Indexing of AV Content

Contents

- Introduction
- Metadata Lifecycle
- Classical features for indexing
  - Textual and basic semantic features
  - Content Management features
- AV features for indexing
  - Spatio-temporal structure features
  - Low-level features
  - Mid-level features
  - High-level features
- Bridging the Semantic Gap
- Collections of AV contents
- Metrics between descriptors and relevance feedback
- Conclusions

Indexing of AV Content

Contents

- Introduction
- Metadata Lifecycle
- Classical features for indexing
  - Textual and basic semantic features
  - Content Management features
- AV features for indexing
  - Spatio-temporal structure features
  - Low-level features
  - Mid-level features
  - High-level features
  - Bridging the Semantic Gap
- Collections of AV contents
- Metrics between descriptors and relevance feedback
- Conclusions

Introduction

Motivation

Digital multimedia content

- Whole digital archives (digital libraries) of images, videos, music, speech, … are being created, guaranteeing an everlasting quality of the stored AV documents
  - **Storage** of huge amounts of digital media
- More and more audiovisual information is available on-line in digital form: still images, (graphics, 3D models,) audio, speech, video, composition information (scenes), ...
  - **Coding and delivery** of multimedia
  - Customized *variations/adaptations* of content => ***Universal Multimedia Access***

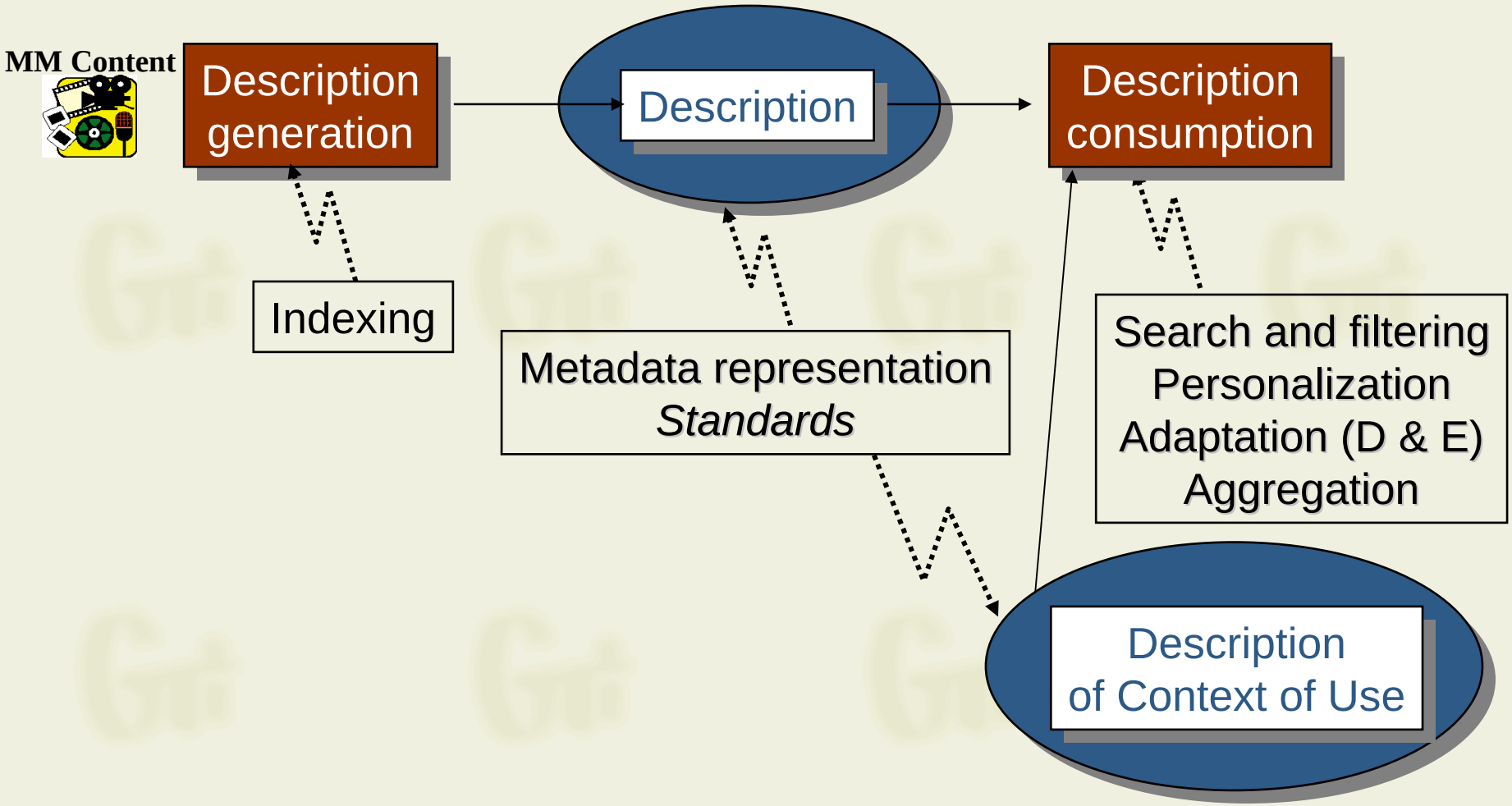
Introduction

Motivation

Applications and services become multimedia and interactive

- Rapid development of innovative tools for effective multimedia interactive applications
- Need of information about the content in order to
  - manage content
  - find, select and filter content
  - use of content by computational systems and agents
  - Customization of content (items and presentation) => ***Universal Multimedia Access***
- **Indexing** (cataloguing, annotating) for searching, filtering, interpreting, adapting, aggregating, … multimedia content
  - Not only the content needs to be indexed, also the user, the terminal, the network, the environmental conditions, …
  - The target is to enhance the user's experience when consuming multimedia (what, when, how s/he wants)

Introduction



Introduction

Indexing

Conventional (classical) systems

- Multimedia content is annotated manually with textual information
  - Mainly without any granularity (shots, scenes, objects, …)
  - Human interaction is time consuming (besides expensive)
  - Media characteristics can be extracted automatically (mainly)
  - Semantic based but biased by human subjectivity

    Ontologies, controlled terms, …
- Information about the content (classical)
