separated (as indicated by the likelihood maps presented in Figure V.6e and Figure V.6f). However, the GMM approach performs better at the level of boundaries (where colour blending effects occur), which are more accurately delineated. This behaviour is also true for the region corresponding to the level of the sheep's legs, which are more accurately segmented by the GMM model, able to better take into account the shadowing effects.

The superiority of the GMM approach is even more evident in the second example (Figure V.7), which represents a boy skating. The image is here acquired by night and the background presents relatively important variations in terms of level of luminance. In addition, both the foreground and the background include similar colours and present more complex distributions (Figure V.7d), which cannot be correctly modelled by simple Gaussians. It can be observed on the

foreground likelihood map (Figure V.7f) that the object of interest has similar likelihood to some regions of the background.

However, in the case of GMMs (Figure V.7e), the differences between the likelihood of foreground and background regions are more accentuated, which makes it possible to obtain a more accurate segmentation result (Figure V.7b).

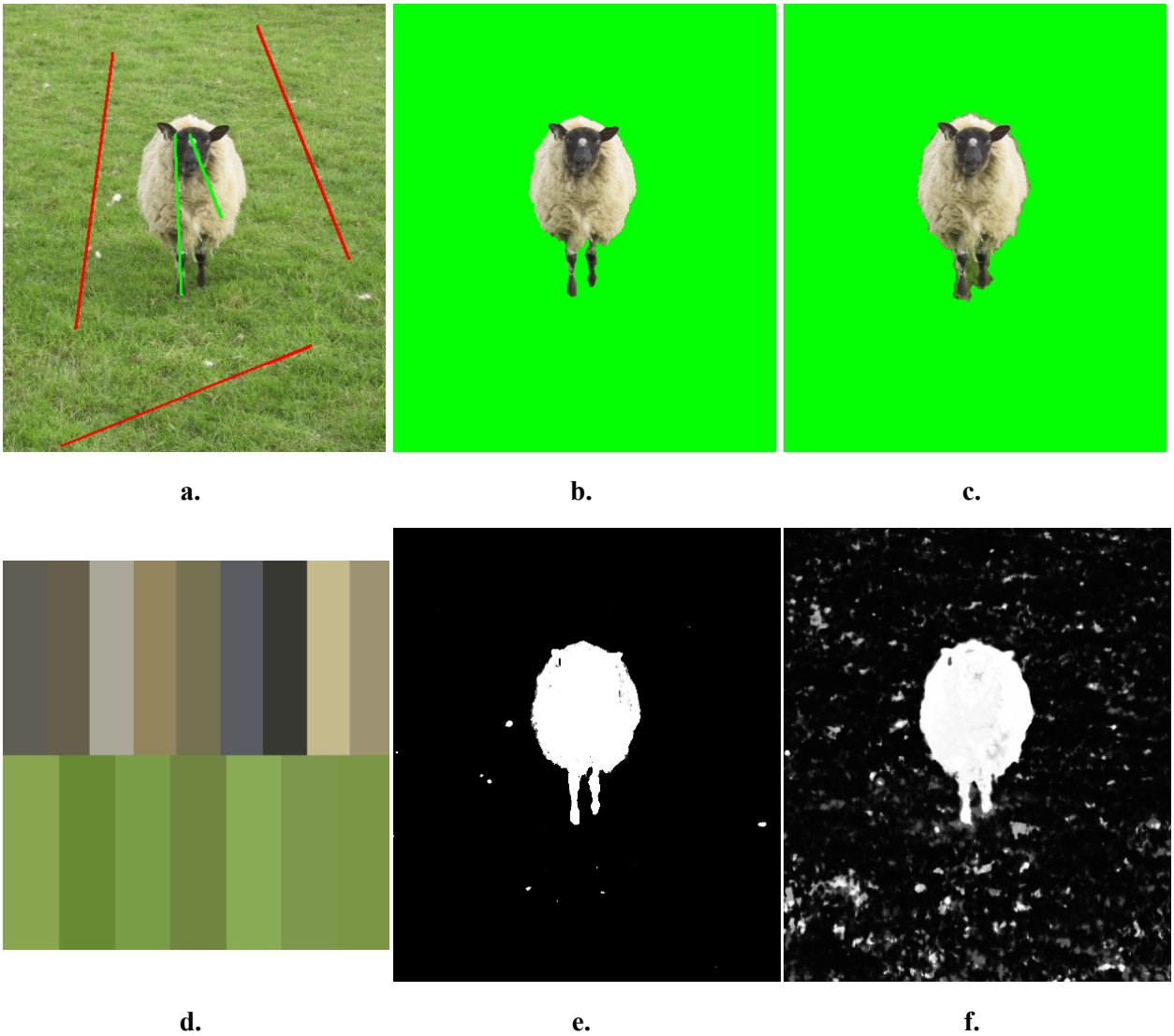



*Figure V.6 Example of segmentation.*
*a. Object of interest representing a sheep on relatively homogeneous grass texture: Input image and scribbles; b. object segmented with GMM; c. object segmented with simple Gaussian PDFs; d. GMM components: foreground (top) and background (bottom); e. foreground likelihood map obtained with GMM; f. foreground likelihood map obtained with single Gaussian PDFs.*

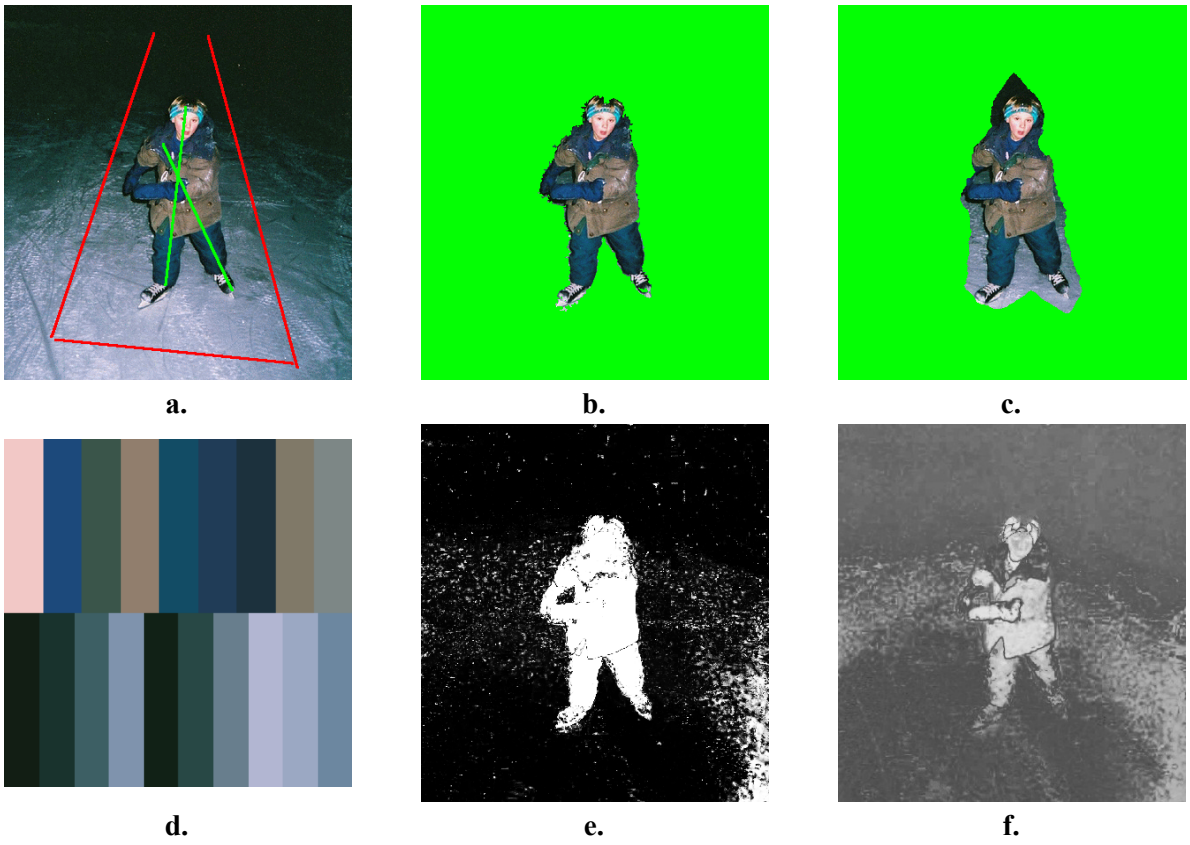



*Figure V.7 Example of segmentation.*
*a. Object of interest representing a boy skating in a night scene: a. input image and scribbles; b. object segmented with GMM; c. object segmented with simple Gaussian PDFs; d. GMM components: foreground (top) and background (bottom); e. foreground likelihood map obtained with GMM; f. foreground likelihood map obtained with single Gaussian PDFs.*

### V.3.2. On the influence of the compression process

When dealing with compressed images, several artefacts, as block or contour effects, may appear. In the case of high quality compression, such artefacts are not visible on the image. However, they can significantly influence the segmentation performances. This problem is illustrated in Figure V.8, which shows the segmentation results obtained by the GMM method described in the previous section for original, uncompressed images, as well as for their JPEG compressed versions (with quality factors of 75%, 50% and 25%). In our work, we have considered JPEG compression because of its high popularity and availability on existing, commercial cameras.

We observe that the segmentation results are strongly affected by the compression process, even when the quality factors are relatively high (*e.g.*, 75%).

In order to analyze this phenomenon, Figure V.9a shows the foreground likelihood map obtained for the 50%-compressed image in Figure V.8. We can observe that the block artefacts are strongly affecting the likelihood map, even if they are not visible on the compressed images. This result is due to the block-based compression mechanism adopted by JPEG but also to the

structure of the GMM model employed. Let us underline that such a phenomenon does not appear in the case of the original method in [Protiere07], where a unique Gaussian function is used to model the corresponding PDFs (Figure V.9b and Figure V.10).

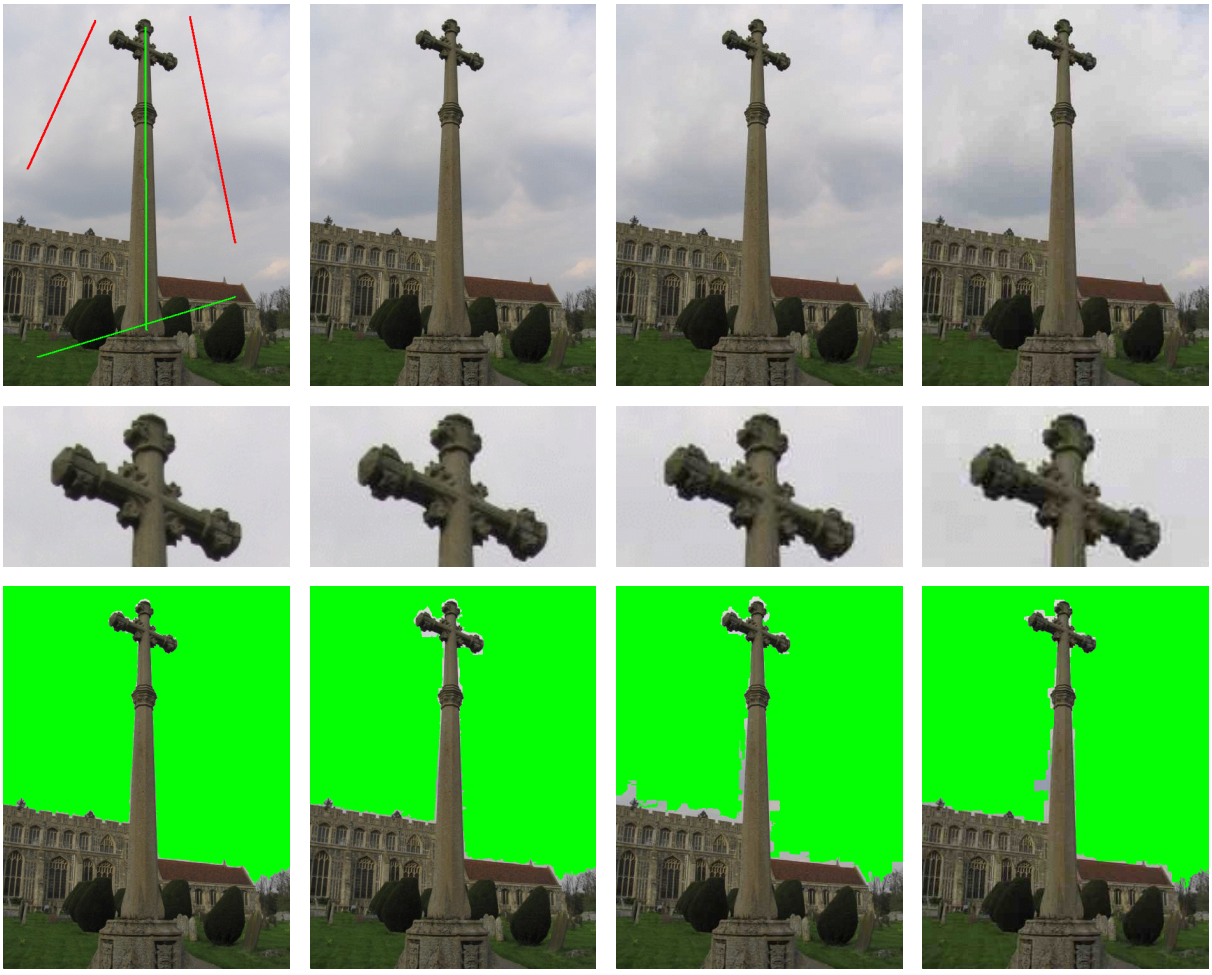
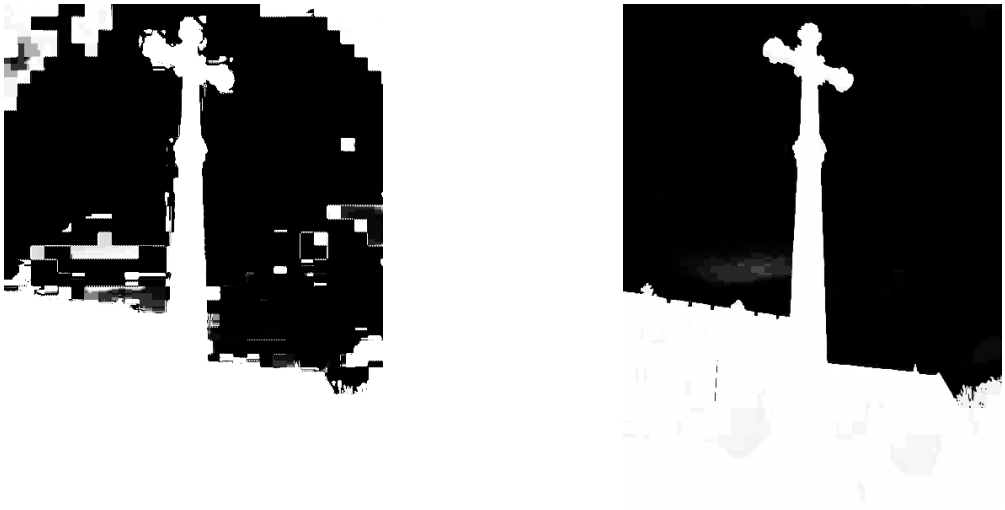



*Figure V.8 Compression influence on the segmentation result.*
*Top row: Input images with different compression levels (quality factor from 100% to 25%); Middle row: details from the compressed images: block effects are not or merely visible (for 25% compressed image); Bottom row: Corresponding GMM segmentation results.*

A finer analysis of such results shows that the block artefacts are caused by the degenerate shape of some components of the GMM model. More precisely, the presence of block effects in the likelihood map (Figure V.9) is caused by the parameters of the GMM. Figure V.11 illustrates the 3D Gaussian distribution of one of the background GMM components associated to the cross image in Figure V.8. The eigenvalues of the corresponding covariance matrix are: $\lambda = [17.26 \quad 0.00125 \quad 0.00125]$.



**a.** **b.**

*Figure V.9 Background likelihood map for the 50% compressed image illustrated in Figure V.8. a. obtained with GMM; b. obtained with Gaussian PDFs.*

100% 75% 50% 25%

*Figure V.10 Segmentation results for uncompressed and compressed images with single Gaussian PDFs.*

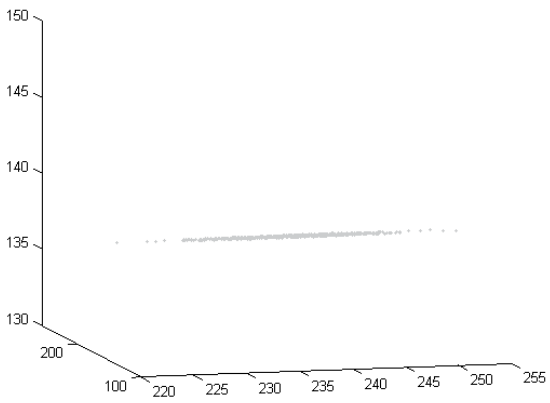



*Figure V.11 Example of elongated Gaussian distribution.*

The last two eigenvalues indicate that the considered Gaussian component has very low variabilities on the corresponding eigen-directions, with respect to the first direction. This explains the elongated shape of the Gaussian function in Figure V.11. As a consequence, the probability field is very high near the centre of the Gaussian distribution and presents a strong decay when moving away in the direction of the last two eigenvectors. Thus, two colours which are close in terms of distance within the Luv colour space can take very different probabilities.

This phenomenon is highlighted in Figure V.12. For simplicity of representation we have considered two 1D Gaussian functions, denoted by $g^B$ (background) and $g^F$ (foreground) and respectively characterized by small and large variances. For the argument $a$ considered in Figure V.12, the probability $g^F(a)$ is higher than the probability $g^B(a)$. This means that the variable a is more likely to belong to the foreground than to the background. However, in terms of distances to the two Gaussian mean values, a is much closer to the foreground than to the background: $d(a, \mu^F) \ll d(a, \mu^B)$. When extrapolating this analysis to the 3D case, this means that a colour which is visually very close to a foreground GMM component can present a high probability to belong to a background GMM component.

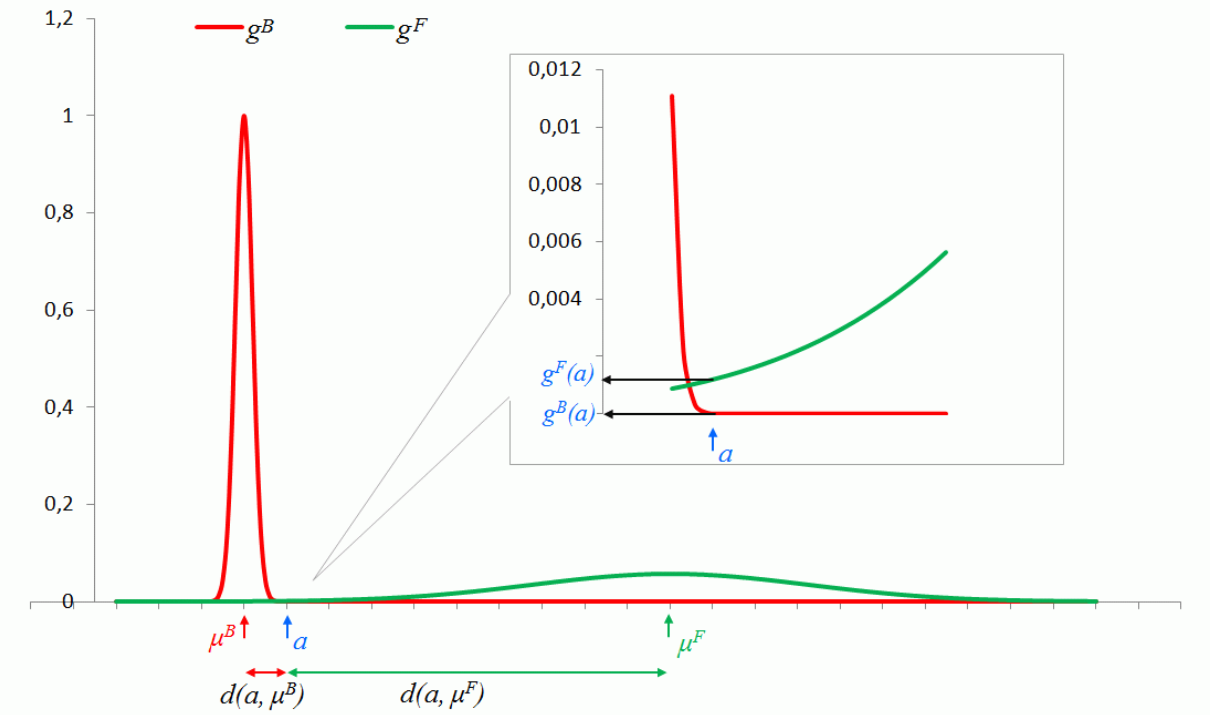



*Figure V.12 Example of 1D Gaussian distributions.*
*with strongly different standard deviations ($\sigma^F >> \sigma^B$). The variable a is much closer, in terms of distance, to the mean value $\mu^B$ of the Gaussian $g^B$ than to the mean value $\mu^F$. However, the probability $g^B(a)$ of belonging to the B distribution is smaller than the probability $g^F(a)$.*

This phenomenon explains the block effect obtained in the case of block-based coding methods such as JPEG. Such compression methods transform relatively uniform regions in blocks of slightly different colours (phenomenon which is often invisible for the human eye, at least for reasonable compression quality factors). However, such small differences may lead to strong variations of the likelihood maps, for the entire blocks. This phenomenon is far from being

isolated and appears frequently in practice, notably in the case of images available over Internet, which are in most cases JPEG compressed. Figure V.13 illustrates a second example, where the object of interest is representing a soft toy placed on a chair.

**a.** **b.** **c.** **d.**

*Figure V.13 Compression block artefacts for a soft toy image.*
*a. the input image; b. Gaussian components determined for foreground and background components; c. the foreground likelihood map; d. segmented object with GMM.*

In this example, some regions of the wall (*e.g.*, the upper left corner, the back of the chair above the soft toy) have high likelihood to belong to the foreground, even if they are visually highly similar to the rest of the background. The segmentation result is illustrated in Figure V.15e. It can be observed that several parts of the background were wrongly extracted as foreground (*e.g.*, the part of the wall above the stuffed toy). The straight contour on the right of the extracted object is also a consequence of the block-based coding. The upper left corner is not detected as foreground (Figure V.13d), although it has a high foreground likelihood, because only the components which are adjacent to an *F* scribble are retained.

In order to overcome this drawback we propose to slightly modify the GMM colour distribution model, as described in the following section.

### V.3.3. Modified GMMs

In order to attenuate the block artefacts, we propose to alter the GMM components that present low eigenvalues. Thus, the Gaussian distributions are expanded (Figure V.14) by modifying the low eigenvalues $\lambda_i$ of each GMM component as follows:

$$\widehat{\lambda_i} = \begin{cases} \lambda_i , & \text{if } \lambda_i > \frac{1}{\beta} \cdot \max_j \lambda_j \\ \frac{1}{\beta} \cdot \max_j \lambda_j , & \text{otherwise} \end{cases} , \qquad (V.11)$$

where $\beta$ is a parameter used to limit the maximum elongation of each GMM component.

The parameter $\beta$ involved should be selected in a manner such that the overall shape of the corresponding Gaussian component should not be severely denatured. In the same time, it should make it possible to overcome the stability issues related to such degenerated Gaussians. In our work, we have experienced various values of the $\beta$ parameter, ranging from 10 to 100, with quite equivalent results in terms of obtained segmentations.

In this way, the variability on the corresponding eigen-directions is increased and thus the variation of the probability is decreased. Figure V.14 illustrates a Gaussian distribution and its modified version in the Luv colour space.

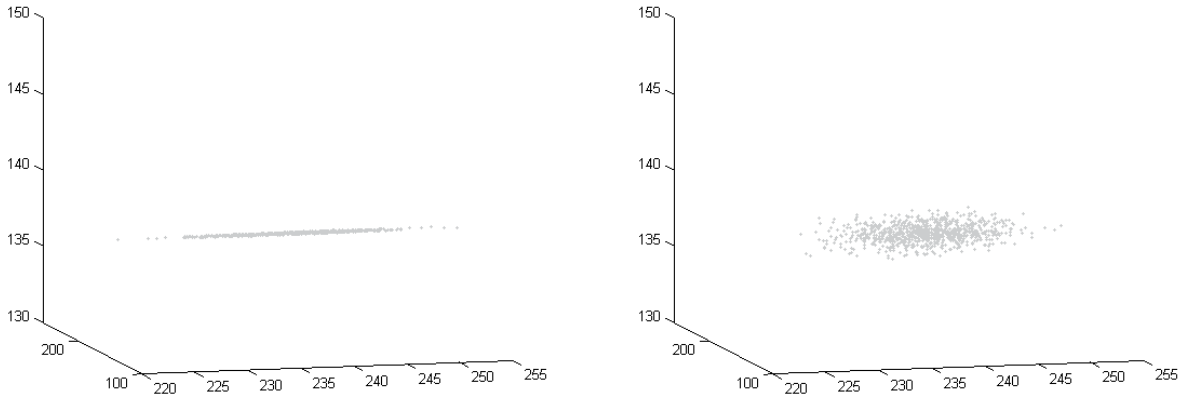



*Figure V.14 Example of Gaussian distribution in Luv colour space.*
*a. original Gaussian distribution; b. modified Gaussian distribution.*

A final refinement has been introduced: we impose likelihood $P^l$ equal to 0.5, for any pixel whose colour has both the foreground and the background probabilities $g^F$ and $g^B$ inferior to a given threshold $\varepsilon$. This makes it possible to eliminate the influence of colours with very small foreground and background probability values, which are often unreliable. In this way, pixels which are highly unlikely to belong to either foreground or background are neutralized, in terms of corresponding likelihood values.

$$P^l(x) = \begin{cases} P^l(x); & \forall l \in \{F,B\}\ g^l(X) > \varepsilon \\ 0.5; & otherwise \end{cases}. \qquad (V.12)$$

In our work, the probability threshold $\varepsilon$ has been set to $10^{-6}$.

The modified GMM representation makes it possible to significantly reduce the block artefacts due to compression, as illustrated in Figure V.15. For the same example previously presented in Figure V.13, we obtain in this case an accurate segmentation result. We can observe that this time the effect of the block artefacts is significantly reduced (Figure V.15b).

**a.** **b.** **c.**

*Figure V.15 Compression artefacts reduction with modified GMM.*
*a. the input image; b. the foreground likelihood map obtained with the modified GMM;*
*c. segmentation result.*

## V.4. EXPERIMENTAL EVALUATION

Experiments have been carried out on the database proposed in [Gulshan10], which includes the GrabCut dataset [GrabCutDB], images from the PASCAL VOC'09 segmentation challenge [Everingham09], and 3 images from the alpha-matting dataset [Rhemann09], for a total number of 151 test images, with resolutions from 284×398 to 800×618 pixels and stored in PNG, BMP and JPEG format. The retained corpus presents a large variety of objects (humans, airplanes, cars, trains, animals, plants, furnish, household items …) acquired indoor, in urban areas or in nature. The ground truth segmentations are also provided.

Concerning the scribble specification, we have considered at most 5 line segments per image. This allows us to keep a relatively low amount of human interaction, which is an important aspect for commercial applications.

Parameter $N_t^l$ (*cf.* Section V.3.1) was set to 2000 (*i.e.*, each scribble pixel set is attempted to be extended to at least 2000 pixels), which ensures sets large enough for estimating GMMs with up to 10 components. The elongation parameter $\beta$ used to modify the GMM (Equation V.11) was set to 50.

The algorithm has been run on an Intel Xeon machine with 2.8GHz and 12GB RAM, under a Windows 7 platform. The segmentation process takes 2 seconds for a 320*480 pixels image and 5 seconds for a 600*450 pixels image, which ensures interactive segmentation rates. Let us note that numerous optimizations are possible, for example using LUT for computing the Gaussian probabilities involved, or exploiting GPU implementations for parallelization.

The performance measure adopted is the *overlap score* (*OS*) already used for evaluation purposes in [Gulshan10, Everingham09]. The *OS* rate measures the number of correctly and incorrectly assigned pixels and it is defined as:

$$OS = \frac{N_{F|F}}{N_{F|F} + N_{F|B} + N_{B|F}}, \qquad (V.13)$$

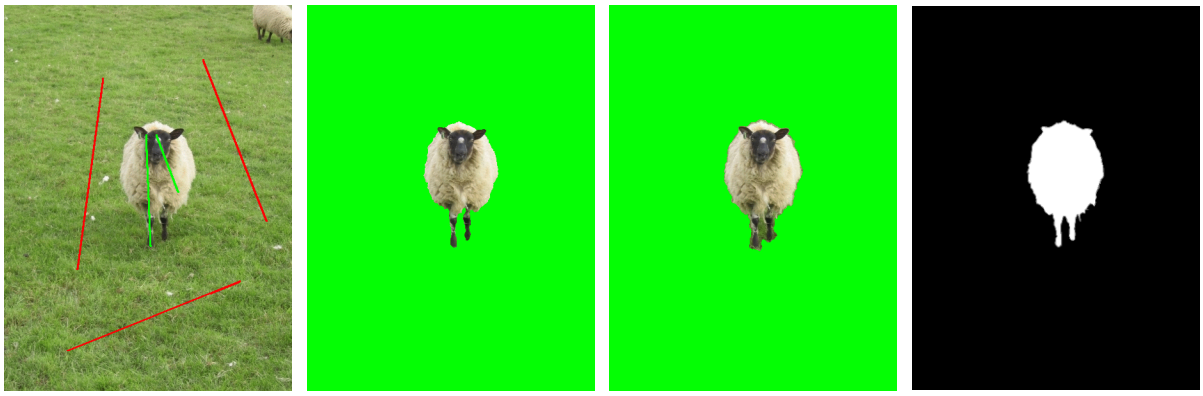
where:

- $N_{F|F}$ is the number of foreground pixels detected as foreground,
- $N_{B|F}$ is the number of background pixels detected as foreground,
- $N_{F|B}$ is the number of foreground pixels detected as background.

We have first analyzed the impact of using a more sophisticated colour model with respect to the reference method in [Protiere07], where single Gaussians, computed marginally on each colour plane, are exploited. Here, the genuine, non-modified GMM representation has been considered. The average overlap score obtained for the whole test data set with GMM is of 85.2%, while with single Gaussian PDF modelling is of only 75.6%. This clearly shows the superiority of a more realistic colour distribution model as the GMM. Several examples of GMM-based segmentation results are illustrated in Figure V.16 to Figure V.19.

Figure V.16 presents an example of a highly accurate segmentation result, where the overlap score is of 95.2%. For the same input, the reference method leads to an *OS* of 92.4%. Figure V.17 illustrates a segmentation result with a score of 93.2%. Here, the few segmentation errors may be explained by the presence of similar materials (*i.e.*, cement) in foreground and background components. In the example illustrated in Figure V.18, the segmentation is less precise (*OS* = 80%) because of the similarity between the black jacket of the character of interest and the dark background. However, the *OS* measure obtained for the reference method in [Protiere07] is here of only 57%. Another difficult example is presented in Figure V.19. The result obtained here can be explained by the fact that the foreground GMM does not contain a white component, suited to model the horse's white spots. The absence of such component is due to the low resolution of the image (the size of the horse's bounding box is of 70×220 pixels) which lead to an insufficient number of white pixels used for GMM estimation. Also, the fact that the foreground and background components include similar colours (*e.g.*, grey) has a negative impact on the segmentation process. However, the obtained *OS* measure is of 77% for the GMM approach, which outperforms the reference method (*OS* = 62%).

Such results demonstrate the superiority of the GMM representation. However, the main limitation is related to the block artefacts which appear in the case of compressed images (*cf.* Section V.3.2, Figure V.13).



**a.** **b.** **c.** **d.**

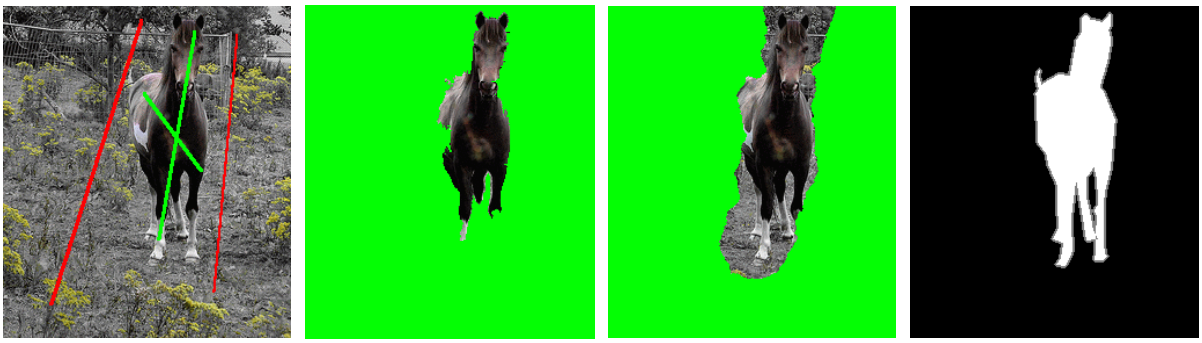
*Figure V.16 Example of segmented object.*
*a. the input image and the used labels; b. the object segmented with the GMM-based method (OS = 95.2%); c. the object segmented with Gaussian PDFs approach (OS = 92.4%); d. the ground truth.*

**a.** **b.** **c.** **d.**

*Figure V.17 Example of segmented object.*
*a. the input image and the used labels; b. the object segmented with the GMM-based method (OS = 93.2%); c. the object segmented with Gaussian PDFs approach (OS = 90.6%); d. the ground truth.*

**a.** **b.** **c.** **d.**

*Figure V.18 Example of segmented object.*
*a. the input image and the used labels; b. the object segmented with the GMM-based method (OS = 80%); c. the object segmented with Gaussian PDFs approach (OS = 57%); d. the ground truth.*



**a.** **b.** **c.** **d.**

*Figure V.19 Example of segmented object.*
*a. the input image and the used labels; b. the object segmented with the GMM-based method (OS = 77.7%); c. the object segmented with Gaussian PDFs approach (OS = 62.7%); d. the ground truth.*

In order to objectively evaluate the improvement introduced by the modification of GMM, we have compressed the images used in the first experiment (JPEG compression with quality factors of 75%, 50% and 25%). Further, we have computed the overlap score when Gaussian PDFs, GMM and modified GMM representations are employed. Table V.1 summarizes the average overlap scores obtained with the three approaches at different compression levels.

*Table V.1 Overlap score on original and compressed images.*

| *Quality factor:* / *Method:* | **100%** | **75%** | **50%** | **25%** |
|---|---|---|---|---|
| **Gaussian PDF** | 75.6 | 75.6 | 75.3 | 75.4 |
| **GMM** | 85.2 | 83.3 | 82.1 | 79.8 |
| **modified GMM** | 85.8 | 84.4 | 83.9 | 82.2 |

Let us observe that the performance of the reference method in [Protiere07] (Gaussian PDFs) is not affected by the compression. This can be explained by the fact that marginal Gaussian distributions provide a global representation, where the various colours are blended together, and which is insensitive to small variations of colour content as those introduced in the JPEG compression process. However, in all cases, the overlap score is inferior to the one obtained with GMM (almost 10% lower for the original images).

The overlap score of the GMM method decreases from 85.2% for original images down to 79.8% on 25%-compressed images. This loss in performance is significantly attenuated in the case of the modified GMM approach, where the OS measure goes down from 85.8% (original images) to 82.2% (25%-compressed images). Over the whole range of compression factors considered, the modified GMM representation offers a slight gain in performances: 1.1% on 75%-compressed images, 1.8% on 50%-compressed images and of 2.4% for a quality factor of 25%.

The superiority of the modified GMM model is even more manifest in terms of visual quality of the obtained segmentations. Figure V.20 to Figure V.23 illustrate the segmentation results obtained with simple GMM on the original images (Figure V.20b to Figure V.23 b) and on

the 25%-compressed images (Figure V.20c to Figure V.23 c) as well as those obtained with modified GMM on the 25%-compressed images (Figure V.20d to Figure V.23 d). We can observe that the compression artefacts mainly affect the contour regions between foreground and background, which become ragged in the case of the genuine GMM approach. In terms of overlap scores such differences are relatively low (1-2%), as the OS measure is computed globally over the object's region of support. However, the visual quality is significantly altered by such a phenomenon. The modified GMM representation makes it possible to overcome this limitation and leads to neat borders, with results approaching those obtained on uncompressed images.

## V.5. CONCLUSION

In this chapter, we have addressed the issue of interactive object segmentation. The proposed method employs a GMM representation for both foreground and background colour distribution estimation. We have observed that when dealing with compressed images, several artefacts may appear. In order to attenuate the negative impact of the compression artefacts on the segmentation process, a modified GMM representation has been proposed. We have shown that the obtained model is well-suited for dealing with compressed images. Thus, the objective experimental evaluation proposed, demonstrated that the block-based coding effect is significantly attenuated and the segmentation quality is improved.

In our future work we intend to adjust the GMM for shadowy regions. The objective here is to extrapolate the colour distribution from the lighted regions to the shaded areas and inversely. A second perspective concerns the integration of a scribble refinement stage, based on an adaptive exploitation of the available contour information.

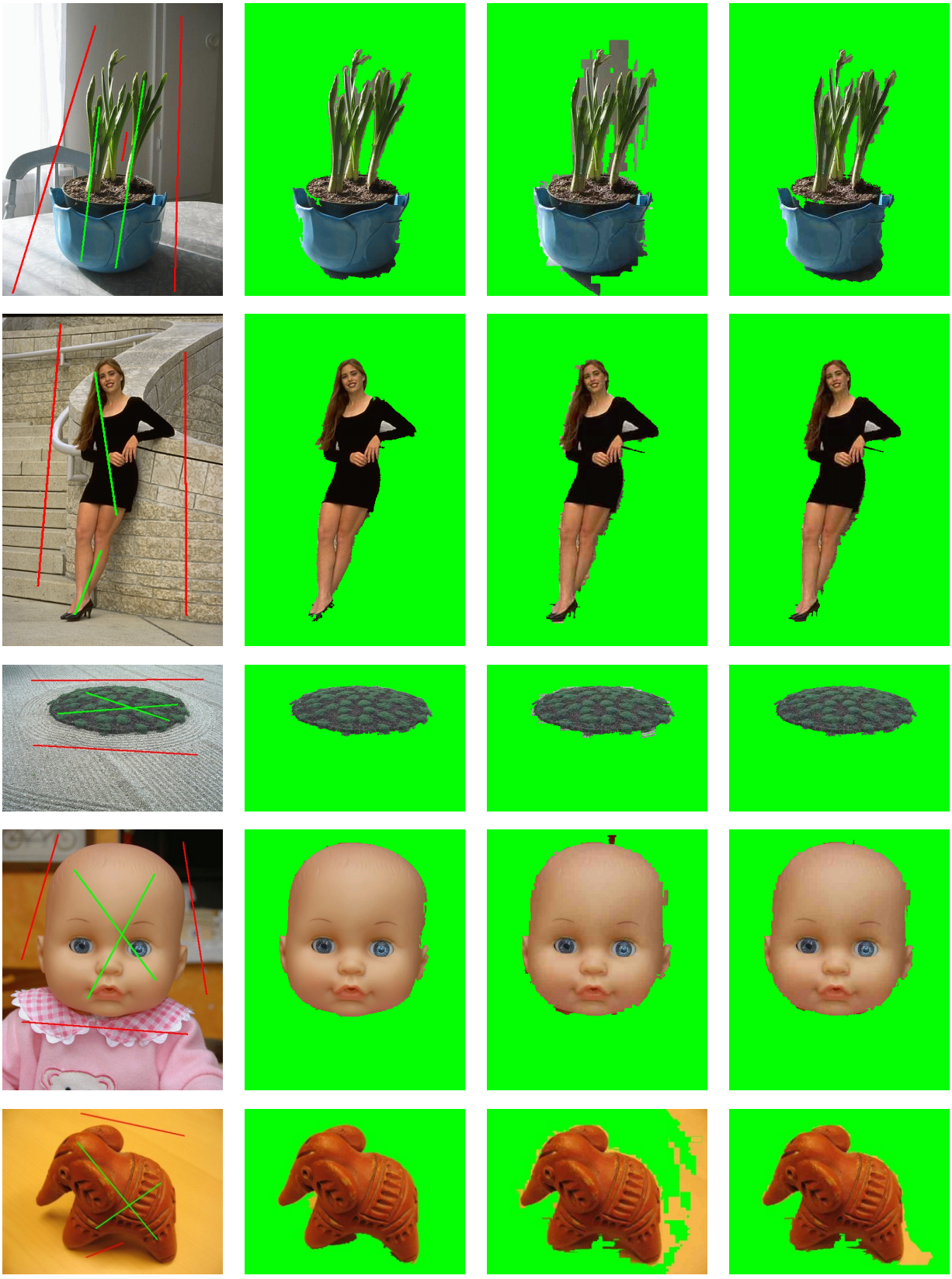


*Figure V.20 The influence of JPEG compression.*
*a. the input images; b. the object segmented from the initial image;*
*c. the object segmented from the 25%-compressed image with simple GMM;*
*d. the object segmented from the 25%-compressed image with modified GMM.*

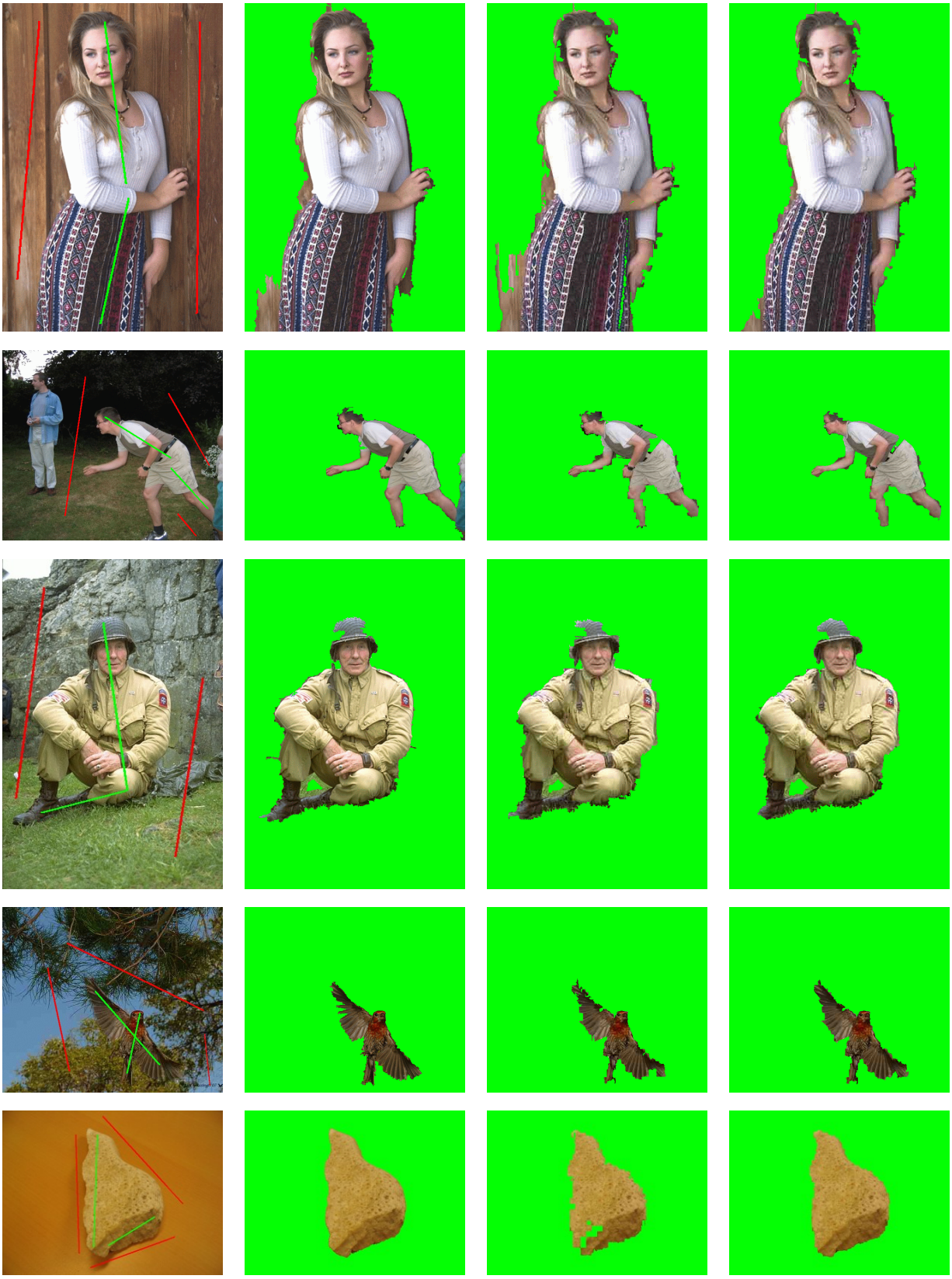


*Figure V.21 The influence of JPEG compression.*
*a. the input images; b. the object segmented from the initial image;*
*c. the object segmented from the 25%-compressed image with simple GMM;*
*d. the object segmented from the 25%-compressed image with modified GMM.*

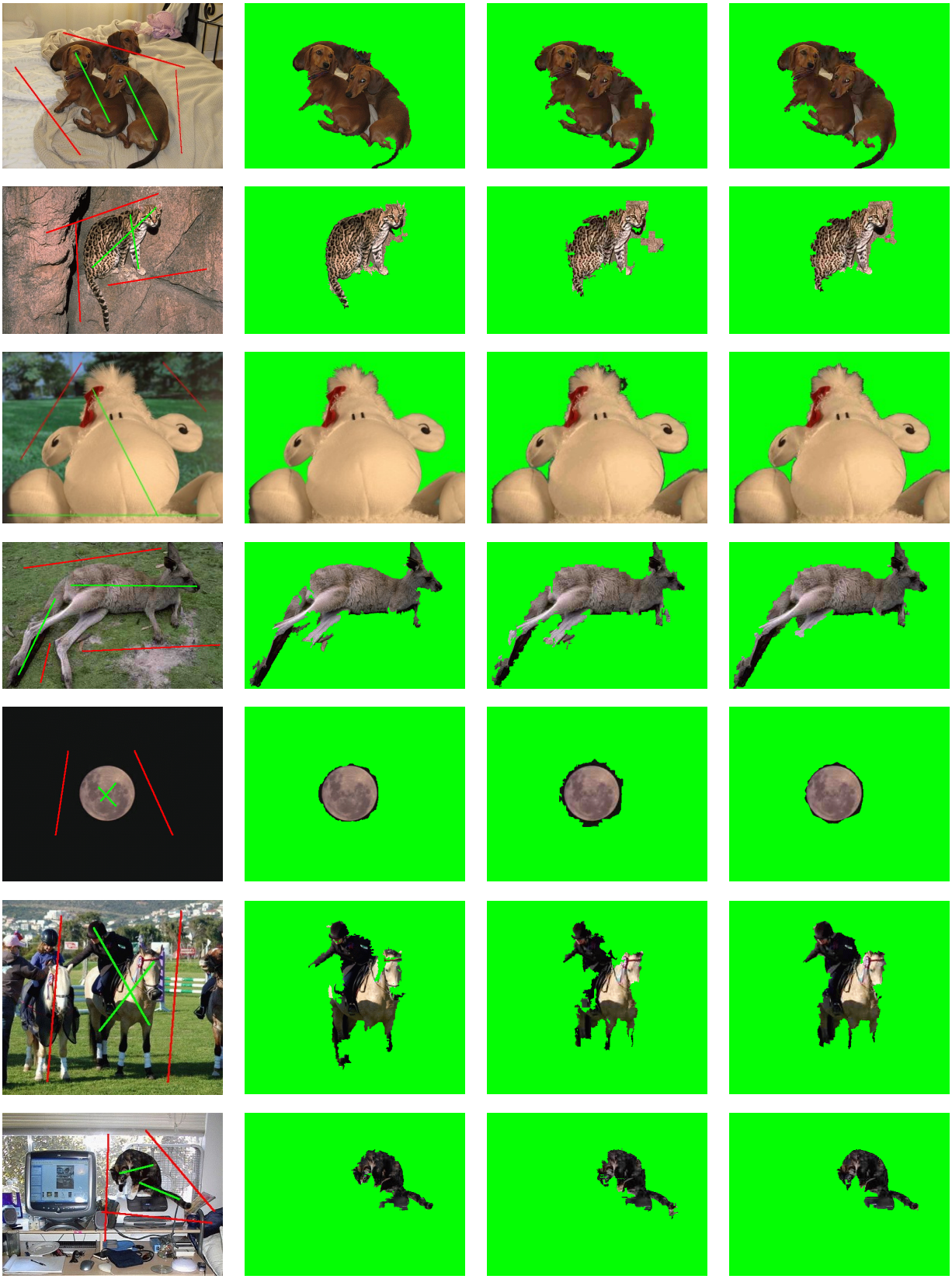


*Figure V.22 The influence of JPEG compression.*
*a. the input images; b. the object segmented from the initial image;*
*c. the object segmented from the 25%-compressed image with simple GMM;*
*d. the object segmented from the 25%-compressed image with modified GMM.*

*Figure V.23 The influence of JPEG compression.*
*a. the input images; b. the object segmented from the initial image;*
*c. the object segmented from the 25%-compressed image with simple GMM;*
*d. the object segmented from the 25%-compressed image with modified GMM.*

# VI. DIANA PLATFORM

**Abstract.** *An important aspect in 2D/3D object retrieval and recognition is to dispose of appropriate user interfaces. For this purpose, we have developed a Web platform designed to help the user in comparing and evaluating the 2D/3D indexing methods adopted in our work. We have considered both 3D model retrieval and 2D object classification frameworks. In addition, the segmentation method presented in chapter V has been integrated in the platform, in order to allow the users to test the system on their own images. The 3D models exploited in our work can be visualized online with the help of a 3D model viewer integrated in the interface. A review of the 2D/3D indexing principle and methods adopted in our work is also proposed.*

**Keywords:** *Graphical user interfaces, SLQ databases, Web services, Web platform;*

In order to facilitate the access to our work, we have developed the so-called DIANA (*Digital Image Analysis aNd Annotation*) Web platform. DIANA integrates the developments proposed in this thesis, including a content-based 3D model search engine, a 2D object classification tool and a segmentation/recognition tool. DIANA is conceived as a Web application, which can be accessed remotely from various devices. Since all computationally intensive tasks (*e.g.*, segmentation, descriptor extraction, similarity computation…) are handled on the server, the users do not need to have high system requirements and can even access the application on mobile devices.

Let us start by presenting the architecture of DIANA platform.

## VI.1. ARCHITECTURE

The proposed architecture is designed to function as a multi-platform, multi-server application. Currently, the platform includes three main components (Figure VI.1): a Web server, a user interface and a SQL server.


*Figure VI.1.* DIANA *platform architecture.*

**The Web server** (Figure VI.2) runs a Web application, written in PHP, which manages user requests. It also includes dedicated software which implements our segmentation and recognition approaches (S&R applications module). The S&R applications were developed in C++ and are used by the segmentation and recognition tool. For any request, the first S&R application (Figure VI.2) to be run is the segmentation approach. It takes as input an image and the associated labels and provides as output the 2D silhouette of the desired object. Further, the recognition approach determines which semantic labels correspond to the silhouette and sends them to the Web application. The recognition process includes the MPEG-7 software for the CS and RS description. In the case of AH, HT and ZM descriptors, our own implementation has been considered. A dedicated database (hosted on the Web server) stores all the descriptors extracted during the 2D/3D indexing of the retained 3D models and exploited by the recognition process. As designed, the recognition tool easily supports further extensions, *e.g.*, new descriptors and similarity measures. Moreover, let us emphasize that within this framework, the various implementations can be deployed on various servers/platforms, whatever the underlying operating system.

The Web server also hosts the various multimedia data (3D models, 2D images) to be displayed to the user.

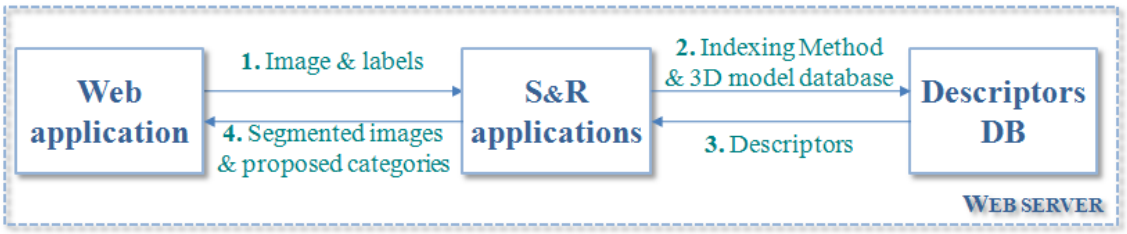



*Figure VI.2 The Web server.*

**The SQL server** runs a MySQL database which contains a set of tables storing the output of the 3D model retrieval/2D object recognition tool for each possible query. It manages the SQL queries issued by the PHP application.

**The user interface** (UI) supports all major browsers. It makes use of JavaScript in order to provide a better user experience. For 3D model rendering purposes, the popular Cortona 3D viewer plug-in [Cortona3D-Website] has been considered. Alternatively, the application can be adapted to use WebGL in order to avoid depending on a specific plug-in like Cortona.

## VI.2. FUNCTIONALITIES

DIANA offers the following functionalities: 3D model viewing and examination, content-based 3D model retrieval, 2D object classification and real time segmentation and classification.

**The 3D model examination** is available on the *3D model databases* page (Figure VI.3). All 3D objects included in the two datasets exploited in our work (*i.e.*, MPEG7 and PSB – section III.5) can be examined with the help of Cortona3D viewer [Cortona3D-Website].

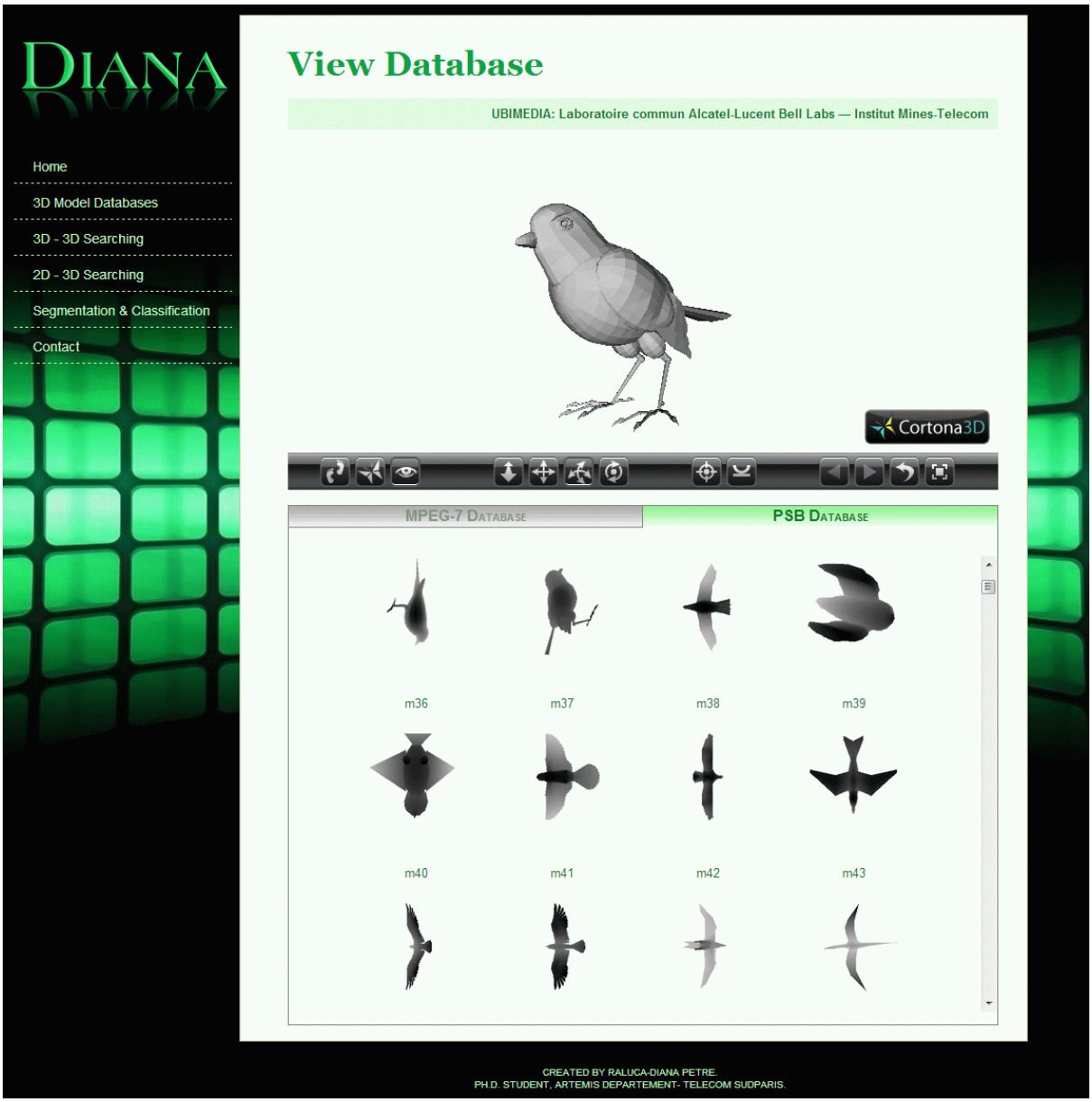



*Figure VI.3* DIANA *Web platform: the 3D models databases page.*

**The content-based 3D model retrieval** tool is available on the *3D – 3D searching* page, illustrated in Figure VI.4.

A typical 3D model retrieval operation consists of specifying one or more indexing methods, one of the two 3D model databases and a 3D object used as query. These inputs are sent through a HTTP GET request to the Web application, which in turn queries the MySQL database according to the user's choice. Next, the Web application displays the images corresponding to the list of sorted models retrieved from the MySQL database.

The retrieved models can be examined with the help of Cortona3D viewer and visually compared with the query object. Thus, the performance of different descriptors and/or selection strategies can be easily analysed by the user.

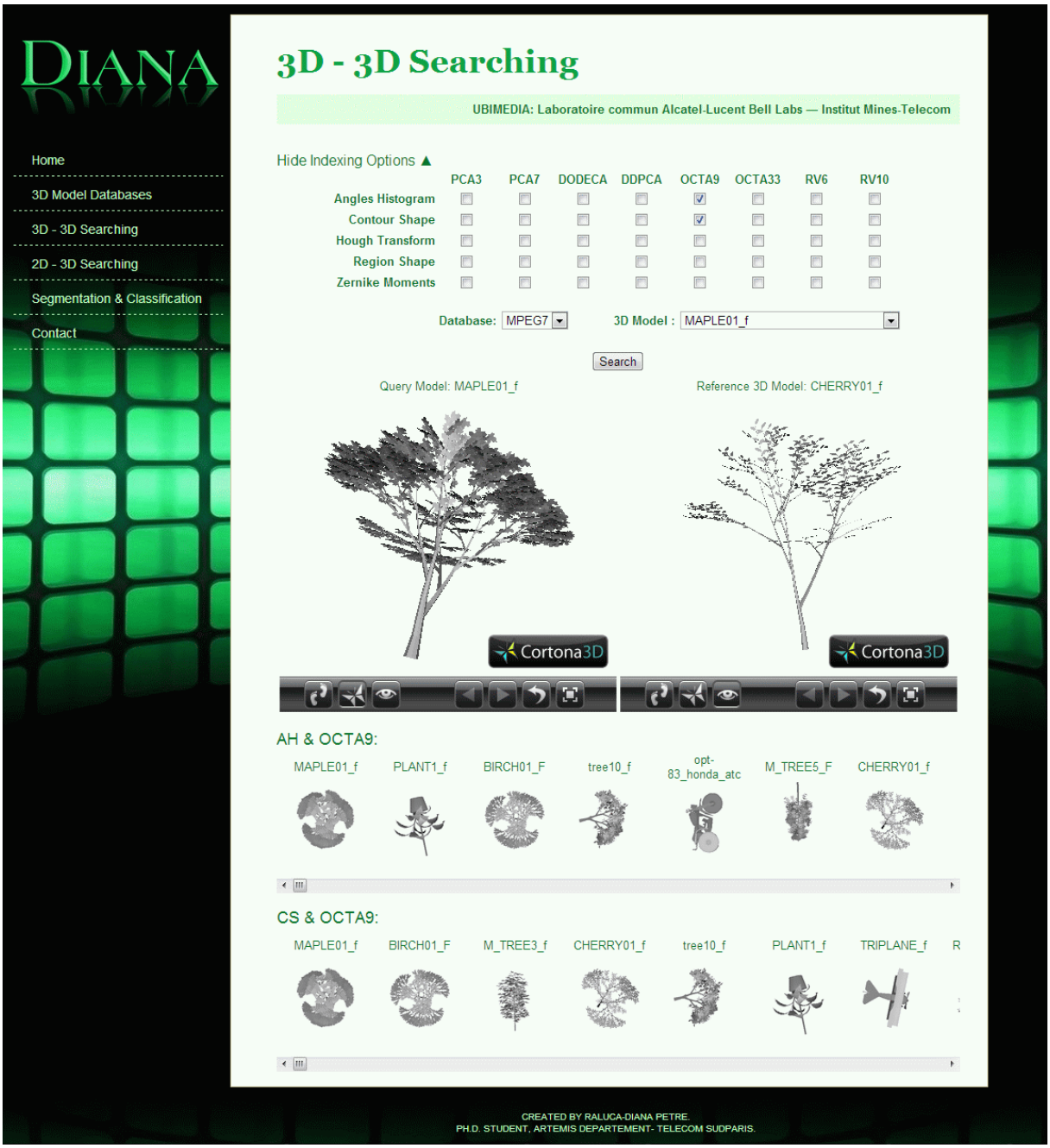



*Figure VI.4* DIANA *Web platform: the 3D – 3D searching page.*

The next functionality is the **2D object recognition**, available on the *2D – 3D searching* page. The request operations are similar to those in the previous case, except that the query is represented as a 2D object.

The system returns as response at most three proposed categories (represented by suggestive, symbolic images) and also the list of 3D models sorted by decreasing order of similarity. Here again, the retrieved 3D models can be examined with the help of Cortona 3D viewer. A 2D query example is illustrated in Figure VI.5.

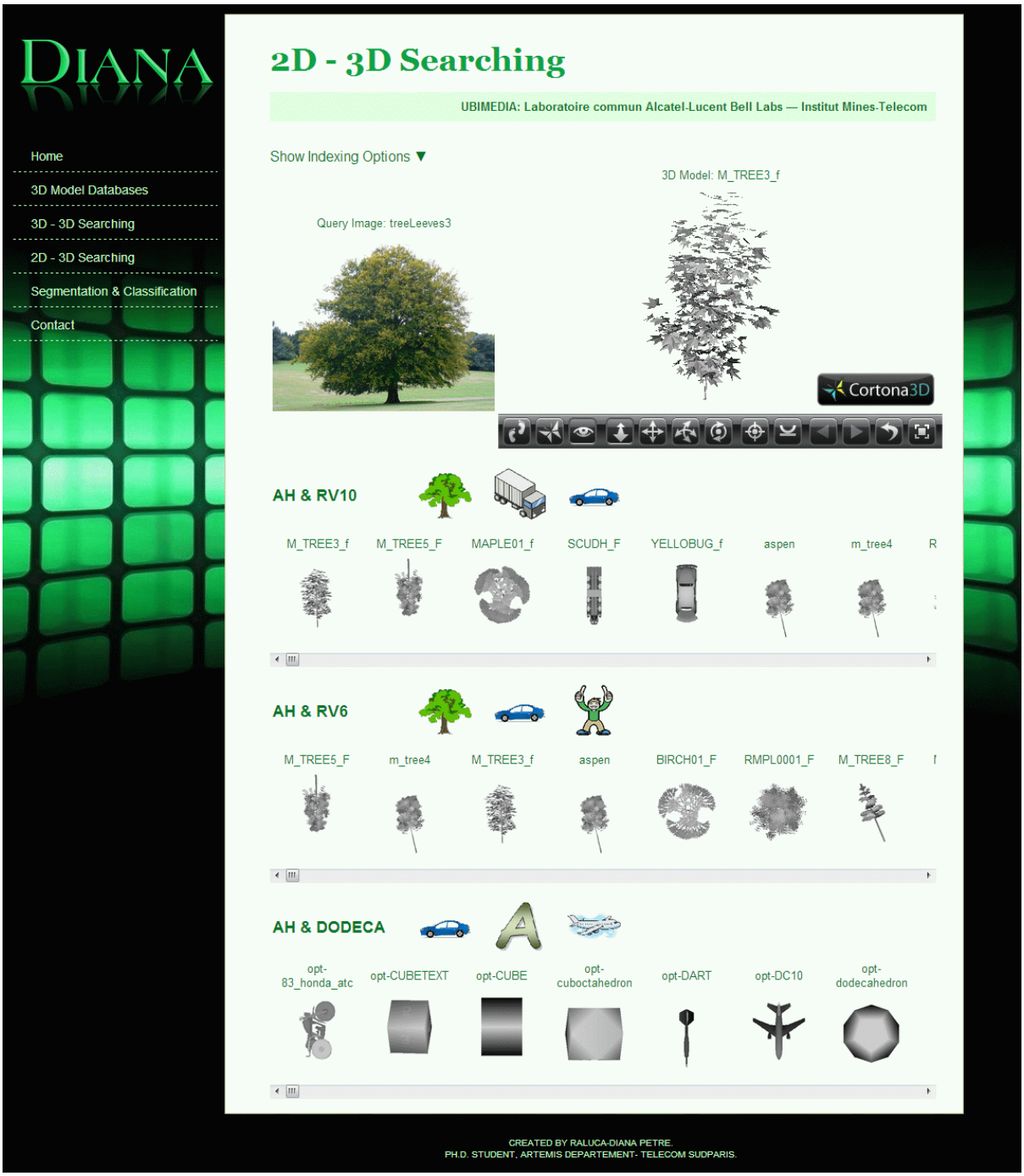



*Figure VI.5* DIANA *Web platform: the 2D – 3D searching page.*

Finally, the **Segmentation and Classification** tool allows user to test the system by uploading his own images. First, the user uploads the desired image on the server through a HTTP POST request. Next, both foreground and background scribbles are specified with the help of a JavaScript straight line drawing tool. We have chosen to draw straight lines and not free form strokes because it requires less gesture precision. When the *Segment & Recognize* button is clicked, the Web application performs an external system call to run the segmentation. Once the object is extracted from the image, the recognition application is employed in order to extract the 2D shape descriptor of the object, to compare it to all 3D models (whose descriptors were

previously extracted and stored on the server) and to determine its semantic class. The Web server then returns an image representing the segmented object and the list of detected categories (Figure VI.6).

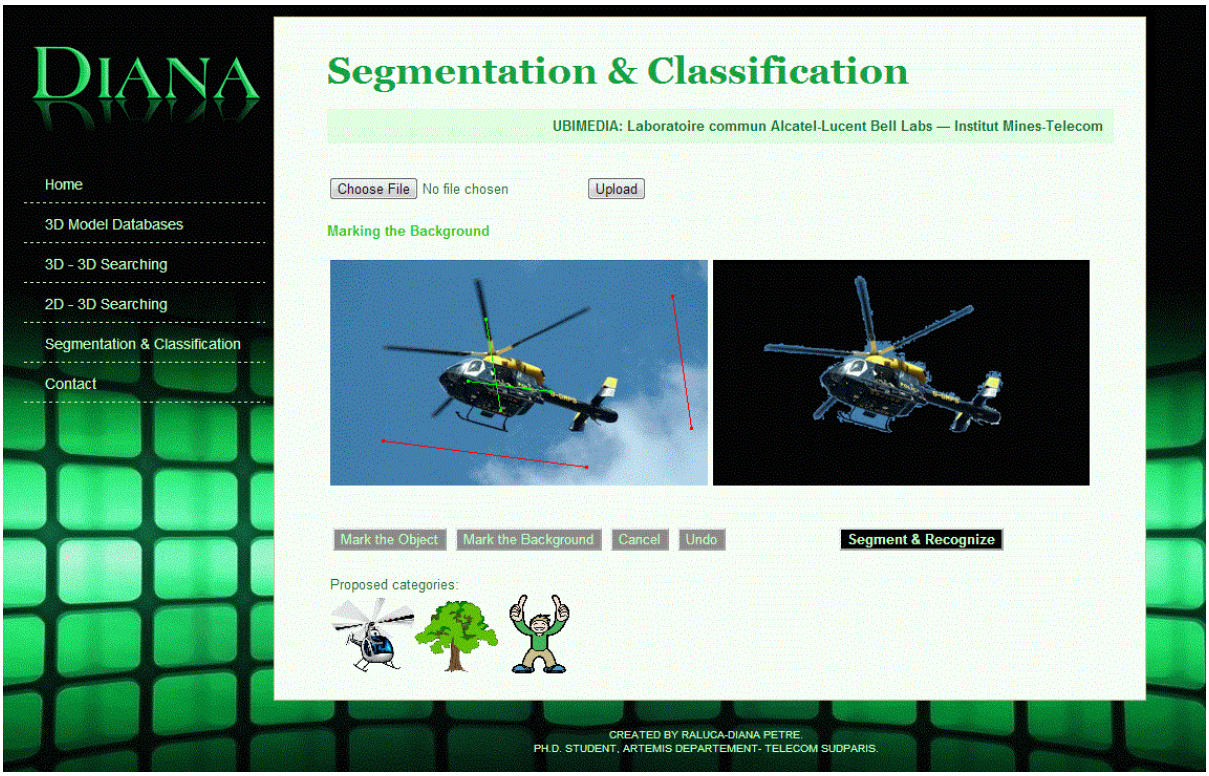



*Figure VI.6* DIANA *Web platform: the segmentation and classification page.*

# VII. CONCLUSIONS AND PERSPECTIVES

In this thesis we have proposed a novel 2D object recognition framework, specifically designed for real time applications. For this purpose, the knowledge inference between 3D and 2D content was exploited with the help of the view-based description.

In order to find the more suited 2D/3D indexing method for our application, we have considered several viewing angle selection strategies and 2D shape descriptors. Besides the state of the art projection methods adopted, we have also proposed a new approach which performs an intelligent selection of views (RV). It exploits a clustering algorithm, similar to the k-means method, intended to select a subset of views with no redundancy. We have also introduced a new contour-based descriptor, so-called *Angular Histogram* (AH). It offers a compact representation of both local and global features of the shape. AH is specially suited for 2D/3D indexing and real time applications because it allows fast extraction (6.6ms) and very quick similarity computation (0.17μs).

Two 3D model repositories were employed in our work: the MPEG-7 dataset and the Princeton Shape Benchmark. In order to better understand the results obtained by exploiting this object (for both 3D model retrieval and 2D shape classification), we have proposed an analytical evaluation protocol for the analysis of the intra- and inter-class variability. This analysis provided some objective information about the adopted databases as well as a first evaluation of the retained indexing methods. Thus, the results have shown that contour-based descriptors (*i.e.,* AH and CS) have a similar discrimination power for all categories of models, while the region-based descriptors (*i.e.*, HT, TS and ZM) are globally less discriminant and seem to advantage some individual categories.

The various 2D/3D indexing methods adopted were first evaluated within the framework of 3D model retrieval. The results have shown the superiority of contour-based descriptors with respect to those exploiting the region support function. The descriptor introduced in our work (*i.e.*, AH) lead to results slightly superior to the one adopted within the MPEG-7 *MultiView* DS (*i.e.*, CS) (about 2%-3% in terms of FT and ST scores). Moreover, AH also present a significant

computational advantage with respect to CS, the associated similarity measure being 400 faster than the one associated with the CS descriptor.

Two different matching approaches, so-called *diagonal* and *minimum*, have been considered. For the projection methods exploiting a canonical representation of 3D models (*i.e.*, model alignment with PCA), the *diagonal* matching approach lead to results which are slightly superior (about 2% of gain in terms of both FT and ST scores) to those obtained with the *minimum* strategy that we proposed. However, *minimum* is more appropriate for projection methods like DODECA, RV6 and RV10 or in those cases where the PCA alignment fails (*cf.* section II.1.1.2).

The selection of representative views – performed to eliminate the redundancy within the set of projections – leads also to an unpleasant effect. The sets of views obtained with RV strategies suffer of lack of consistency. Thus, two similar objects can be represented through perspectives which are different and, therefore, cannot be matched correctly. This explains why RV methods lead to results similar to strategies which extract an equivalent number of views with less computation effort.

However, the representative views selection shows its interest within the proposed framework of 2D shape recognition. For classification purpose, RV6 lead to results similar to OCTA33, while involving 5 times fewer views (and, implicitly 5 times less computation complexity).

The experiments concerning the recognition framework lead to promising results, with $RR(1)$ scores on real images up to 74% for still objects and up to 85% for video objects. When three categories are accepted as response, the same scores are up to 86%, respectively 93%. The experiments have also shown that further increasing the number of instance per video object does not lead to a significant improvement of the recognition process.

However, the performance of the classification system can be improved by exploiting in the same time several descriptors. In the case of combined description methods, the proposed framework presents the advantage of allowing parallelization up to the *Results Analyser* module.

Compared to the state of the art 3D model based classification methods, our approach was designed and tested for a large variety of semantic classes. However, its main limitation concerns the occluded objects, whose class is harder to determine from incomplete shape information.

Our perspectives of future improvements of the recognition system concern the elaboration of a post-filtering module. The aim of such module would be to reduce the number of proposed categories from $N_{RMC}$ to 1. By involving only a limited number of objects and classes, more sophisticated and computationally expensive techniques can be considered in this framework. Within this context, a possible solution in eliminating the shape ambiguities is to exploit the internal contours of the objects.

A second perspective concerns the estimation of the object's pose. The video object recognition could also be improved by introducing a pose coherence criterion relying on the assumption that the pose of the object cannot vary drastically between consecutive frames within

the same video shot. Another perspective is to speed up the recognition process by introducing an early rejection step.

In order to integrate the approach on a completely automatic annotation system, the segmentation input – specified in the current framework by the user – should be automatically determined. In the case of video objects, a solution would be to exploit the motion consistency of the different regions composing the image. In addition, such motion information can also improve the accuracy of the segmentation results. Another solution is to exploit the saliency information in order to extract the labels required by the proposed segmentation approach.

The object extraction methods proposed in this thesis was specially designed to overcome the compression artefacts. We shown how, by remodelling the shape of the GMM components, the effects of the block and contour artefacts are attenuated. Future work can concern a similar adjustment of the GMM, intend to extrapolate the colour distribution from lighted regions to shaded areas and inversely. Another perspective concerns the integration of a scribble refinement stage, based on an adaptive exploitation of the available contour information.

Finally, let us recall that the various methodologies developed in this thesis have been integrated within a Web platform, called DIANA (*Digital Image Analysis aNd Annotation*).

# LIST OF PUBLICATIONS

## JOURNALS

Petre R.-D., Zaharia T., "3D Model-based Semantic Categorization of Still Image 2D Object", *International Journal of Multimedia Data Engineering and Management (IGI Global)*, pp. 19-37, Vol. 2, Issue 4, 2011.

Sambra-Petre R.-D., Zaharia T., "Scribble-based Object Segmentation with Modified Gaussian Mixture Models" – submitted to *Pattern Analysis and Applications*.

## CONFERENCES

Petre R.-D., Zaharia T., "2D/3D semantic categorization of visual objects", *20th European Signal Processing Conference (EUSIPCO 2012)*, pp. 2387-2391, Bucharest, Romania, August 2012.

Petre R.-D., Zaharia T., "3D models-based semantic 155abelling of 2D objects", *International Conference on Digital Image Computing: Techniques and Applications* (*DICTA 2011*), pp. 152-157, Noosa, QLD, Australia, December 2011.

Petre R.-D., Zaharia T., "Semantic labelling of 2D objects with 3D models", *Fifth IEEE International Conference on Semantic Computing (ICSC 2011)*, pp. 419-423, Palo Alto, CA, USA, September 2011.

Petre R.-D., Zaharia T., "Still Image Object Categorization Using 3D Models", *The 1st IEEE International Conference on Consumer Electronics (ICCE 2011)*, pp. 347-351, Berlin, Germany, September 2011.

Petre R.-D., Zaharia T., "3D model-based still image object categorization", *Proceedings of SPIE Conference on Mathematics of Data/Image Pattern Coding, Compression, and Encryption with Applications XIII*, Vol. 8136, pp. 81360C, San Diego, CA, USA, August 2011.

Petre R.-D., Zaharia T., "An experimental evaluation of view-based 2D/3D indexing methods", *IEEE 26th Convention of Electrical and Electronics Engineers (IEEEI 2010)*, pp. 924-928, Eilat, Israel, November 2010.

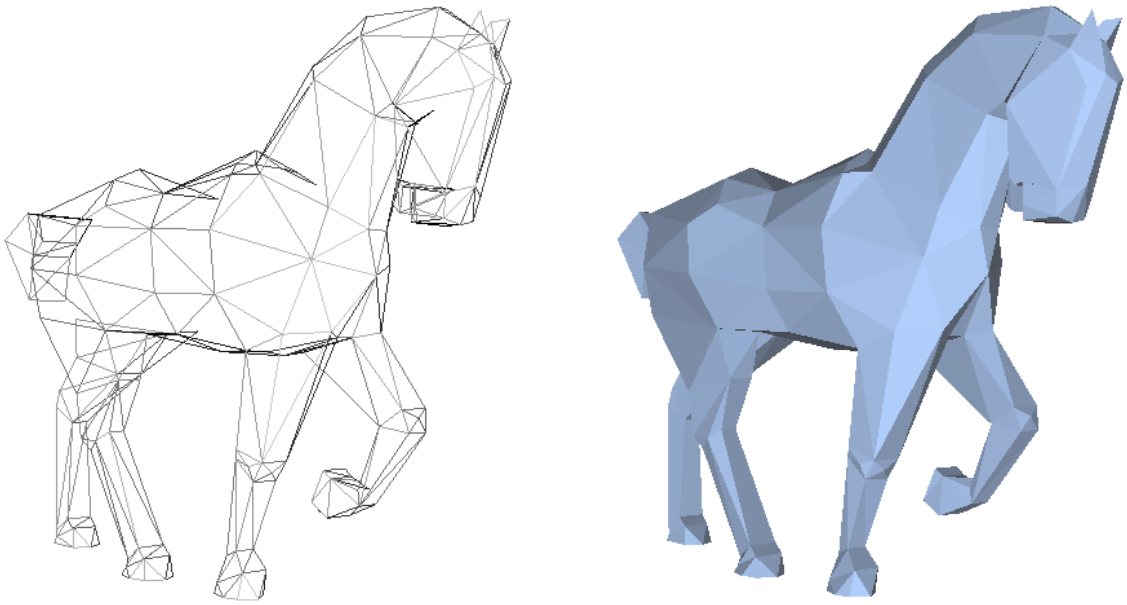
Petre R.-D., Zaharia T., Prêteux F., "An overview of view-based 2D/3D indexing methods", *Proceedings of SPIE Conference on Mathematics of Data/Image Pattern Recognition, Compression, and Encryption, with Applications XIII (SPIE Optics + Photonics 2010)*, Vol. 7799, pp. 779904, San Diego, CA, USA, August 2010.

# ANNEXE

## A1. 3D MESH MODELS

The 3D models employed in the current work are represented as triangular meshes and stored in Virtual Reality Modelling Language (VRML) format.

The triangular mesh represents a collection of triangular faces in the 3D space that defines the surface of the model (Figure A.1a).



**a.** **b.**

*Figure A.1 Mesh representation.*

The storage strategy is based on the face–vertex mesh representation method. First, the vertices positions in the 3D space (sample points with *(x, y, z)* coordinates) are stored in an unorganized way consisting on the vertex list. Then, the triangular faces are also defined by an unorganized face list.

Each entry of the faces list defines a triangle by the indices of its vertices (indexed by their order of appearance in the vertex list). In other words, any mesh file format will store mainly the geometry and the connectivity of the model. However, additional information can be included, such as colour, normal vertices, transparency or texture data.

In the VRML format, 3D objects are specified in a dedicated node, so-called “Shape”. This node type has several attributes, including its material appearance and its geometry. The “geometry” attributes can be valued with pre-defined shape primitive nodes or with an “IndexedFaceSet” node. This latter node has two main attributes which are the coordinates of the sample points (“coord” field, valued with a “Coordinate” node) and the face specification

("coordIndex" valued with an array of vertex integer indices) [Sch98]. Consequently, the VRML format could cover the most basic mesh needs:

- a list of vertices;
- a list of faces;
- a list of materials (texture and colour);
- a list of texture coordinates;
- a list of lights (material, description and position).

A standard layout for surface mesh storage with VRML v2.0 can be written in the following way:

```
#VRML V2.0 utf8
  DirectionalLight {
      ambientIntensity  1
      colour             1 1 1
      direction         0 0 -1
      intensity         0
      on                TRUE
  }
  DEF MATERIAL Material {
       diffuseColour 1 1 1
  }
      Shape {
          geometry IndexedFaceSet {
          coord Coordinate {
          point [
          # sample point coordinate (x, y, z) list
          0 0 0
          1 0 0
          ...
              ]
            }
          texCoord TextureCoordinate {
          point [
          0.1291 0.3485
          0.1706 0.3248
          ...
          ]
          }
            coordIndex [
          # face list: vertex indices (face separator: "-1")
          0 1 2 -1
          0 1 5 -1
          ...
          ]
          }
        }
```

In our work, only the geometry of the modes is exploited and any material or light information is ignored.

## A2. CATEGORIES OF 3D MODELS

This annexe contains the full list of categories for each one of the 3D model databases involved in our work.

*Table A.1 List of categories included in the MPEG7_23 database*

| | | | | | |
|---|---|---|---|---|---|
| 1 | airplanes | 9 | helicopters | 17 | spherical |
| 2 | humanoides | 10 | pistols | 18 | finger |
| 3 | cars | 11 | rifles | 19 | letter_a |
| 4 | tanks | 12 | chess | 20 | letter_b |
| 5 | trucks | 13 | screwdrivers | 21 | letter_c |
| 6 | formula_1 | 14 | missiles_cylinders_submarines | 22 | letter_d |
| 7 | motorcycles_3_wheels | 15 | trees_without_leeves | 23 | letter_e |
| 8 | motorcycles_2_wheels | 16 | trees | | |

*Table A.2 List of categories included in the PSB_53 database*

| | | | | | |
|---|---|---|---|---|---|
| 1 | winged_vehicle__aircraft | 19 | chest | 37 | microchip |
| 2 | balloon_vehicle__aircraft | 20 | city | 38 | musical_instrument |
| 3 | helicopter__aircraft | 21 | display_device | 39 | plant |
| 4 | arthropod__animal | 22 | door | 40 | satellite_dish |
| 5 | human__biped__animal | 23 | dragon__fantasy_animal | 41 | sea_vessel |
| 6 | trex__biped__animal | 24 | fireplace | 42 | shoe |
| 7 | flying_creature__animal | 25 | bed__furniture | 43 | sign |
| 8 | quadruped__animal | 26 | cabinet__furniture | 44 | sink |
| 9 | snake__animal | 27 | seat__furniture | 45 | slot_machine |
| 10 | underwater_creature__animal | 28 | shelves__furniture | 46 | snowman |
| 11 | blade | 29 | table__furniture | 47 | staircase |
| 12 | head__body_part | 30 | geographic_map | 48 | swingset |
| 13 | hand__body_part | 31 | gun | 49 | handheld |
| 14 | skeleton__body_part | 32 | hat | 50 | car__vehicle |
| 15 | torso__body_part | 33 | ladder | 51 | cycle__vehicle |
| 16 | bridge | 34 | lamp | 52 | train__vehicle |
| 17 | building | 35 | liquid_container | 53 | wheel |
| 18 | chess_piece | 36 | mailbox | | |

*Table A.3 List of categories included in the PSB_161 database*

| | | | |
|---|---|---|---|
| 1 | F117__airplane__aircraft | 41 | butcher_knife__blade |
| 2 | biplane__airplane__aircraft | 42 | sword__blade |
| 3 | commercial__airplane__aircraft | 43 | brain__body_part |
| 4 | fighter_jet__airplane__aircraft | 44 | face__body_part |
| 5 | glider__airplane__aircraft | 45 | hand__body_part |
| 6 | multi_fuselage__airplane__aircraft | 46 | head__body_part |
| 7 | stealth_bomber__airplane__aircraft | 47 | skeleton__body_part |
| 8 | hot_air_balloon__balloon_vehicle__aircra | 48 | torso__body_part |
| 9 | dirigible__balloon_vehicle__aircraft | 49 | skull__body_part |
| 10 | helicopter__aircraft | 50 | bridge |
| 11 | enterprise_like__spaceship__aircraft | 51 | book |
| 12 | space_shuttle__spaceship__aircraft | 52 | castle__building |
| 13 | x_wing__spaceship__aircraft | 53 | dome_church__building |
| 14 | satellite__spaceship__aircraft | 54 | lighthouse__building |
| 15 | flying_saucer__spaceship__aircraft | 55 | roman_building__building |
| 16 | tie_fighter__spaceship__aircraft | 56 | barn__building |
| 17 | ant__insect__arthropod__animal | 57 | church__building |
| 18 | bee__insect__arthropod__animal | 58 | gazebo__building |
| 19 | butterfly__insect__arthropod__animal | 59 | one_story_home__building |
| 20 | spider__arthropod__animal | 60 | skyscraper__building |
| 21 | human__biped__animal | 61 | one_peak_tent__tent__building |
| 22 | human_arms_out__human__biped__anim | 62 | multiple_peak_tent__tent__building |
| 23 | walking_human__biped__animal | 63 | two_story_home__building |
| 24 | trex__biped__animal | 64 | chess_set |
| 25 | flying_bird__bird__flying_creature__anim | 65 | chess_piece |
| 26 | duck__bird__flying_creature__animal | 66 | chest |
| 27 | standing_bird__bird__flying_creature__an | 67 | city |
| 28 | dog__quadruped__animal | 68 | desktop__computer |
| 29 | 160abelling160s__quadruped__animal | 69 | laptop__computer |
| 30 | feline__quadruped__animal | 70 | computer_monitor__display_device |
| 31 | pig__quadruped__animal | 71 | tv__display_device |
| 32 | horse__quadruped__animal | 72 | door |
| 33 | rabbit__quadruped__animal | 73 | double_doors__door |
| 34 | snake__animal | 74 | eyeglasses |
| 35 | dolphin__underwater_creature__animal | 75 | dragon__fantasy_animal |
| 36 | shark__underwater_creature__animal | 76 | fireplace |
| 37 | sea_turtle__underwater_creature__animal | 77 | bed__furniture |
| 38 | fish__underwater_creature__animal | 78 | cabinet__furniture |
| 39 | axe__blade | 79 | school_desk__furniture |
| 40 | knife__blade | 80 | desk_with_hutch__desk__furniture |

82 stool__chair__seat__furniture
83 dining_chair__chair__seat__furniture
84 couch__seat__furniture
85 desk_chair__seat__furniture
86 shelves__furniture
87 rectangular__table__furniture
88 round__table__furniture
89 single_leg__round__table__furniture
90 table_and_chairs__furniture
91 geographic_map
92 handgun__gun
93 rifle__gun
94 hat
95 helmet__hat
96 hourglass
97 ice_cream
98 ladder
99 desk_lamp__lamp
100 streetlight__lamp
101 bottle__liquid_container
102 mug__liquid_container
103 tank__liquid_container
104 glass_with_stem__liquid_container
105 pail__liquid_container
106 vase__liquid_container
107 mailbox
108 microchip
109 electrical_guitar__guitar__musical_instru
110 acoustic_guitar__guitar__musical_instrum
111 piano__musical_instrument
112 161abelling_toy
113 phone_handle
114 bush__plant
115 flowers__plant
116 flower_with_stem__plant
117 potted_plant__plant
118 tree__plant
119 barren__tree__plant
120 conical__tree__plant
121 palm__tree__plant
122 satellite_dish
123 sailboat__sea_vessel
124 large_sail_boat__sailboat__sea_vessel
125 sailboat_with_oars__sailboat__sea_vessel
126 ship__sea_vessel
127 submarine__sea_vessel
128 shoe
129 billboard__sign
130 street_sign__sign
131 sink
132 skateboard
133 slot_machine
134 snowman
135 staircase
136 swingset
137 hammer__tool
138 screwdriver__tool
139 shovel__tool
140 wrench__tool
141 umbrella
142 antique_car__car__vehicle
143 race_car__car__vehicle
144 sedan__car__vehicle
145 sports_car__car__vehicle
146 covered_wagon__vehicle
147 bicycle__cycle__vehicle
148 motorcycle__cycle__vehicle
149 military_tank__vehicle
150 monster_truck__vehicle
151 pickup_truck__vehicle
152 semi__vehicle
153 suv__vehicle
154 jeep__suv__vehicle
155 train__vehicle
156 train_car__train__vehicle
157 watch
158 wheel
159 tire__wheel
160 gear__wheel
161 microscope

## A3. CATEGORIES OF 2D OBJECTS

*Table A.4 List of SOI categories tested with MPEG7_23 database*

| | | | |
|---|---|---|---|
| 1 | airplanes | 13 | screwdrivers |
| 2 | humanoids | 14 | missiles_cylinders_submarines |
| 3 | cars | 15 | trees_without_leeves |
| 4 | tanks | 16 | trees |
| 5 | trucks | 17 | spherical |
| 6 | formula_1 | 18 | finger |
| 7 | motorcycles_3_wheels | 19 | letter_a |
| 8 | motorcycles_2_wheels | 20 | letter_b |
| 9 | helicopters | 21 | letter_c |
| 10 | pistols | 22 | letter_d |
| 11 | rifles | 23 | letter_e |
| 12 | chess | 12 | chess |

*Table A.5 List of SOI categories with PSB-53 and PSB_161_23 databases*

| | | | |
|---|---|---|---|
| 1 | airplanes | 8 | pistols |
| 2 | humanoids | 9 | rifles |
| 3 | cars | 10 | chess |
| 4 | tanks | 11 | screwdrivers |
| 5 | formula_1 | 12 | trees_without_leeves |
| 6 | motorcycles_2_wheels | 13 | trees |
| 7 | helicopters | 13 | trees |

*Table A.6 List of SOV and VOV categories*

| | | | |
|---|---|---|---|
| 1 | airplanes | 5 | humanoids |
| 2 | cars | 6 | motorcycles_2_wheels |
| 3 | chess | 7 | pistols |
| 4 | helicopters | 8 | tanks |

*Table A.7 List of SOSy and VOSy categories*

| | | | |
|---|---|---|---|
| 1 | airplanes | 8 | pistols |
| 2 | humanoids | 9 | rifles |
| 3 | cars | 10 | chess |
| 4 | tanks | 11 | screwdrivers |
| 5 | formula_1 | 12 | trees_without_leeves |
| 6 | motorcycles_2_wheels | 13 | trees |
| 7 | helicopters | | |

## A4. 2D OBJECT RECOGNITION RESULTS

*Table A.8 SOI database: Recognition rates obtained with the help of MPEG7_23 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 34,78 | 44,35 | 22,61 | 44,35 | 46,09 | 48,70 | 51,30 | 47,83 |
| | *RR(2)* | 50,43 | 56,52 | 35,65 | 59,13 | 60,87 | 60,00 | 59,13 | 60,00 |
| | *RR(3)* | 58,26 | 65,22 | 42,61 | 66,09 | 67,83 | 71,30 | 66,96 | 69,57 |
| CS | *RR(1)* | 40,00 | 40,87 | 42,61 | 39,13 | 42,61 | 44,35 | 44,35 | 42,61 |
| | *RR(2)* | 46,96 | 53,04 | 52,17 | 51,30 | 53,91 | 54,78 | 53,04 | 53,91 |
| | *RR(3)* | 55,65 | 60,87 | 60,00 | 55,65 | 63,48 | 61,74 | 60,87 | 65,22 |
| HT | *RR(1)* | 20,00 | 23,48 | 32,17 | 29,57 | 33,91 | 40,00 | 35,65 | 39,13 |
| | *RR(2)* | 29,57 | 31,30 | 42,61 | 33,91 | 38,26 | 42,61 | 41,74 | 45,22 |
| | *RR(3)* | 38,26 | 40,00 | 54,78 | 40,00 | 47,83 | 48,70 | 52,17 | 54,78 |
| RS | *RR(1)* | 20,87 | 24,35 | 30,43 | 24,35 | 30,43 | 35,65 | 36,52 | 34,78 |
| | *RR(2)* | 33,91 | 37,39 | 43,48 | 43,48 | 40,87 | 46,09 | 44,35 | 40,87 |
| | *RR(3)* | 40,87 | 47,83 | 49,57 | 47,83 | 51,30 | 51,30 | 54,78 | 51,30 |
| ZM | *RR(1)* | 34,78 | 36,52 | 40,00 | 31,30 | 42,61 | 40,00 | 40,87 | 40,87 |
| | *RR(2)* | 38,26 | 44,35 | 51,30 | 40,00 | 46,96 | 50,43 | 51,30 | 51,30 |
| | *RR(3)* | 43,48 | 52,17 | 58,26 | 50,43 | 53,04 | 59,13 | 58,26 | 56,52 |
| AH + CS | *RR(1)* | 39,13 | 44,35 | 41,74 | 45,22 | 47,83 | 47,83 | 50,43 | 51,30 |
| | *RR(2)* | 53,91 | 54,78 | 48,70 | 60,00 | 60,87 | 59,13 | 60,00 | 60,00 |
| | *RR(3)* | 60,87 | 65,22 | 61,74 | 68,70 | 69,57 | 68,70 | 67,83 | 70,43 |
| AH + ZM | *RR(1)* | 40,87 | 42,61 | 40,00 | 40,87 | 51,30 | 46,09 | 51,30 | 49,57 |
| | *RR(2)* | 51,30 | 54,78 | 48,70 | 53,91 | 60,00 | 61,74 | 56,52 | 58,26 |
| | *RR(3)* | 54,78 | 61,74 | 59,13 | 65,22 | 68,70 | 66,09 | 65,22 | 65,22 |
| CS + ZM | *RR(1)* | 42,61 | 41,74 | 44,35 | 36,52 | 48,70 | 49,57 | 48,70 | 46,96 |
| | *RR(2)* | 48,70 | 51,30 | 53,91 | 48,70 | 55,65 | 56,52 | 56,52 | 58,26 |
| | *RR(3)* | 55,65 | 58,26 | 62,61 | 58,26 | 60,00 | 65,22 | 60,87 | 65,22 |

*Table A.9 SOI database: Recognition rates obtained with the help of PSB_53 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 50,77 | 55,38 | 73,85 | 61,54 | 63,08 | 64,62 | 63,08 | 69,23 |
| | *RR(2)* | 64,62 | 69,23 | 81,54 | 72,31 | 72,31 | 72,31 | 80,00 | 78,46 |
| | *RR(3)* | 66,15 | 73,85 | 83,08 | 76,92 | 81,54 | 83,08 | 84,62 | 86,15 |
| CS | *RR(1)* | 47,69 | 52,31 | 58,46 | 55,38 | 58,46 | 61,54 | 55,38 | 60,00 |
| | *RR(2)* | 58,46 | 69,23 | 69,23 | 69,23 | 72,31 | 67,69 | 72,31 | 66,15 |
| | *RR(3)* | 63,08 | 75,38 | 73,85 | 75,38 | 75,38 | 72,31 | 76,92 | 70,77 |
| HT | *RR(1)* | 26,15 | 32,31 | 38,46 | 32,31 | 38,46 | 47,69 | 46,15 | 49,23 |
| | *RR(2)* | 35,38 | 38,46 | 55,38 | 35,38 | 47,69 | 60,00 | 58,46 | 64,62 |
| | *RR(3)* | 40,00 | 44,62 | 63,08 | 41,54 | 53,85 | 63,08 | 63,08 | 67,69 |
| RS | *RR(1)* | 32,31 | 36,92 | 35,38 | 33,85 | 44,62 | 40,00 | 38,46 | 41,54 |
| | *RR(2)* | 46,15 | 52,31 | 58,46 | 52,31 | 55,38 | 64,62 | 53,85 | 56,92 |
| | *RR(3)* | 53,85 | 58,46 | 64,62 | 60,00 | 67,69 | 69,23 | 58,46 | 67,69 |
| ZM | *RR(1)* | 44,62 | 44,62 | 43,08 | 44,62 | 47,69 | 46,15 | 47,69 | 50,77 |
| | *RR(2)* | 49,23 | 49,23 | 60,00 | 47,69 | 55,38 | 61,54 | 61,54 | 67,69 |
| | *RR(3)* | 55,38 | 58,46 | 66,15 | 60,00 | 63,08 | 64,62 | 66,15 | 73,85 |

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH + CS | *RR(1)* | 56,92 | 64,62 | 70,77 | 67,69 | 73,85 | 72,31 | 69,23 | 72,31 |
| | *RR(2)* | 63,08 | 76,92 | 81,54 | 80,00 | 83,08 | 81,54 | 80,00 | 80,00 |
| | *RR(3)* | 72,31 | 80,00 | 83,08 | 86,15 | 86,15 | 84,62 | 86,15 | 84,62 |
| AH + ZM | *RR(1)* | 56,92 | 56,92 | 66,15 | 67,69 | 61,54 | 69,23 | 67,69 | 72,31 |
| | *RR(2)* | 63,08 | 70,77 | 78,46 | 72,31 | 73,85 | 78,46 | 78,46 | 84,62 |
| | *RR(3)* | 67,69 | 76,92 | 84,62 | 73,85 | 83,08 | 81,54 | 86,15 | 86,15 |
| CS + ZM | *RR(1)* | 52,31 | 52,31 | 58,46 | 53,85 | 61,54 | 64,62 | 58,46 | 58,46 |
| | *RR(2)* | 61,54 | 66,15 | 67,69 | 60,00 | 72,31 | 66,15 | 66,15 | 72,31 |
| | *RR(3)* | 64,62 | 70,77 | 73,85 | 70,77 | 75,38 | 72,31 | 76,92 | 75,38 |

*Table A.10 SOI database: Recognition rates obtained with the help of PSB_161 models.*

| PSB_161 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 35,38 | 38,46 | 40,00 | 43,08 | 43,08 | 43,08 | 43,08 | 46,15 |
| | *RR(2)* | 50,77 | 50,77 | 56,92 | 58,46 | 55,38 | 55,38 | 64,62 | 55,38 |
| | *RR(3)* | 60,00 | 55,38 | 61,54 | 66,15 | 64,62 | 64,62 | 64,62 | 64,62 |
| | *RR(10)* | 67,69 | 69,23 | 78,46 | 80,00 | 80,00 | 80,00 | 83,08 | 90,77 |
| CS | *RR(1)* | 35,38 | 35,38 | 44,62 | 38,46 | 41,54 | 53,85 | 41,54 | 33,85 |
| | *RR(2)* | 44,62 | 50,77 | 58,46 | 47,69 | 55,38 | 61,54 | 52,31 | 46,15 |
| | *RR(3)* | 49,23 | 61,54 | 63,08 | 56,92 | 63,08 | 61,54 | 58,46 | 55,38 |
| | *RR(10)* | 64,62 | 75,38 | 72,31 | 72,31 | 73,85 | 69,23 | 73,85 | 76,92 |
| HT | *RR(1)* | 12,31 | 16,92 | 29,23 | 21,54 | 27,69 | 36,92 | 36,92 | 40,00 |
| | *RR(2)* | 15,38 | 23,08 | 32,31 | 24,62 | 36,92 | 44,62 | 44,62 | 49,23 |
| | *RR(3)* | 21,54 | 23,08 | 36,92 | 32,31 | 40,00 | 47,69 | 47,69 | 52,31 |
| | *RR(10)* | 30,77 | 36,92 | 49,23 | 38,46 | 50,77 | 58,46 | 53,85 | 58,46 |
| RS | *RR(1)* | 21,54 | 26,15 | 24,62 | 21,54 | 32,31 | 30,77 | 27,69 | 32,31 |
| | *RR(2)* | 32,31 | 27,69 | 41,54 | 33,85 | 40,00 | 44,62 | 46,15 | 44,62 |
| | *RR(3)* | 36,92 | 35,38 | 50,77 | 47,69 | 46,15 | 49,23 | 50,77 | 53,85 |
| | *RR(10)* | 47,69 | 52,31 | 56,92 | 58,46 | 64,62 | 61,54 | 60,00 | 63,08 |
| ZM | *RR(1)* | 27,69 | 30,77 | 33,85 | 33,85 | 36,92 | 41,54 | 38,46 | 41,54 |
| | *RR(2)* | 29,23 | 38,46 | 41,54 | 43,08 | 41,54 | 50,77 | 44,62 | 53,85 |
| | *RR(3)* | 36,92 | 41,54 | 46,15 | 49,23 | 47,69 | 50,77 | 47,69 | 55,38 |
| | *RR(10)* | 52,31 | 60,00 | 60,00 | 60,00 | 60,00 | 63,08 | 61,54 | 66,15 |
| AH + CS | *RR(1)* | 36,92 | 46,15 | 49,23 | 49,23 | 50,77 | 46,15 | 50,77 | 50,77 |
| | *RR(2)* | 53,85 | 60,00 | 66,15 | 61,54 | 66,15 | 63,08 | 66,15 | 63,08 |
| | *RR(3)* | 64,62 | 69,23 | 70,77 | 70,77 | 73,85 | 67,69 | 73,85 | 76,92 |
| | *RR(10)* | 75,38 | 78,46 | 84,62 | 81,54 | 84,62 | 83,08 | 84,62 | 90,77 |
| AH + ZM | *RR(1)* | 38,46 | 41,54 | 44,62 | 47,69 | 46,15 | 47,69 | 47,69 | 52,31 |
| | *RR(2)* | 50,77 | 50,77 | 60,00 | 58,46 | 61,54 | 64,62 | 56,92 | 66,15 |
| | *RR(3)* | 56,92 | 58,46 | 69,23 | 66,15 | 69,23 | 70,77 | 67,69 | 73,85 |
| | *RR(10)* | 72,31 | 75,38 | 81,54 | 81,54 | 81,54 | 81,54 | 83,08 | 92,31 |
| CS + ZM | *RR(1)* | 40,00 | 40,00 | 43,08 | 43,08 | 50,77 | 52,31 | 46,15 | 38,46 |
| | *RR(2)* | 46,15 | 50,77 | 56,92 | 49,23 | 56,92 | 60,00 | 56,92 | 53,85 |
| | *RR(3)* | 52,31 | 56,92 | 67,69 | 53,85 | 58,46 | 63,08 | 61,54 | 63,08 |
| | *RR(10)* | 64,62 | 73,85 | 72,31 | 73,85 | 73,85 | 75,38 | 73,85 | 78,46 |

*Table A.11 SOSy database: Recognition rates obtained with the help of MPEG7_23 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 37,18 | 48,21 | 53,59 | 53,33 | 51,54 | 56,15 | 54,62 | 52,82 |
| | *RR(2)* | 54,10 | 65,90 | 77,69 | 72,05 | 67,69 | 74,87 | 71,03 | 70,77 |
| | *RR(3)* | 64,10 | 75,38 | 84,36 | 80,77 | 79,49 | 83,85 | 80,77 | 81,54 |
| CS | *RR(1)* | 38,72 | 49,23 | 58,21 | 54,10 | 48,46 | 54,87 | 52,82 | 50,26 |
| | *RR(2)* | 48,97 | 63,33 | 68,97 | 69,49 | 60,77 | 71,79 | 65,38 | 67,95 |
| | *RR(3)* | 61,28 | 72,31 | 78,97 | 77,44 | 72,56 | 82,05 | 79,23 | 80,00 |
| HT | *RR(1)* | 23,33 | 28,46 | 36,67 | 27,95 | 32,56 | 41,03 | 34,87 | 36,15 |
| | *RR(2)* | 34,87 | 43,08 | 54,36 | 48,46 | 48,46 | 57,69 | 47,69 | 51,03 |
| | *RR(3)* | 45,90 | 57,44 | 62,82 | 58,21 | 58,97 | 68,72 | 55,90 | 61,79 |
| RS | *RR(1)* | 27,69 | 34,87 | 39,23 | 39,74 | 37,18 | 44,36 | 40,00 | 42,31 |
| | *RR(2)* | 44,87 | 49,74 | 57,95 | 56,15 | 55,90 | 62,31 | 57,69 | 58,21 |
| | *RR(3)* | 50,00 | 61,03 | 71,03 | 66,92 | 64,36 | 71,79 | 68,72 | 71,03 |
| ZM | *RR(1)* | 28,97 | 34,87 | 42,82 | 39,74 | 40,51 | 51,03 | 39,74 | 45,13 |
| | *RR(2)* | 42,82 | 53,33 | 61,03 | 56,92 | 58,46 | 67,95 | 55,90 | 59,49 |
| | *RR(3)* | 55,64 | 66,15 | 71,28 | 66,92 | 67,69 | 78,72 | 67,18 | 68,21 |
| AH + CS | *RR(1)* | 42,31 | 52,56 | 59,49 | 56,92 | 54,62 | 59,23 | 56,92 | 58,21 |
| | *RR(2)* | 50,77 | 64,62 | 75,90 | 70,00 | 67,95 | 74,36 | 70,26 | 72,31 |
| | *RR(3)* | 66,15 | 75,90 | 84,62 | 82,56 | 79,74 | 85,13 | 81,79 | 83,08 |
| AH + ZM | *RR(1)* | 37,69 | 48,21 | 55,13 | 53,08 | 52,56 | 58,97 | 52,31 | 55,38 |
| | *RR(2)* | 49,23 | 61,28 | 74,36 | 66,67 | 65,13 | 75,13 | 66,92 | 68,97 |
| | *RR(3)* | 63,33 | 72,31 | 85,13 | 76,92 | 76,15 | 83,85 | 80,51 | 80,51 |
| CS + ZM | *RR(1)* | 36,15 | 48,21 | 54,62 | 52,56 | 50,26 | 55,64 | 50,51 | 54,62 |
| | *RR(2)* | 48,46 | 60,26 | 68,21 | 64,36 | 62,82 | 71,03 | 63,85 | 66,92 |
| | *RR(3)* | 60,00 | 69,49 | 80,26 | 75,64 | 72,82 | 83,08 | 77,69 | 81,03 |

*Table A.12 SOSy database: Recognition rates obtained with the help of PSB_53 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 44,36 | 49,49 | 56,92 | 51,79 | 54,87 | 59,23 | 55,64 | 57,18 |
| | *RR(2)* | 60,51 | 66,67 | 72,82 | 71,03 | 68,72 | 75,64 | 71,54 | 73,33 |
| | *RR(3)* | 72,31 | 74,87 | 81,28 | 77,69 | 77,69 | 82,56 | 80,77 | 81,28 |
| CS | *RR(1)* | 41,03 | 54,62 | 60,26 | 55,38 | 58,97 | 62,82 | 58,21 | 59,23 |
| | *RR(2)* | 53,85 | 65,38 | 68,97 | 67,44 | 70,51 | 74,10 | 70,77 | 70,77 |
| | *RR(3)* | 63,85 | 70,77 | 76,67 | 74,62 | 76,92 | 79,74 | 77,95 | 76,67 |
| HT | *RR(1)* | 22,82 | 33,85 | 45,13 | 34,87 | 38,46 | 46,15 | 45,13 | 46,92 |
| | *RR(2)* | 30,51 | 43,08 | 54,87 | 44,36 | 49,23 | 57,44 | 53,59 | 56,67 |
| | *RR(3)* | 35,13 | 52,82 | 62,82 | 53,33 | 57,18 | 67,44 | 61,03 | 64,62 |
| RS | *RR(1)* | 33,85 | 41,79 | 51,03 | 41,79 | 44,62 | 52,56 | 47,18 | 49,74 |
| | *RR(2)* | 50,51 | 56,67 | 63,85 | 59,49 | 58,72 | 64,62 | 60,77 | 63,85 |
| | *RR(3)* | 60,26 | 65,90 | 71,03 | 70,77 | 67,95 | 73,59 | 68,97 | 72,56 |
| ZM | *RR(1)* | 35,64 | 44,36 | 54,87 | 47,18 | 47,95 | 54,62 | 53,33 | 56,41 |
| | *RR(2)* | 46,67 | 53,59 | 64,87 | 59,49 | 57,69 | 67,44 | 64,10 | 66,15 |
| | *RR(3)* | 55,64 | 63,08 | 71,03 | 67,18 | 66,15 | 77,95 | 70,26 | 73,85 |

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH + CS | *RR(1)* | 48,21 | 55,38 | 63,85 | 57,44 | 63,08 | 66,92 | 64,62 | 64,36 |
| | *RR(2)* | 62,56 | 69,74 | 74,10 | 71,28 | 74,62 | 78,72 | 74,62 | 74,10 |
| | *RR(3)* | 74,10 | 79,23 | 83,33 | 81,54 | 82,05 | 85,90 | 83,08 | 83,59 |
| AH + ZM | *RR(1)* | 47,18 | 53,33 | 62,31 | 56,41 | 57,18 | 63,59 | 61,79 | 62,05 |
| | *RR(2)* | 62,56 | 67,69 | 72,31 | 70,77 | 68,72 | 75,90 | 70,51 | 71,79 |
| | *RR(3)* | 72,56 | 76,92 | 82,56 | 78,72 | 77,44 | 82,82 | 80,51 | 80,51 |
| CS + ZM | *RR(1)* | 43,59 | 57,18 | 63,08 | 59,74 | 61,54 | 67,69 | 63,85 | 63,85 |
| | *RR(2)* | 55,90 | 66,15 | 71,03 | 69,49 | 68,72 | 74,87 | 72,05 | 73,33 |
| | *RR(3)* | 65,64 | 72,56 | 78,21 | 77,44 | 76,15 | 82,31 | 80,51 | 78,46 |

*Table A.13 SOSy database: Recognition rates obtained with the help of PSB_161 models.*

| PSB_161 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 35,38 | 42,56 | 51,79 | 47,95 | 46,41 | 55,13 | 49,23 | 51,28 |
| | *RR(2)* | 47,69 | 53,33 | 60,77 | 60,51 | 59,49 | 66,41 | 60,26 | 66,41 |
| | *RR(3)* | 55,64 | 63,59 | 70,26 | 67,44 | 66,67 | 75,64 | 68,72 | 72,82 |
| | *RR(10)* | 68,21 | 78,21 | 85,38 | 80,26 | 80,77 | 85,38 | 84,62 | 84,87 |
| CS | *RR(1)* | 30,26 | 45,13 | 50,77 | 46,92 | 48,97 | 56,15 | 47,44 | 53,59 |
| | *RR(2)* | 43,08 | 56,15 | 63,85 | 60,00 | 60,77 | 67,18 | 62,56 | 64,62 |
| | *RR(3)* | 48,97 | 62,82 | 70,00 | 66,15 | 67,95 | 74,62 | 70,51 | 71,28 |
| | *RR(10)* | 64,10 | 74,87 | 80,26 | 77,69 | 78,46 | 84,62 | 82,56 | 82,56 |
| HT | *RR(1)* | 18,46 | 27,95 | 36,92 | 27,95 | 33,33 | 41,28 | 38,97 | 41,03 |
| | *RR(2)* | 22,56 | 33,85 | 44,62 | 35,38 | 42,82 | 53,59 | 46,67 | 52,05 |
| | *RR(3)* | 27,18 | 39,74 | 50,51 | 40,26 | 47,18 | 57,44 | 51,28 | 54,36 |
| | *RR(10)* | 35,90 | 52,82 | 69,49 | 56,67 | 61,03 | 73,59 | 68,97 | 69,49 |
| RS | *RR(1)* | 27,44 | 33,59 | 46,67 | 38,72 | 36,92 | 47,69 | 42,56 | 43,08 |
| | *RR(2)* | 36,67 | 44,62 | 57,69 | 48,21 | 49,49 | 58,46 | 55,13 | 55,90 |
| | *RR(3)* | 40,26 | 52,31 | 60,77 | 55,13 | 55,64 | 65,90 | 60,51 | 62,05 |
| | *RR(10)* | 52,31 | 63,85 | 73,85 | 68,72 | 66,41 | 78,21 | 74,10 | 76,92 |
| ZM | *RR(1)* | 27,18 | 35,64 | 46,41 | 38,46 | 40,77 | 48,97 | 43,85 | 47,95 |
| | *RR(2)* | 35,38 | 45,38 | 58,72 | 48,97 | 46,92 | 61,03 | 55,13 | 58,46 |
| | *RR(3)* | 40,51 | 51,03 | 65,90 | 55,13 | 53,08 | 68,72 | 62,82 | 62,56 |
| | *RR(10)* | 57,18 | 66,92 | 77,95 | 70,26 | 67,95 | 83,59 | 75,90 | 80,26 |
| AH + CS | *RR(1)* | 39,23 | 50,51 | 58,21 | 53,85 | 57,18 | 65,90 | 58,97 | 60,26 |
| | *RR(2)* | 50,51 | 62,56 | 70,51 | 66,92 | 64,87 | 74,36 | 69,74 | 71,03 |
| | *RR(3)* | 57,69 | 69,49 | 77,69 | 74,36 | 71,79 | 80,77 | 76,41 | 77,95 |
| | *RR(10)* | 72,82 | 81,79 | 86,41 | 83,33 | 85,13 | 88,21 | 87,44 | 87,69 |
| AH + ZM | *RR(1)* | 37,95 | 46,15 | 58,46 | 51,03 | 52,82 | 61,28 | 56,92 | 58,21 |
| | *RR(2)* | 49,74 | 59,74 | 71,03 | 64,36 | 63,08 | 72,31 | 65,90 | 69,23 |
| | *RR(3)* | 56,15 | 66,15 | 75,90 | 69,49 | 69,23 | 80,51 | 73,33 | 77,44 |
| | *RR(10)* | 73,33 | 79,49 | 87,44 | 81,28 | 82,56 | 88,97 | 86,41 | 88,46 |
| CS + ZM | *RR(1)* | 34,36 | 45,90 | 56,67 | 48,97 | 52,56 | 60,51 | 56,92 | 56,67 |
| | *RR(2)* | 44,87 | 57,44 | 66,41 | 62,05 | 63,33 | 71,03 | 68,72 | 68,21 |
| | *RR(3)* | 51,54 | 63,59 | 72,56 | 68,46 | 68,97 | 76,67 | 74,10 | 74,10 |
| | *RR(10)* | 68,21 | 76,41 | 82,56 | 80,77 | 79,49 | 85,90 | 84,62 | 85,64 |

*Table A.14 SOV database: Recognition rates obtained with the help of MPEG7_23 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 28,33 | 40,00 | 46,67 | 44,17 | 39,17 | 51,67 | 45,00 | 46,67 |
| | *RR(2)* | 47,50 | 60,83 | 62,50 | 64,17 | 53,33 | 65,00 | 61,67 | 62,50 |
| | *RR(3)* | 64,17 | 69,17 | 73,33 | 70,83 | 70,00 | 73,33 | 73,33 | 73,33 |
| CS | *RR(1)* | 37,50 | 50,00 | 50,83 | 49,17 | 52,50 | 54,17 | 49,17 | 52,50 |
| | *RR(2)* | 48,33 | 60,00 | 66,67 | 62,50 | 63,33 | 69,17 | 63,33 | 65,00 |
| | *RR(3)* | 60,83 | 70,00 | 79,17 | 72,50 | 71,67 | 76,67 | 70,83 | 78,33 |
| HT | *RR(1)* | 26,67 | 32,50 | 36,67 | 34,17 | 39,17 | 49,17 | 42,50 | 50,00 |
| | *RR(2)* | 35,00 | 45,83 | 48,33 | 48,33 | 47,50 | 59,17 | 52,50 | 57,50 |
| | *RR(3)* | 44,17 | 56,67 | 57,50 | 57,50 | 55,00 | 67,50 | 60,83 | 70,00 |
| RS | *RR(1)* | 27,50 | 42,50 | 48,33 | 39,17 | 40,00 | 47,50 | 49,17 | 50,83 |
| | *RR(2)* | 45,83 | 57,50 | 62,50 | 52,50 | 57,50 | 68,33 | 60,00 | 63,33 |
| | *RR(3)* | 57,50 | 63,33 | 69,17 | 62,50 | 66,67 | 77,50 | 70,00 | 70,83 |
| ZM | *RR(1)* | 31,67 | 40,83 | 44,17 | 40,00 | 45,83 | 52,50 | 42,50 | 50,00 |
| | *RR(2)* | 40,83 | 50,00 | 64,17 | 57,50 | 58,33 | 70,00 | 55,00 | 61,67 |
| | *RR(3)* | 50,00 | 69,17 | 70,83 | 70,00 | 64,17 | 75,00 | 62,50 | 70,00 |
| AH + CS | *RR(1)* | 43,33 | 45,83 | 53,33 | 45,83 | 50,83 | 54,17 | 53,33 | 50,00 |
| | *RR(2)* | 56,67 | 62,50 | 65,00 | 67,50 | 60,83 | 67,50 | 67,50 | 63,33 |
| | *RR(3)* | 67,50 | 73,33 | 79,17 | 79,17 | 70,00 | 78,33 | 75,00 | 75,00 |
| AH + ZM | *RR(1)* | 34,17 | 49,17 | 54,17 | 45,00 | 51,67 | 55,83 | 50,00 | 50,83 |
| | *RR(2)* | 46,67 | 57,50 | 65,83 | 63,33 | 63,33 | 70,00 | 68,33 | 64,17 |
| | *RR(3)* | 56,67 | 73,33 | 78,33 | 74,17 | 69,17 | 79,17 | 73,33 | 75,00 |
| CS + ZM | *RR(1)* | 40,00 | 50,83 | 50,83 | 50,00 | 52,50 | 52,50 | 49,17 | 49,17 |
| | *RR(2)* | 49,17 | 61,67 | 66,67 | 62,50 | 60,83 | 69,17 | 62,50 | 66,67 |
| | *RR(3)* | 58,33 | 70,83 | 76,67 | 76,67 | 73,33 | 76,67 | 70,83 | 75,00 |

*Table A.15 SOV database: Recognition rates obtained with the help of PSB_53 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 32,50 | 42,50 | 50,83 | 39,17 | 35,83 | 50,83 | 48,33 | 50,83 |
| | *RR(2)* | 46,67 | 57,50 | 62,50 | 58,33 | 55,00 | 65,83 | 65,83 | 65,83 |
| | *RR(3)* | 58,33 | 67,50 | 70,83 | 69,17 | 64,17 | 75,83 | 73,33 | 77,50 |
| CS | *RR(1)* | 44,17 | 60,83 | 64,17 | 55,83 | 61,67 | 62,50 | 60,83 | 63,33 |
| | *RR(2)* | 58,33 | 73,33 | 75,00 | 69,17 | 72,50 | 76,67 | 74,17 | 74,17 |
| | *RR(3)* | 68,33 | 79,17 | 82,50 | 75,83 | 77,50 | 83,33 | 80,83 | 80,83 |
| HT | *RR(1)* | 28,33 | 34,17 | 51,67 | 30,00 | 40,00 | 49,17 | 50,83 | 55,00 |
| | *RR(2)* | 32,50 | 40,83 | 64,17 | 40,00 | 51,67 | 59,17 | 60,00 | 64,17 |
| | *RR(3)* | 37,50 | 45,00 | 69,17 | 44,17 | 57,50 | 65,83 | 66,67 | 68,33 |
| RS | *RR(1)* | 31,67 | 41,67 | 50,00 | 40,00 | 44,17 | 51,67 | 52,50 | 52,50 |
| | *RR(2)* | 42,50 | 54,17 | 65,83 | 50,83 | 57,50 | 64,17 | 60,00 | 65,83 |
| | *RR(3)* | 45,83 | 60,00 | 70,00 | 56,67 | 62,50 | 66,67 | 65,00 | 69,17 |
| ZM | *RR(1)* | 31,67 | 38,33 | 55,83 | 43,33 | 47,50 | 51,67 | 59,17 | 50,83 |
| | *RR(2)* | 36,67 | 52,50 | 65,83 | 51,67 | 56,67 | 68,33 | 71,67 | 66,67 |
| | *RR(3)* | 45,83 | 60,83 | 71,67 | 60,83 | 60,83 | 72,50 | 73,33 | 70,00 |

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH + CS | *RR(1)* | 47,50 | 55,00 | 63,33 | 52,50 | 56,67 | 65,00 | 60,00 | 63,33 |
| | *RR(2)* | 63,33 | 68,33 | 76,67 | 72,50 | 70,83 | 79,17 | 75,00 | 76,67 |
| | *RR(3)* | 69,17 | 82,50 | 85,00 | 80,83 | 78,33 | 87,50 | 84,17 | 87,50 |
| AH + ZM | *RR(1)* | 40,00 | 41,67 | 58,33 | 40,83 | 46,67 | 54,17 | 65,00 | 58,33 |
| | *RR(2)* | 50,00 | 60,00 | 70,00 | 63,33 | 61,67 | 74,17 | 76,67 | 73,33 |
| | *RR(3)* | 60,00 | 69,17 | 80,00 | 70,83 | 70,00 | 79,17 | 81,67 | 79,17 |
| CS + ZM | *RR(1)* | 48,33 | 56,67 | 65,00 | 54,17 | 63,33 | 67,50 | 67,50 | 66,67 |
| | *RR(2)* | 59,17 | 70,83 | 76,67 | 66,67 | 70,83 | 80,00 | 79,17 | 78,33 |
| | *RR(3)* | 65,00 | 78,33 | 84,17 | 80,00 | 75,83 | 86,67 | 87,50 | 85,00 |

*Table A.16 SOV database: Recognition rates obtained with the help of PSB_161 models.*

| PSB_161 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 27,50 | 35,00 | 38,33 | 35,00 | 30,00 | 47,50 | 45,00 | 44,17 |
| | *RR(2)* | 40,00 | 45,83 | 50,00 | 40,83 | 45,83 | 51,67 | 55,00 | 55,00 |
| | *RR(3)* | 45,83 | 52,50 | 53,33 | 50,00 | 53,33 | 58,33 | 59,17 | 65,83 |
| | *RR(10)* | 62,50 | 66,67 | 72,50 | 63,33 | 72,50 | 72,50 | 74,17 | 78,33 |
| CS | *RR(1)* | 34,17 | 51,67 | 53,33 | 48,33 | 51,67 | 52,50 | 48,33 | 45,83 |
| | *RR(2)* | 45,00 | 59,17 | 62,50 | 57,50 | 62,50 | 61,67 | 57,50 | 60,00 |
| | *RR(3)* | 52,50 | 63,33 | 69,17 | 59,17 | 65,00 | 65,00 | 61,67 | 65,83 |
| | *RR(10)* | 68,33 | 75,00 | 79,17 | 70,83 | 76,67 | 81,67 | 76,67 | 76,67 |
| HT | *RR(1)* | 21,67 | 22,50 | 38,33 | 22,50 | 35,83 | 38,33 | 40,83 | 46,67 |
| | *RR(2)* | 22,50 | 27,50 | 46,67 | 27,50 | 42,50 | 47,50 | 50,83 | 55,00 |
| | *RR(3)* | 27,50 | 32,50 | 54,17 | 33,33 | 45,83 | 55,00 | 56,67 | 57,50 |
| | *RR(10)* | 35,83 | 46,67 | 60,83 | 49,17 | 54,17 | 69,17 | 66,67 | 66,67 |
| RS | *RR(1)* | 22,50 | 31,67 | 34,17 | 32,50 | 34,17 | 38,33 | 36,67 | 40,00 |
| | *RR(2)* | 30,00 | 38,33 | 50,00 | 40,83 | 44,17 | 47,50 | 47,50 | 50,83 |
| | *RR(3)* | 35,83 | 45,83 | 52,50 | 42,50 | 50,83 | 54,17 | 52,50 | 53,33 |
| | *RR(10)* | 50,83 | 58,33 | 66,67 | 55,83 | 64,17 | 70,00 | 69,17 | 68,33 |
| ZM | *RR(1)* | 24,17 | 30,83 | 46,67 | 29,17 | 41,67 | 42,50 | 44,17 | 42,50 |
| | *RR(2)* | 29,17 | 37,50 | 53,33 | 36,67 | 47,50 | 54,17 | 55,00 | 54,17 |
| | *RR(3)* | 33,33 | 43,33 | 55,83 | 41,67 | 50,00 | 58,33 | 58,33 | 59,17 |
| | *RR(10)* | 50,83 | 57,50 | 73,33 | 58,33 | 60,00 | 72,50 | 66,67 | 68,33 |
| AH + CS | *RR(1)* | 36,67 | 47,50 | 55,83 | 47,50 | 52,50 | 58,33 | 55,00 | 53,33 |
| | *RR(2)* | 47,50 | 52,50 | 64,17 | 54,17 | 62,50 | 66,67 | 63,33 | 68,33 |
| | *RR(3)* | 55,83 | 65,00 | 67,50 | 61,67 | 70,00 | 71,67 | 70,83 | 74,17 |
| | *RR(10)* | 73,33 | 80,83 | 85,00 | 75,83 | 85,00 | 85,00 | 83,33 | 87,50 |
| AH + ZM | *RR(1)* | 33,33 | 38,33 | 50,83 | 39,17 | 40,83 | 51,67 | 50,00 | 47,50 |
| | *RR(2)* | 42,50 | 44,17 | 57,50 | 47,50 | 50,83 | 60,83 | 56,67 | 57,50 |
| | *RR(3)* | 48,33 | 55,00 | 63,33 | 53,33 | 58,33 | 64,17 | 64,17 | 65,83 |
| | *RR(10)* | 66,67 | 71,67 | 80,00 | 66,67 | 75,00 | 80,83 | 80,00 | 85,00 |
| CS + ZM | *RR(1)* | 40,83 | 46,67 | 59,17 | 44,17 | 55,00 | 55,83 | 54,17 | 53,33 |
| | *RR(2)* | 48,33 | 56,67 | 65,00 | 55,00 | 63,33 | 67,50 | 64,17 | 62,50 |
| | *RR(3)* | 55,83 | 64,17 | 71,67 | 60,83 | 69,17 | 72,50 | 70,83 | 71,67 |
| | *RR(10)* | 71,67 | 76,67 | 81,67 | 70,00 | 80,00 | 84,17 | 81,67 | 81,67 |

*Table A.17 VOV database: Recognition rates obtained with the help of MPEG7_23 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 35,00 | 50,00 | 50,00 | 57,50 | 50,00 | 57,50 | 52,50 | 50,00 |
| | *RR(2)* | 65,00 | 75,00 | 75,00 | 72,50 | 65,00 | 77,50 | 77,50 | 70,00 |
| | *RR(3)* | 75,00 | 87,50 | 82,50 | 82,50 | 72,50 | 85,00 | 85,00 | 80,00 |
| CS | *RR(1)* | 50,00 | 62,50 | 75,00 | 65,00 | 72,50 | 70,00 | 62,50 | 65,00 |
| | *RR(2)* | 62,50 | 77,50 | 82,50 | 72,50 | 77,50 | 80,00 | 75,00 | 77,50 |
| | *RR(3)* | 75,00 | 82,50 | 85,00 | 77,50 | 80,00 | 82,50 | 82,50 | 82,50 |
| HT | *RR(1)* | 35,00 | 42,50 | 42,50 | 37,50 | 47,50 | 52,50 | 52,50 | 65,00 |
| | *RR(2)* | 40,00 | 57,50 | 60,00 | 47,50 | 52,50 | 67,50 | 62,50 | 72,50 |
| | *RR(3)* | 42,50 | 67,50 | 75,00 | 60,00 | 60,00 | 75,00 | 67,50 | 80,00 |
| RS | *RR(1)* | 35,00 | 57,50 | 62,50 | 50,00 | 50,00 | 67,50 | 57,50 | 65,00 |
| | *RR(2)* | 60,00 | 77,50 | 75,00 | 70,00 | 80,00 | 90,00 | 75,00 | 82,50 |
| | *RR(3)* | 75,00 | 82,50 | 80,00 | 80,00 | 87,50 | 92,50 | 85,00 | 87,50 |
| ZM | *RR(1)* | 40,00 | 50,00 | 62,50 | 50,00 | 60,00 | 62,50 | 55,00 | 55,00 |
| | *RR(2)* | 52,50 | 62,50 | 80,00 | 72,50 | 75,00 | 80,00 | 70,00 | 77,50 |
| | *RR(3)* | 52,50 | 80,00 | 85,00 | 82,50 | 85,00 | 87,50 | 82,50 | 87,50 |
| AH + CS | *RR(1)* | 47,50 | 62,50 | 62,50 | 67,50 | 62,50 | 67,50 | 60,00 | 62,50 |
| | *RR(2)* | 67,50 | 77,50 | 77,50 | 80,00 | 75,00 | 85,00 | 80,00 | 77,50 |
| | *RR(3)* | 82,50 | 87,50 | 87,50 | 85,00 | 82,50 | 87,50 | 85,00 | 85,00 |
| AH + ZM | *RR(1)* | 42,50 | 57,50 | 62,50 | 62,50 | 55,00 | 67,50 | 62,50 | 60,00 |
| | *RR(2)* | 52,50 | 70,00 | 77,50 | 75,00 | 65,00 | 77,50 | 77,50 | 75,00 |
| | *RR(3)* | 62,50 | 87,50 | 82,50 | 87,50 | 80,00 | 82,50 | 87,50 | 82,50 |
| CS + ZM | *RR(1)* | 45,00 | 70,00 | 67,50 | 60,00 | 72,50 | 67,50 | 57,50 | 62,50 |
| | *RR(2)* | 57,50 | 75,00 | 77,50 | 70,00 | 77,50 | 82,50 | 75,00 | 77,50 |
| | *RR(3)* | 62,50 | 82,50 | 87,50 | 85,00 | 82,50 | 92,50 | 85,00 | 87,50 |

*Table A.18 VOV database: Recognition rates obtained with the help of PSB_53 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 37,50 | 55,00 | 62,50 | 47,50 | 47,50 | 62,50 | 57,50 | 60,00 |
| | *RR(2)* | 42,50 | 72,50 | 77,50 | 57,50 | 67,50 | 77,50 | 72,50 | 65,00 |
| | *RR(3)* | 62,50 | 82,50 | 87,50 | 70,00 | 72,50 | 87,50 | 85,00 | 85,00 |
| CS | *RR(1)* | 60,00 | 75,00 | 85,00 | 72,50 | 85,00 | 85,00 | 82,50 | 77,50 |
| | *RR(2)* | 72,50 | 85,00 | 90,00 | 85,00 | 87,50 | 90,00 | 87,50 | 87,50 |
| | *RR(3)* | 77,50 | 85,00 | 92,50 | 92,50 | 92,50 | 92,50 | 90,00 | 92,50 |
| HT | *RR(1)* | 32,50 | 32,50 | 60,00 | 27,50 | 47,50 | 60,00 | 52,50 | 75,00 |
| | *RR(2)* | 37,50 | 50,00 | 72,50 | 42,50 | 50,00 | 72,50 | 67,50 | 80,00 |
| | *RR(3)* | 40,00 | 50,00 | 77,50 | 47,50 | 57,50 | 77,50 | 77,50 | 82,50 |
| RS | *RR(1)* | 30,00 | 62,50 | 62,50 | 50,00 | 52,50 | 70,00 | 62,50 | 77,50 |
| | *RR(2)* | 45,00 | 72,50 | 75,00 | 60,00 | 67,50 | 80,00 | 75,00 | 85,00 |
| | *RR(3)* | 55,00 | 72,50 | 77,50 | 65,00 | 72,50 | 80,00 | 77,50 | 85,00 |
| ZM | *RR(1)* | 45,00 | 50,00 | 72,50 | 47,50 | 55,00 | 65,00 | 62,50 | 67,50 |
| | *RR(2)* | 52,50 | 60,00 | 85,00 | 55,00 | 60,00 | 80,00 | 75,00 | 77,50 |
| | *RR(3)* | 57,50 | 67,50 | 87,50 | 65,00 | 70,00 | 82,50 | 80,00 | 82,50 |

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH + CS | *RR(1)* | 57,50 | 60,00 | 87,50 | 60,00 | 75,00 | 85,00 | 75,00 | 80,00 |
| | *RR(2)* | 70,00 | 77,50 | 97,50 | 87,50 | 90,00 | 92,50 | 90,00 | 92,50 |
| | *RR(3)* | 82,50 | 90,00 | 97,50 | 95,00 | 97,50 | 95,00 | 95,00 | 97,50 |
| AH + ZM | *RR(1)* | 45,00 | 57,50 | 80,00 | 47,50 | 60,00 | 70,00 | 75,00 | 77,50 |
| | *RR(2)* | 55,00 | 80,00 | 90,00 | 62,50 | 75,00 | 82,50 | 80,00 | 82,50 |
| | *RR(3)* | 60,00 | 87,50 | 97,50 | 75,00 | 85,00 | 87,50 | 90,00 | 92,50 |
| CS + ZM | *RR(1)* | 60,00 | 72,50 | 87,50 | 62,50 | 85,00 | 82,50 | 85,00 | 82,50 |
| | *RR(2)* | 72,50 | 85,00 | 90,00 | 82,50 | 87,50 | 92,50 | 92,50 | 90,00 |
| | *RR(3)* | 77,50 | 90,00 | 92,50 | 95,00 | 92,50 | 95,00 | 92,50 | 92,50 |

*Table A.19 VOV database: Recognition rates obtained with the help of PSB_161 models.*

| PSB_161 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 50,00 | 47,50 | 50,00 | 35,00 | 47,50 | 70,00 | 65,00 | 60,00 |
| | *RR(2)* | 55,00 | 62,50 | 65,00 | 40,00 | 52,50 | 72,50 | 67,50 | 65,00 |
| | *RR(3)* | 62,50 | 70,00 | 67,50 | 47,50 | 60,00 | 75,00 | 72,50 | 77,50 |
| | *RR(10)* | 77,50 | 80,00 | 85,00 | 77,50 | 92,50 | 90,00 | 85,00 | 90,00 |
| CS | *RR(1)* | 57,50 | 72,50 | 70,00 | 55,00 | 75,00 | 77,50 | 67,50 | 62,50 |
| | *RR(2)* | 65,00 | 75,00 | 77,50 | 70,00 | 80,00 | 85,00 | 75,00 | 80,00 |
| | *RR(3)* | 70,00 | 77,50 | 85,00 | 77,50 | 85,00 | 87,50 | 80,00 | 80,00 |
| | *RR(10)* | 77,50 | 87,50 | 92,50 | 87,50 | 90,00 | 90,00 | 90,00 | 90,00 |
| HT | *RR(1)* | 30,00 | 30,00 | 45,00 | 25,00 | 45,00 | 47,50 | 55,00 | 60,00 |
| | *RR(2)* | 35,00 | 40,00 | 57,50 | 32,50 | 50,00 | 62,50 | 62,50 | 65,00 |
| | *RR(3)* | 35,00 | 40,00 | 60,00 | 35,00 | 50,00 | 70,00 | 65,00 | 65,00 |
| | *RR(10)* | 45,00 | 50,00 | 72,50 | 50,00 | 62,50 | 80,00 | 77,50 | 82,50 |
| RS | *RR(1)* | 27,50 | 47,50 | 45,00 | 42,50 | 52,50 | 60,00 | 52,50 | 67,50 |
| | *RR(2)* | 37,50 | 52,50 | 65,00 | 52,50 | 57,50 | 67,50 | 65,00 | 75,00 |
| | *RR(3)* | 42,50 | 62,50 | 67,50 | 52,50 | 62,50 | 72,50 | 67,50 | 80,00 |
| | *RR(10)* | 57,50 | 70,00 | 82,50 | 67,50 | 80,00 | 85,00 | 80,00 | 82,50 |
| ZM | *RR(1)* | 32,50 | 45,00 | 60,00 | 40,00 | 45,00 | 57,50 | 62,50 | 65,00 |
| | *RR(2)* | 42,50 | 52,50 | 70,00 | 47,50 | 55,00 | 75,00 | 75,00 | 72,50 |
| | *RR(3)* | 42,50 | 55,00 | 72,50 | 55,00 | 62,50 | 77,50 | 80,00 | 77,50 |
| | *RR(10)* | 55,00 | 77,50 | 90,00 | 70,00 | 77,50 | 90,00 | 87,50 | 87,50 |
| AH + CS | *RR(1)* | 60,00 | 62,50 | 77,50 | 40,00 | 77,50 | 82,50 | 77,50 | 80,00 |
| | *RR(2)* | 62,50 | 75,00 | 82,50 | 62,50 | 80,00 | 87,50 | 87,50 | 82,50 |
| | *RR(3)* | 77,50 | 85,00 | 85,00 | 77,50 | 90,00 | 90,00 | 87,50 | 87,50 |
| | *RR(10)* | 90,00 | 95,00 | 92,50 | 92,50 | 95,00 | 95,00 | 95,00 | 97,50 |
| AH + ZM | *RR(1)* | 40,00 | 47,50 | 65,00 | 40,00 | 52,50 | 70,00 | 65,00 | 72,50 |
| | *RR(2)* | 55,00 | 57,50 | 75,00 | 45,00 | 55,00 | 77,50 | 72,50 | 75,00 |
| | *RR(3)* | 57,50 | 80,00 | 85,00 | 62,50 | 67,50 | 85,00 | 82,50 | 82,50 |
| | *RR(10)* | 80,00 | 92,50 | 95,00 | 87,50 | 92,50 | 95,00 | 92,50 | 95,00 |
| CS + ZM | *RR(1)* | 55,00 | 67,50 | 75,00 | 50,00 | 72,50 | 80,00 | 77,50 | 75,00 |
| | *RR(2)* | 62,50 | 82,50 | 80,00 | 70,00 | 82,50 | 90,00 | 87,50 | 85,00 |
| | *RR(3)* | 70,00 | 85,00 | 82,50 | 77,50 | 82,50 | 92,50 | 90,00 | 90,00 |
| | *RR(10)* | 85,00 | 92,50 | 95,00 | 90,00 | 92,50 | 95,00 | 92,50 | 95,00 |

*Table A.20 VOSy database: Recognition rates obtained with the help of MPEG7_23 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 41,03 | 58,97 | 69,23 | 71,79 | 61,54 | 71,79 | 76,92 | 76,92 |
| | *RR(2)* | 74,36 | 87,18 | 87,18 | 87,18 | 89,74 | 87,18 | 89,74 | 92,31 |
| | *RR(3)* | 79,49 | 92,31 | 97,44 | 94,87 | 89,74 | 89,74 | 92,31 | 97,44 |
| CS | *RR(1)* | 61,54 | 71,79 | 76,92 | 66,67 | 71,79 | 76,92 | 74,36 | 69,23 |
| | *RR(2)* | 66,67 | 82,05 | 94,87 | 89,74 | 84,62 | 92,31 | 89,74 | 87,18 |
| | *RR(3)* | 79,49 | 92,31 | 94,87 | 94,87 | 89,74 | 97,44 | 94,87 | 92,31 |
| HT | *RR(1)* | 23,08 | 33,33 | 48,72 | 30,77 | 38,46 | 51,28 | 38,46 | 43,59 |
| | *RR(2)* | 38,46 | 48,72 | 66,67 | 51,28 | 71,79 | 71,79 | 56,41 | 61,54 |
| | *RR(3)* | 51,28 | 71,79 | 79,49 | 64,10 | 76,92 | 84,62 | 64,10 | 71,79 |
| RS | *RR(1)* | 41,03 | 51,28 | 64,10 | 61,54 | 53,85 | 69,23 | 61,54 | 64,10 |
| | *RR(2)* | 61,54 | 66,67 | 79,49 | 74,36 | 64,10 | 76,92 | 76,92 | 76,92 |
| | *RR(3)* | 69,23 | 87,18 | 84,62 | 84,62 | 82,05 | 87,18 | 84,62 | 87,18 |
| ZM | *RR(1)* | 48,72 | 64,10 | 66,67 | 66,67 | 74,36 | 74,36 | 61,54 | 69,23 |
| | *RR(2)* | 61,54 | 74,36 | 79,49 | 76,92 | 79,49 | 79,49 | 74,36 | 82,05 |
| | *RR(3)* | 71,79 | 87,18 | 89,74 | 84,62 | 84,62 | 84,62 | 82,05 | 89,74 |
| AH + CS | *RR(1)* | 53,85 | 61,54 | 74,36 | 74,36 | 71,79 | 82,05 | 74,36 | 76,92 |
| | *RR(2)* | 74,36 | 84,62 | 94,87 | 84,62 | 87,18 | 92,31 | 92,31 | 89,74 |
| | *RR(3)* | 87,18 | 92,31 | 94,87 | 94,87 | 92,31 | 97,44 | 94,87 | 97,44 |
| AH + ZM | *RR(1)* | 51,28 | 66,67 | 71,79 | 82,05 | 71,79 | 76,92 | 74,36 | 71,79 |
| | *RR(2)* | 69,23 | 82,05 | 92,31 | 92,31 | 84,62 | 87,18 | 82,05 | 87,18 |
| | *RR(3)* | 82,05 | 92,31 | 92,31 | 94,87 | 92,31 | 92,31 | 89,74 | 94,87 |
| CS + ZM | *RR(1)* | 66,67 | 69,23 | 74,36 | 69,23 | 74,36 | 76,92 | 69,23 | 69,23 |
| | *RR(2)* | 69,23 | 79,49 | 84,62 | 76,92 | 82,05 | 87,18 | 87,18 | 84,62 |
| | *RR(3)* | 76,92 | 89,74 | 89,74 | 92,31 | 87,18 | 92,31 | 92,31 | 94,87 |

*Table A.21 VOSy database: Recognition rates obtained with the help of PSB_53 models.*

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 56,41 | 64,10 | 71,79 | 69,23 | 74,36 | 79,49 | 66,67 | 69,23 |
| | *RR(2)* | 87,18 | 84,62 | 87,18 | 87,18 | 89,74 | 92,31 | 79,49 | 82,05 |
| | *RR(3)* | 92,31 | 89,74 | 94,87 | 92,31 | 89,74 | 94,87 | 89,74 | 92,31 |
| CS | *RR(1)* | 66,67 | 76,92 | 79,49 | 74,36 | 71,79 | 74,36 | 74,36 | 74,36 |
| | *RR(2)* | 79,49 | 87,18 | 87,18 | 89,74 | 87,18 | 84,62 | 87,18 | 82,05 |
| | *RR(3)* | 82,05 | 89,74 | 92,31 | 89,74 | 89,74 | 97,44 | 92,31 | 92,31 |
| HT | *RR(1)* | 38,46 | 46,15 | 56,41 | 41,03 | 58,97 | 64,10 | 56,41 | 56,41 |
| | *RR(2)* | 46,15 | 58,97 | 66,67 | 56,41 | 64,10 | 74,36 | 66,67 | 64,10 |
| | *RR(3)* | 53,85 | 69,23 | 66,67 | 66,67 | 74,36 | 76,92 | 74,36 | 71,79 |
| RS | *RR(1)* | 46,15 | 58,97 | 71,79 | 71,79 | 69,23 | 74,36 | 69,23 | 74,36 |
| | *RR(2)* | 64,10 | 69,23 | 74,36 | 76,92 | 82,05 | 82,05 | 74,36 | 84,62 |
| | *RR(3)* | 74,36 | 76,92 | 79,49 | 84,62 | 84,62 | 84,62 | 82,05 | 92,31 |
| ZM | *RR(1)* | 61,54 | 71,79 | 76,92 | 74,36 | 76,92 | 84,62 | 66,67 | 74,36 |
| | *RR(2)* | 66,67 | 74,36 | 84,62 | 79,49 | 79,49 | 87,18 | 74,36 | 79,49 |
| | *RR(3)* | 74,36 | 82,05 | 89,74 | 84,62 | 82,05 | 89,74 | 82,05 | 84,62 |

| MPEG7_23 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH + CS | *RR(1)* | 74,36 | 74,36 | 79,49 | 74,36 | 79,49 | 82,05 | 71,79 | 74,36 |
| | *RR(2)* | 84,62 | 87,18 | 89,74 | 87,18 | 87,18 | 89,74 | 89,74 | 84,62 |
| | *RR(3)* | 92,31 | 94,87 | 89,74 | 92,31 | 92,31 | 92,31 | 92,31 | 89,74 |
| AH + ZM | *RR(1)* | 74,36 | 74,36 | 82,05 | 76,92 | 79,49 | 82,05 | 76,92 | 74,36 |
| | *RR(2)* | 82,05 | 84,62 | 89,74 | 84,62 | 87,18 | 89,74 | 87,18 | 84,62 |
| | *RR(3)* | 89,74 | 94,87 | 92,31 | 92,31 | 89,74 | 92,31 | 89,74 | 89,74 |
| CS + ZM | *RR(1)* | 71,79 | 76,92 | 82,05 | 76,92 | 76,92 | 79,49 | 76,92 | 79,49 |
| | *RR(2)* | 74,36 | 84,62 | 84,62 | 82,05 | 84,62 | 87,18 | 84,62 | 84,62 |
| | *RR(3)* | 87,18 | 87,18 | 89,74 | 87,18 | 89,74 | 87,18 | 89,74 | 87,18 |

*Table A.22 VOSy database: Recognition rates obtained with the help of PSB_161 models.*

| PSB_161 | | PCA3 | PCA7 | DODECA | DDPCA | OCTA9 | OCTA33 | RV6 | RV10 |
|---|---|---|---|---|---|---|---|---|---|
| AH | *RR(1)* | 53,85 | 66,67 | 71,79 | 76,92 | 76,92 | 82,05 | 64,10 | 74,36 |
| | *RR(2)* | 71,79 | 82,05 | 76,92 | 84,62 | 82,05 | 89,74 | 79,49 | 84,62 |
| | *RR(3)* | 84,62 | 84,62 | 84,62 | 89,74 | 92,31 | 92,31 | 89,74 | 87,18 |
| | *RR(10)* | 87,18 | 92,31 | 92,31 | 92,31 | 92,31 | 92,31 | 92,31 | 94,87 |
| CS | *RR(1)* | 61,54 | 69,23 | 76,92 | 66,67 | 71,79 | 76,92 | 76,92 | 76,92 |
| | *RR(2)* | 71,79 | 82,05 | 87,18 | 84,62 | 84,62 | 89,74 | 87,18 | 87,18 |
| | *RR(3)* | 79,49 | 82,05 | 89,74 | 87,18 | 84,62 | 92,31 | 89,74 | 89,74 |
| | *RR(10)* | 82,05 | 92,31 | 92,31 | 92,31 | 89,74 | 92,31 | 92,31 | 92,31 |
| HT | *RR(1)* | 23,08 | 41,03 | 56,41 | 41,03 | 56,41 | 64,10 | 56,41 | 46,15 |
| | *RR(2)* | 38,46 | 48,72 | 58,97 | 48,72 | 66,67 | 74,36 | 64,10 | 53,85 |
| | *RR(3)* | 38,46 | 56,41 | 64,10 | 56,41 | 66,67 | 76,92 | 64,10 | 69,23 |
| | *RR(10)* | 61,54 | 82,05 | 84,62 | 79,49 | 82,05 | 84,62 | 84,62 | 84,62 |
| RS | *RR(1)* | 56,41 | 61,54 | 76,92 | 71,79 | 69,23 | 76,92 | 71,79 | 74,36 |
| | *RR(2)* | 61,54 | 71,79 | 79,49 | 76,92 | 76,92 | 82,05 | 76,92 | 79,49 |
| | *RR(3)* | 69,23 | 79,49 | 79,49 | 82,05 | 79,49 | 82,05 | 79,49 | 84,62 |
| | *RR(10)* | 79,49 | 84,62 | 87,18 | 84,62 | 87,18 | 87,18 | 87,18 | 89,74 |
| ZM | *RR(1)* | 48,72 | 64,10 | 74,36 | 66,67 | 69,23 | 79,49 | 74,36 | 71,79 |
| | *RR(2)* | 61,54 | 82,05 | 82,05 | 79,49 | 79,49 | 82,05 | 79,49 | 82,05 |
| | *RR(3)* | 71,79 | 84,62 | 84,62 | 82,05 | 84,62 | 84,62 | 79,49 | 84,62 |
| | *RR(10)* | 84,62 | 87,18 | 89,74 | 87,18 | 89,74 | 87,18 | 89,74 | 89,74 |
| AH + CS | *RR(1)* | 69,23 | 74,36 | 74,36 | 79,49 | 76,92 | 79,49 | 84,62 | 76,92 |
| | *RR(2)* | 79,49 | 84,62 | 89,74 | 87,18 | 84,62 | 87,18 | 87,18 | 87,18 |
| | *RR(3)* | 79,49 | 84,62 | 92,31 | 87,18 | 87,18 | 89,74 | 89,74 | 87,18 |
| | *RR(10)* | 89,74 | 92,31 | 92,31 | 92,31 | 92,31 | 92,31 | 92,31 | 94,87 |
| AH + ZM | *RR(1)* | 66,67 | 74,36 | 82,05 | 76,92 | 76,92 | 82,05 | 82,05 | 82,05 |
| | *RR(2)* | 76,92 | 84,62 | 87,18 | 84,62 | 82,05 | 82,05 | 87,18 | 82,05 |
| | *RR(3)* | 82,05 | 84,62 | 87,18 | 89,74 | 87,18 | 89,74 | 89,74 | 84,62 |
| | *RR(10)* | 87,18 | 92,31 | 92,31 | 92,31 | 92,31 | 92,31 | 92,31 | 94,87 |
| CS + ZM | *RR(1)* | 66,67 | 74,36 | 76,92 | 74,36 | 76,92 | 79,49 | 74,36 | 82,05 |
| | *RR(2)* | 74,36 | 84,62 | 82,05 | 82,05 | 82,05 | 84,62 | 84,62 | 84,62 |
| | *RR(3)* | 76,92 | 84,62 | 89,74 | 82,05 | 84,62 | 89,74 | 87,18 | 87,18 |
| | *RR(10)* | 82,05 | 92,31 | 92,31 | 92,31 | 89,74 | 92,31 | 92,31 | 92,31 |

# REFERENCES


[Adamek04] T. Adamek and N. E. O'Connor, "A multiscale representation method for nonrigid shapes with a single closed contour", *IEEE Trans. Circuits Syst. Video Techn*, Vol.14, Issue 5, pp. 742–753, May 2004.

[Alizadeh95] F. Alizadeh. "Interior point methods in semidefinite programming with applications to combinatorial optimization", *SIAM J. Optim.*, Vol. 5, No. 1, pp. 13–51, 1995.

[Alt98] H. Alt, U. Fuchs, G.Rote, G. Weber, "Matching Convex Shapes with Respect to the Symmetric Difference", *Lecture Notes in Computer Science,* Vol.1136/1996, pp. 320-333, May 1998.

[Ansary07] T. F. Ansary, M. Daoudi, J.-P. Vandeborre, "A Bayesian 3-D Search Engine Using Adaptive Views Clustering", *IEEE Transactions on Multimedia*, Vol. 9, No. 1, pp. 78-88, January 2007.

[Atallah83] M. J. Atallah, "A linear time algorithm for the hausdorff distance between convex polygons," *Information Processing Letters*, Vol. 17, pp. 207–209, 1983.

[Bai07] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting", *IEEE 11th International Conference on Computer Vision* (ICCV'07), pp. 1-8, 2007.

[Bay06] H. Bay, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features", *In European Conference on Computer Vision* (ECCV'06), pp. 404-417, 2006.

[Belongie02] S. Belongie, J. Malik, J. Puzicha, "Shape matching and object recognition using shape contexts", *In IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI'02), Vol. 24, No. 4, pp. 509–522, 2002.

[Bengoetxea02] E. Bengoetxea, "Inexact Graph Matching Using Estimation of Distribution Algorithms", *PhD Thesis*, Ecole Nationale Supérieure des Télécommunications, Paris, France, December 2002.

[Blake04] A. Blake, C. Rother, M. Brown, P. Perez, P. Torr, "Interactive image segmentation using an adaptive GMMRF model", *Proceedings of Computer Vision* (ECCV'04), Vol. 3021, pp. 428–441, 2004.

[Bober02] M. Bober, "MPEG-7 Visual Shape Descriptors", *IEEE Transaction on Circuits and Systems for Video Technology*, Vol.11, Issue 6, pp. 716-719, August 2002.

[Bouman97] C.A. Bouman, M. Shapiro, G.W. Cook, C.B. Atkins and H. Cheng. "Cluster: An Unsupervised Algorithm for Modelling Gaussian Mixtures", School of Electrical Engineering, Purdue University, 1997. http://dynamo.ecn.purdue.edu/~bouman/

[Boykov01] Y. Boykov, M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images", *International Conference on Computer Vision (ICCV)*, Vol. 1, pp. 105–112, 2001.

[Boykov04] Y. Boykov, V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pp. 1124-1137, 2004.

[Canny86] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 6, pp. 679-698, Nov. 1986.

[Chaouch09] M. Chaouch, "Recherce par le contenu d'objets 3D", PhD Thesis, Telecom ParisTech, France, 2009.

[Chen02] D.-Y. Chen, M. Ouhyoung, "A 3D model alignment and retrieval system", *Proceedings of International Computer Symposium, Workshop on Multimedia Technologies*, Hualien,Taiwan, pp. 1436-1443, December 2002.

[Chen03] D.-Y. Chen, X.-P. Tian, Y.-T. Shen and M. Ouhyoung, "On visual similarity based 3D model retrieval", *Computer Graphics Forum*, Vol. 22, No. 3, pp. 223-232, 2003.

[Ciaccia97] P. Ciaccia, M. Patella, F. Rabitti, P. Zezula, "Indexing Metric Spaces with M-tree", *in SEBD 1997*, pp. 67-86, 1997.

[Conte04] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty Years of Graph Matching in Pattern Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 18, No. 3, pp. 265-298, 2004.

[Cormack71] R.M. Cormack, "A Review of Classification", *Journal of the Royal Statistical Society*, Vol. 134, pp. 321–367, 1971.

[Cortona3D-Website] http://www.cortona3d.com/Products/Viewer/Cortona-3D-Viewer.aspx

[Cover67] T. Cover, P. Hart, "Nearest neighbour pattern classification", *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp. 21-27, 1967.

[Cyr01] C. Cyr, B. Kimia, "3D object recognition using shape similarity-based aspect graph", *Proc. 8th IEEE Int. Conf. Comput. Vision*, Vancouver, BC, Canada, pp. 254–261, 2001.

[Dalal05] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", *In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'05), Vol. 1, pp. 886–893, 2005.

[Daras09] P. Daras, A. Axenopoulos, "A Compact Multi-View descriptor for 3D Object Retrieval", *International Workshop on Content-Based Multimedia Indexing*, June 2009.

[Dempster77] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statiscal Sociecty*, Series B, Vol. 39, No.1, pp.1-38, 1977.

[Denton04a] T. Denton, J. Abrahamson, A. Shokoufandeh, "Approximation of canonical sets and their applications to 2D view simplification," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'04), Vol. 2, pp. II-550 - II-557, 27 June-2 July 2004.

[Denton04b] T. Denton, M.F. Demirci, J. Abrahamson, A. Shokoufandeh, S. Dickinson, "Selecting canonical views for view-based 3-D object recognition", *Proceedings of the 17th International Conference on Pattern Recognition* (ICPR'04), Vol.2, No., pp. 273-276 Vol.2, 23-26 Aug. 2004.

[Deselaers10] Deselaers, T., Heigold, G., Ney, H., "Object classification by fusing SVMs and Gaussian mixtures", *Pattern Recognition*, Vol. 43, Issue 7, pp. 2476-2484, July 2010.

[Dubuisson94] M. P. Dubuisson, A. K. Jain, "Modified Hausdorff distance for object matching," in *Proceedings of the IAPR International Conference on Pattern Recognition*, pp. 566–568, 1994.

[Duda72] R. O. Duda, P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM,* Vol. 15, pp. 11–15, January 1972.

[Elad02] M. Elad, A. Tal, S. Ar, "Content Based Retrieval of VRML Objects – An Iterative and Interactive Approach", *Proceedings of the sixth Eurographics workshop on Multimedia*, pp. 107-118, 2002.

[Everingham09] M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman. "The PASCAL Visual Object Classes Challenge", VOC2009 – Results, 2009.

[Felzenszwalb10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained partbased models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, pp. 1627–1645, 2010.

[Fergus03] R. Fergus, P. Perona, A. Zisserman, "Object class recognition by unsupervised scale-invariant learning", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 264-271, June 2003.

[Fernandez08] A. Fernandez, S. Gomez, "Solving nonuniqueness in agglomerative hierarchical clustering using multidendrograms", *Journal of Classification*, Vol. 25, Issue 1, pp. 43-65, June 2008.

[Ferrari04] V. Ferrari, T. Tuytelaars, L. Van Gool, "Integrating Multiple Model Views for Object Recognition", *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR2004), Vol. II, pp. 105-112, 2004.

[Gao11] Y.Gao, Q.i Dai, M. Wang, N. Zhang, "3D model retrieval using weighted bipartite graph matching", *Image Communication*, Vol. 26 No. 1, pp. 39-47, January, 2011.

[Gao12a] Y.Gao, M. Wang,R. Ji, Z. Zha, J. Shen, "K-Partite Graph Reinforcement and Its Application in Multimedia Information Retrieval", *Information Sciences*, Vol. 194, No. 1, pp. 224-239, 2012.

[Gao12b] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, "3-D Object Retrieval and Recognition With Hypergraph Analysis", *IEEE Transactions on Image Processing,* Vol. 21, No. 9, pp. 4290-4303, September 2012.

[Garey79] M.R. Garey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", *Freeman & co*., New York, 1979.

[Glasner11] D. Glasner, M. Galun, S. Alpert, R. Basri, G. Shakhnarovich, "Viewpoint-aware object detection and pose estimation", *2011 IEEE International Conference on Computer Vision* (ICCV), pp. 1275-1282, 2011.

[GrabCutDB] GrabCut image dataset http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm

[Gulshan10] V. Gulshan, C. Rother, A. Criminisi, A. Blake, A. Zisserman, "Geodesic star convexity for interactive image segmentation", *In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3129-3136, 2010.

[Gupta08] N. Gupta, R. Gupta, A. Singh, M. Wytock, "Object Recognition using Template Matching", *Stanford University*, Decembre 2008. http://cs229.stanford.edu/proj2008

[Heisele09] B. Heisele, G. Kim, A. J.Meyer, "Object recognition with 3D models", *In British Machine Vision Conference*, 2009.

[Hodlmoser12] M. Hodlmoser, B. Micusik, M.-Y.Liu, M. Pollefeys, M. Kampel, "Classification and Pose Estimation of Vehicles in Videos by 3D Modeling within Discrete-Continuous Optimization", *Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission* (3DIMPVT), pp. 198-205, 2012.

[Hoeim07] D. Hoeim, C. Rother, J. Winn, "3D LayoutCRF for multi-view object class recognition and segmentation", *In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[ISO/IEC02] ISO/IEC 15938-3: 2002, "MPEG-7-Visual, Information Technology – Multimedia content description interface – Part 3: Visual", 2002.

[ISO/IEC03] ISO/ IEC 15938-5: 2003, "Information technology – MultimediaContent Description. Interface – Part 5: Multimedia Description Schemes", 2003.

[Kadir01] T. Kadir and M. Brady, "Scale, saliency and image description", *International Journal of Computer Vision*, Vol. 45, No. 2, pp. 83–105, 2001.

[Kim99] W.-Y. Kim, Y.-S. Kim, "A New Region-Based Shape Descriptor", ISO/IEC MPEG99/M5472, Maui, Hawaii, December 1999.

[Koenderink76] J. J. Koenderink, A. J. Van Doorn, "The singularilarities of the visual mapping". *Biol. Cyber.*, Vol.24, pp. 51–59, 1976.

[Kushal07] A. Kushal, C. Schmid, J. Ponce, " Flexible object models for category-level 3d object recognition", *In IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'07), pp. 1-8, 2007.

[Kyrki04] S. V. Kyrki, J.K. Kamarainen, "Simple Gabor feature space for invariant object recognition", *Pattern Recognition Letters*, Vol. 25, No. 3, pp. 311-318, 2004.

[Leibe04] B. Leibe and B. Schiele. "Scale-Invariant Object Categorization using a Scale-Adaptive Mean-Shift Search", *In DAGM Annual Pattern Recognition Symposium*, Vol. 3175, pp. 145-153, 2004.

[Leotta11] M. Leotta, J. Mundy, "Vehicle surveillance with a generic, adaptive, 3d vehicle model", *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI'11), Vol. 33, No. 7, pp. 1457–1469, 2011.

[Lewis95] J.P. Lewis, "Fast Template Matching", *Vision Interface 95, Canadian Image Processing and Pattern Recognition Society*, p. 120-123, Canada, May 1995.

[Li06] L. Li, "Data complexity in machine learning and novel classification algorithms", *Ph.D. Dissertation*, California Institute of Technology, 2006.

[Li95] S.Z.Li, "Markov random field modeling in computer vision", Inc. Springer-Verlag New York, 1995.

[Liebelt08] J. Liebelt, C. Schmid, and K. Schertler "Viewpoint-independent object class detection using 3D Feature Maps", *In IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'08), pp. 1-8, 2008.

[Liebelt10] J. Liebelt and C. Schmid, "Multi-view object class detection with a 3D geometric model", *In IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'10), pp. 1688-1695, 2010.

[Liu09] Y. Liu, X.-D. Zhang, Z. Li, H. Li, "3D model feature extraction method based on the projection of principle plane", *Computer-Aided Design and Computer Graphics, 2009 (CAD/Graphics '09),* pp. 463-469, August 2009.

[Liu10] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching", *2010 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp. 1696-1703, 2010.

[Long06] B. Long, X. Wu, Z.M. and Zhang, P.S. and Yu, "Unsupervised learning on k-partite graphs", *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 317-326, 2006.

[Lowe04] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.

[Lowe99] D. Lowe, "Object recognition from local scale-invariant features", *In Proceedings of the International Conference on Computer Vision*, pp. 1150–1157, 1999.

[Mahmoudi02] S. Mahmoudi, M. Daoudi, "3D models retrieval by using characteristic views", *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 457-460, Quebec, Canada, 2002.

[Mahmoudi07] S. Mahmoudi, M. Daoudi, "A Probabilistic Approach for 3D Shape Retrieval by Characteristic Views," *Pattern Recognition Letters*, Vol. 28, No. 13, pp. 1705-1718, 2007.

[Makadia08] A. Makadia, V. Pavlovic, S. Kumar, "A new baseline for image annotation", *In Conputer Vision − ECCV 2008*, Springer Berlin Heidelberg, Part III. LNCS, Vol. 5304, pp. 316–329, 2008.

[Manjunath02] B.S. Manjunath, Phillipe Salembier, Thomas Sikora, "Introduction to MPEG-7: Multimedia Content Description Interface", *John Wiley & Sons, Inc.*, New York, NY, 2002.

[Matas02] J. Matas, O. Chum, M. Urban, T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", *In British Machine Vision Conference* (BMVC), Vol. 1, pp. 384-393, 2002.

[Mather04] P.M. Mather, "Computer processing of remotely-sensed images", 3rd ed. West Sussex, England: John Wiley and Sons Ltd, pp. 1-324, 2004.

[Mikolajczyk02] K. Mikolajczyk, C. Schmid "An affine invariant interest point detector", *In Computer Vision - ECCV*, pp. 128-142, 2002.

[Mitchell97] T. M. Mitchell, "Machine Learning", New York: McGraw-Hill. 1997.

[Mokhtarian92] F. Mokhtarian, A.K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pp. 789-805, August 1992

[Moon96] T.K. Moon, "The expectation-maximation algorithm", *IEEE Signal Processing Magazine*, Vol. 13, No. 6, pp. 47-70, 1996.

[Mukundan08] R. Mukundan, "A Comparative Analysis of Radial-Tchebichef Moments and Zernike Moments", *British Machine Vision Association*, 2008.

[Mukundan98] R. Mukundan, K. R. Ramakrishnan, "Moment Functions in Image Analysis: Theory and Applications", *World Scientific Publishing Co Pte Ltd.*, Vol. 100, September 1998.

[Napoléon07] T. Napoléon, T. Adamek, F. Schmitt, N.E. O'Connor, "Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation", *SHREC 2007 – Shape Retrieval Contest*, Lyon, France, June 2007.

[Napoléon08] T. Napoléon, T. Adamek, F. Schmitt, N. E. O'Connor, "SHREC'08 Entry: Multi-view 3D Retrieval using Multi-scale Contour Representation," *IEEE International Conference on Shape Modeling and Applications* (SMI '08), pp.227-228, 4-6 June 2008.

[Pados94] G.A. Pados, P. Papantoni-Kazakos, "A note on the estimation of the generalization error and prevention of overfitting [machine learning]", *IEEE Conference on Neural Networks*, Vol. 1, pp 321, July 1994.

[Papadakis08] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, S. Perantonis, "3D object retrieval using an efficient and compact hybrid shape descriptor", *Eurographics Workshop on 3D Object Retrieval*, Vol. 5, No. 6, Crete, Greece, 2008.

[Paquet00] E. Paquet, A. Murching, T. Naveen, A. Tabatabai, M. Roux, "Description of shape information for 2-D and 3-D objects", *Signal Processing: Image Commun.*, Vol. 16, pp. 103–122, 2000.

[Patterson08] A. Patterson, P. Mordohai, K. Daniilidis. "Object detection from large-scale 3-D datasets using bottom-up and top-down descriptors", *In Computer Vision -ECCV*, pp. 553-566, 2008.

[Payet11] N. Payet, S. Todorovic, "From Contours to 3D Object Detection and Pose Estimation", *2011 IEEE International Conference on Computer Vision* (ICCV2011), pp. 983-990, 2011.

[Protiere07] A. Protiere, G. Sapiro, "Interactive image segmentation via adaptive weighted distances", *IEEE Transactions on Image Processing*, Vol. 16, No. 4, pp. 1046–1057, April 2007.

[Pu05] J. Pu, K. Ramani, "An Approach to Drawing-Like View Generation From 3D Models", *In Proc. Of International Design Engineering Technical Conferences*, September 2005.

[Radon86] J. Radon, "On the determination of functions from their integral values along certain manifolds," *IEEE Transactions on Medical Imaging,* Vol.5, No.4, pp.170-176, Dec. 1986.

[Rasheed05] Z. Rasheed, M. Shah, "Detection and Representation of Scenes in Videos", *IEEE transactions on Multimedia*, Issue 6, pp. 1097-1105, December 2005.

[Reynolds07] D. Reynolds, "Gaussian Mixture Models", *Encyclopedia of Biometric Recognition*, pp. 659-663, 2007.

[Rhemann09] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, P. Rott. "A perceptually motivated online benchmark for image matting", *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1826–1833, June 2009.

[Rissanen83] J. Rissanen, "A Universal Prior for Integers and Estimation by Minimum Description Length", *The Annals of Statistics*, Vol. 11, No. 2, pp. 416-431, 1983.

[Rother04] C. Rother, V. Kolmogorov, A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts", *ACM Transactions on Graphics*, Vol. 23, No. 3, pp. 309-314, 2004.

[Rubner00] Y. Rubner, C. Tomasi, L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, Vol. 40, No. 2, pp. 99–121, 2000.

[Savarese07] S. Savarese, L. Fei-Fei, "3D generic object categorization, localization and pose estimation", *In IEEE International Conference on Computer Vision* (ICCV'07), pp. 1-8, Octomber 2007.

[Schels11] J. Schels, J. Liebelt, K. Schertler, R. Lienhart, "Synthetically trained multi-view object class and viewpoint detection for advanced image retrieval", *Proceedings of the 1st ACM International Conference on Multimedia Retrieval* (ICMR'11), pp. 3, 2011.

[Schels12] J. Schels, J. Liebelt, R. Lienhart, "Learning an object class representation on a continuous viewsphere", *2012 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'12), pp. 3170-3177, June 2012.

[Schwarz79] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, Vol. 6, pp. 461–464, 1978.

[Schwengerdt97] R.A. Schwengerdt, "Remote Sensing: Models and Methods for Image Processing", 2nd. Ed., Academic Press, 1997.

[Sebastian01] T. B. Sebastian, P. N. Klein, B. B. Kimia, "Recognition of shapes by editing shock graphs," *Proceedings of the Eighth IEEE International Conference on Computer Vision* (ICCV 2001), Vol.1, pp.755-762, 2001.

[Shi94] J. Shi, C. Tomasi, "Good features to track", *In IEEE Computer Society Conference on Computer Vision and Pattern Recogniiton* (CVPR), pp. 593-600, 1994.

[Shih07] J.L. Shih, W.C. Wang, "A 3D Model Retrieval Approach based on The Principal Plane Descriptor", *Proceedings of The Second Internat. Conf. On Innovative Computing, Information and Control (ICICIC)*, pp. 59-62, 2007.

[Shilane04] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, "The Princeton Shape Benchmark", *Shape Modeling International*, pp. 167-178, Genova, Italy, 2004.

[Silverman02] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.24, No.7, pp.881-892, July 2002.

[Smith68] C. Smith, "A characterization of star-shaped sets*", American Mathematical Monthly*, Vol. 75, No. 4, pp. 386, 1968.

[Su09] H. Su, M. Sun, L. Fei-Fei, S. Savarese, "Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories", *In 2009 IEEE International Conference on Computer Vision* (ICCV), pp. 213-220, 2009.

[Sun09] M. Sun, H. Su, S. Savarese, L. Fei-Fei, "A multi-view probabilistic model for 3D object classes", In *In IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp. 1247-1254, 2009.

[Tangelder04] J. Tangelder, R. Veltkamp, "A Survey of Content Based 3D Shape Retrieval Methods", *Proceedings of the Shape Modeling International 2004 (SMI'04)*, pp. 145-156, Genova, Italy, 2004.

[Tapu11] R.G. Tapu, T. Zaharia, "High Level Video Temporal Segmentation", *Proceedings of the 7th international conference on Advances in visual computing*, Vol. 1, pp. 224-235, September 2011.

[Thomas06] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, L. Van Gool, "Towards multi-view object class detection", *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 1589–1596, 2006.

[Tkalcic03] M. Tkalcic, J. F. Tasic, “Colour spaces: Perceptual, historical and application background”, *Proceedings of IEEE EUROCON*, Vol. 1, pp. 304–308, Sep. 2003.

[Tola10] E. Tola, V. Lepetit, P. Fua, "Daisy: An efficient dense descriptor applied to wide baseline stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 5, pp. 815-830, 2010.

[Torralba08] A. Torralba, R. Fergus, Y. Weiss, "Small codes and large image databases for recognition", *In IEEE Conference on Computer Vision and Pattern Recognition* (CVPR 2008), pp. 1–8, 2008.

[Toshev09] A. Toshev, A. Makadia, K. Daniilidis, "Shape-based Object Recognition in Videos Using 3D Synthetic Object Models", *IEEE Conference on Computer Vision and Pattern Recognition*, Vol.60, No. 2, pp. 91-110, Miami, FL, 2009.

[Vapnik95] V. N. Vapnik, "The Nature of statistical learning theory", *New York: Springer-Verlag*, 1995.

[Veksler08] O. Veksler, "Star shape prior for graph-cut image segmentation", *In Proceedings of the 10th European Conference on Computer Vision*, Part III, pp. 454–467, 2008.

[Vranic01] D.V. Vranic, D. Saupe, J. Richter, "Tools for 3D-object retrieval: Karhunenloeve trans-form and spherical harmonics", *In IEEE 2001 Workshop Multimedia Signal Processing*, pp. 293-298, 2001.

[Vranic04] D.V. Vranic, "3D Model Retrieval", *PhD Thesis*, Univeristy of Liepzig, 2004.

[Weber00] M. Weber, M. Welling, P. Perona, "Unsupervised learning of models for recognition". *In Proc. ECCV*, pp. 18–32, 2000.

[Wen08] L. Wen, G. Tan, "Enhanced 3D Shape Retrieval Using View-Based Silhouette Representation", *International Conference on Audio, Language and Image Processing*, pp. 928-931, August 2008.

[Xue09] M. Xue, C. Zhu, "A Study and Application on Machine Learning of Artificial Intelligence", *International Joint Conference on Artificial Intelligence*, pp. 272, July 2009.

[Yamauchi06] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. Belyaev, H.-P. Seidel, "Towards Stable and Salient Multi-View Representation of 3D Shapes", *IEEE International Conference On Shape Modeling and Applications,* pp.40-40, 14-16 June 2006.

[Yang03] C. Yang, R. Duraiswami, N. Gumerov, L. Davis, "Improved fast Gauss transform and effcient kernel density estimation", *Ninth IEEE International Conference on Computer Vision*, pp. 664–671, 2003.

[Yang08] T. Yang, B. Liu, H. Zhang, "3D model retrieval based on exact visual similarity", *9th IEEE International Conference on Signal Processing*, pp. 1556-1560, December 2008.

[Yap03] P.T.Yap, R.Paramesran, S.H.Ong, "Image Analysis by Krawtchouk Moments", *IEEE Transactions on Image Processing*, Vol. 12, No. 11, pp. 1367-1377, November 2003.

[Zaharia01] T. Zaharia, F. Prêteux, "3D shape-based retrieval within the MPEG-7 framework", *Proceedings of SPIE Conference On Nonlinear Image Processing and Pattern Analysis XII*, Vol. 4304, pp.133-145, San Jose, CA, USA, January 2001.

[Zaharia02] T.Zaharia, F. Preteux, "Shape-based retrieval of 3D mesh models", *Proceedings of 2002 IEEE International Conference on Multimedia and Expo,* (ICME02). Vol. 1, pp. 437-440, August 2002.

[Zaharia04] T. Zaharia, F. Prêteux, " 3D versus 2D/3D Shape Descriptors: A Comparative study", *In SPIE Conference On Image Processing: Algorithms and Systems*, Vol. 2004, pp. 47-58, Toulouse, France, January 2004.

[Zahn72] C. T. Zahn, R. Z. Roskies, "Fourier Descriptors for Plane closed Curves", *IEEE Transactions On Computers*, Vol. 21, No. 3, pp. 269-281, 1972.

# GLOSSARY

## ACRONYMS

*a priori* – not based on prior study or examination

*cf.* – *confer* (compare)

*i.e.* – *id est* (in other words)

*e.g.* – *exempli gratia* (for example)

*et al.* – *et alii* (and others)

*vice-versa* – with position turned, in reverse order from the way something has been stated

AH – Angular Histogram (0)

ART – Angular Radial Transform (III.2.2.1)

AVC – Adaptive Views Clustering (II.2.3)

B – background (V.2)

BCS – Bounded Canonical Set (II.2.3)

BIC – Bayesian Information Criteria (II.2.3)

BMP – bitmap image file format

CMVD – Compact Multi-View Descriptor (II.2.2)

CPCA – Continuous Principal Component Analysis (II.1.1.2)

CS – Contour Shape (III.2.2.4)

DB – Database

DDPCA – Dodecahedron-based viewing angle selection for an aligned 3D model (III.2.1.2)

DODECA – Dodecahedron-based viewing angle selection (III.2.1.2)

DS – Description Scheme

EM – Expectation Maximization (V.2)

ESA – Enhances Silhouette-based Approach (II.2.1)

ESI – Enhanced Silhouette Intersection (II.2.2)

F – foreground (V.2)

FT – First Tier (III.6.1)

GMM – Gaussian Mixture Model (V.3.1)

GPU – Graphics processing unit

HAC – Hierarchical Agglomerative Clustering (II.2.3)

HT – Hough Transform (III.2.2.2)

JPEG – Joint Photographic Experts Group image file format

LFD – Light Field Descriptor (II.2.2)

LUT – Look-up table

Luv – The Luv colour space (V.2)

MCC – Multi Curve Convexity/Concavity (II.2.1)

ML – machine learning (0)

MLFD – modified Light Field Descriptor (II.2.2)

MPEG – Moving Picture Experts Group

MPEG7_23 – MPEG7 database (the classification includes 23 categories) (III.5.1)

MRF – Markov Random Field (IV.2)

NN – Nearest Neighbour (IV.2)

OCTA33 – Octahedron-based viewing angle selection – 33 views/model (III.2.1.3)

OCTA9 – Octahedron-based viewing angle selection – 9 views/model (III.2.1.3)

OS – Overlap score (V.4)

PCA – Principal Component Analysis (II.1.1.2)

PCA3 – PCA-based viewing angle selection – 3 views/model (III.2.1.1)

PCA7 – PCA-based viewing angle selection – 7 views/model (III.2.1.1)

PDF – The probability density function (V.2)

PNG – Portable Network Graphics image file format

POI – Point of Interest (IV.2)

PR – Precision-Recall curve (III.6.1)

PSB_161 – Princeton Shape Benchmark database (the classification includes 161 categories) (III.5.2)

PSB_53 – Princeton Shape Benchmark database (the classification includes 53 categories) (III.5.2)

RR – The recognition rate (IV.4.2)

RS – Region Shape descriptor (III.2.2.1)

RV10 – Representative views selection – 10 views/model (III.2.1.4)

RV6 – Representative views selection – 6 views/model (III.2.1.4)

SI – Silhouette Intersection (II.2.1)

SIFT – Scale Invariant Feature Transform (IV.2)

SO – Still object (IV.3.1)

SOI – Still objects from images database (IV.4.1.1)

SOSy – Still objects from synthetic images database (IV.4.1.3)

SOV – Still objects from videos database (IV.4.1.2)

SQL – Structured Query Language

ST – Second Tier score (III.6.1)

SURF – Speeded Up Robust Features (IV.2)

SVM – Support Vector Machine (IV.2)

UI – User Interface

V – Video (IV.3.2)

VCA – Vertex Component Analysis (II.2.2)

VO – Video object (IV.3.2)

VOSy – Video objects from synthetic images database (IV.4.1.3)

VOV – Video objects from videos database (IV.4.1.2)

VRML – Virtual Reality Modelling Language (A1)

WebGL – Web Graphics Library

ZM – Zernike Moments (III.2.2.3)

## NOTATIONS

$\Delta(X,Y)$ – The distance between two semantic classes X and Y (III.5.3.1)

$\Delta_{DB}$ – The mean distance between each two semantic classes of a database DB (III.5.3.1)

$\Delta_X$ – The mean distance between the semantic class X and the other classes of the database (III.5.3.1)

$\alpha_l$ – Parameter used to enlarge the set of labelled pixels (V.3.1)

$\beta$ – The elongation parameter (V.3.3)

$\delta_{DB}$ – The average intra-class variability of a database DB (III.5.3.1)

$\delta_X$ – The intra-class variability of the semantic category X (III.5.3.1)

$\varepsilon$ – The probability threshold (V.3.3)

$\lambda_i$ – eigenvalues (V.3.3)

$\Sigma_i$ – The covariance matrix of the ith GMM component (V.3.1)

$\Sigma_{ij}$ – The element ij of the covariance matrix (II.1.1.2)

$\omega_{Ch}{}^{i}$ – The weighting coefficient of the ith channel (V.2)

$\omega_i$ – The weight of the ith GMM component (V.3.1)

$c(u)$ – The sampled contour of a shape

$D(M_A, M_B)$ – The distance between the 3D model MA and the 3D model MB (III.3)

$d(X, Y)$ – The distance between two 2D shapes X and Y (II.1.4.1)

$d^l(x)$ – The geodesic distance between pixel x and label l (V.2)

$F_i(V)$ – The ith frame of the video V (IV.3.2)

$g(X)$ – The GMM probability density function (V.3.1)

$g_i(X)$ – The ith component of the GMM (V.3.1)

$I_i(VO)$ – The ith instance of the video object VO (IV.3.2)

$l$ – The set {F, B} of possible labels (V.2)

$L^l$ – The length of l scribbles (V.3.1)

$N_{C/X}$ – The number of correct elements among the top X retrieved (III.6.1)

$N_{Ch}$ – The number of channels (V.2)

$N_F$ – The number of frames selected from each video (IV.3.2)

$N_I$ – The number of instances that compose a video object (IV.3.2)

$N^l$ – The number of pixel in the l set (V.3.1)

$N_M$ – The number of 3D models in the database (III.5)

$N_{MRC}$ – The number of the most represented categories (within the NTRM first models) which are proposed as response to a query (IV.3)

$N_P$ – The number of 2D projections associated to a 2D model

$N_Q$ – The number of elements in the database which are similar to the query (III.6.1)

$N_R$ – The number of elements retrieved from the database (III.6.1)

$N_{RV}$ – The number of representative views selected to describe a 3D model (III.2.1.4)

$N_t^l$ – The target number of pixel in the l set (V.3.1)

$N_{TRM}$ – The number of top retrieved models which are taken into account in the recognition process (IV.3)

$PDF_i^l$ – The probability density function of the ith channel w.r.t. the l label (V.2)

$P_i(M)$ – The ith projection of model M

$P^l(x)$ – The likelihood of the pixel x to belong to the l label (V.2)

$RR(NMRC)$ – The recognition rate obtain when NMRC categories are considered (IV.4.2)

$RR(O,N_{MRC})$ – The recognition label associated to object O when NMRC categories are considered (IV.4.2)

$S(X,Y)$ – The separability between two semantic classes X and Y (III.5.3.1)

$S_{DB}$ – The mean inter-class separability between the categories of a database DB (III.5.3.1)

$S_X$ – The separability of the semantic class X (III.5.3.1)

$W^l(x)$ – The weight of the pixel x on the likelihood map associated to the label l (V.2)

$X_x$ – A D-dimensional vector storing the colour components of pixel x (V.3.1)

# ABSTRACT

Automatic classification and interpretation of 2D objects is a key issue for various computer vision applications. In particular, when considering image/video indexing and retrieval applications, automatically labelling, in a semantically pertinent manner, still remains an open challenge, especially when huge multimedia databases are involved.

This Ph.D. thesis tackles the issue of sill and video object categorization. The objective is to associate semantic labels to 2D objects present in natural images/videos. The principle of the proposed approach consists of exploiting categorized 3D model repositories in order to identify unknown 2D objects based on 2D/3D matching techniques. Notably, we use view-based indexing methods, where 3D models are described through a set of 2D views.

We propose here an object recognition framework, designed to work for real time applications. The similarity between classified 3D models and unknown 2D content is evaluated with the help of the 2D/3D description. A voting procedure is further employed in order to determine the most probable categories of the 2D object.

The highest recognition rates obtained on real objects were up to 74% for still objects and up to 85% for video objects. When three categories are accepted as response, the same scores were up to 86%, respectively 93%.

In our work, we consider several state of the art projection strategies and 2D shape descriptors. In addition, a representative viewing angle selection strategy (so-called RV) and a new contour based descriptor (so-called AH), are proposed. The experimental evaluation proved that, by employing the intelligent selection of views, the number of projections can be decreased significantly (up to 5 times) while obtaining similar performance. The results have also shown the superiority of AH with respect to other state of the art descriptors.

An objective evaluation of the intra and inter class variability of the 3D model repositories involved in this work (*i.e.*, MPEG-7 and Princeton datasets) is also proposed, together with a comparative study of the retained indexing approaches within the framework of 3D model retrieval.

An interactive, scribble-based segmentation approach, designed to facilitate the task of 2D object extraction, is also introduced. The proposed method is based on colour distributions, estimated with Gaussian Mixture Models (GMM), and is specifically designed to overcome compression artefacts such as those introduced by JPEG compression.

We finally present an indexing/retrieval/classification Web platform, so-called DIANA – *Digital Image Analysis aNd Annotation*, which integrates the various methodologies proposed in this thesis.