  - recording date and context, title, author, copyright data, conditions for access, media profiles, coding format, genre, parental rating, links to related material, abstract (natural language), basic semantic classification, keywords, …

Introduction

Indexing

Content based (*"innovative"*) systems

- Multimedia content is annotated (automatically/supervised) with content based descriptors (metadata)
  - Efficiency (speed, resources, …)
  - Accuracy (effectiveness)
  - Low/mid/high level descriptors:

Information present in the content (innovative)

- Low level features allowing automatic extraction such as: (V) colour, texture, shape, position, motion, …; (A) key, mood, tempo, timbre, …; and (MDS) spatio-temporal structure, AV objects, …
- "Mid" level features: syntactic relationships about "signal" objects (close to, moves towards, …), type of object recognition (persons, sky, grass, tress?, …), …
- High level (semantic) features ("real world" events and objects, and semantic relationships) related to human annotation and interpretation of the content: "boy picking up flowers for his mother while the dog plays around", "beautiful sunset from the dock of the bay", ...

Introduction

Searching: querying

Text (free text, controlled vocabularies, keywords)

- Find AV material with subject corresponding to some keywords

Semantic description

- Find AV material corresponding to a specified semantic (places, people, objects, events, ...)

Audio: a few notes of music (melody and rhythm feature)

- Find corresponding musical pieces or movies

Video: motion trajectory (low level video feature) sketch

- Find video with specific object motion trajectories (e.g., similar player movements in a soccer game, intrusion into safe zone)

Image as an example:

- Find an image with similar global or local characteristics (e.g., find logo on a TV channel)

Introduction

Search and browse

Pull services

- Play a few notes on a keyboard and retrieve a list of musical pieces similar to the required tune, or images matching the notes in a certain way, e.g. in terms of emotions.
- Draw a few lines on a screen and find a set of images containing similar graphics, logos, ideograms,...
- Define objects, including colour patches or textures and retrieve examples among which you select the interesting objects to compose your design.
- On a given set of multimedia objects, describe movements and relations between objects and so search for animations fulfilling the described temporal and spatial relations.
- Describe actions and get a list of scenarios containing such actions.
- Using an excerpt of Pavarotti's voice, obtain a list of Pavarotti's records, video clips where Pavarotti is singing and photographic material portraying Pavarotti.

Introduction

Filtering

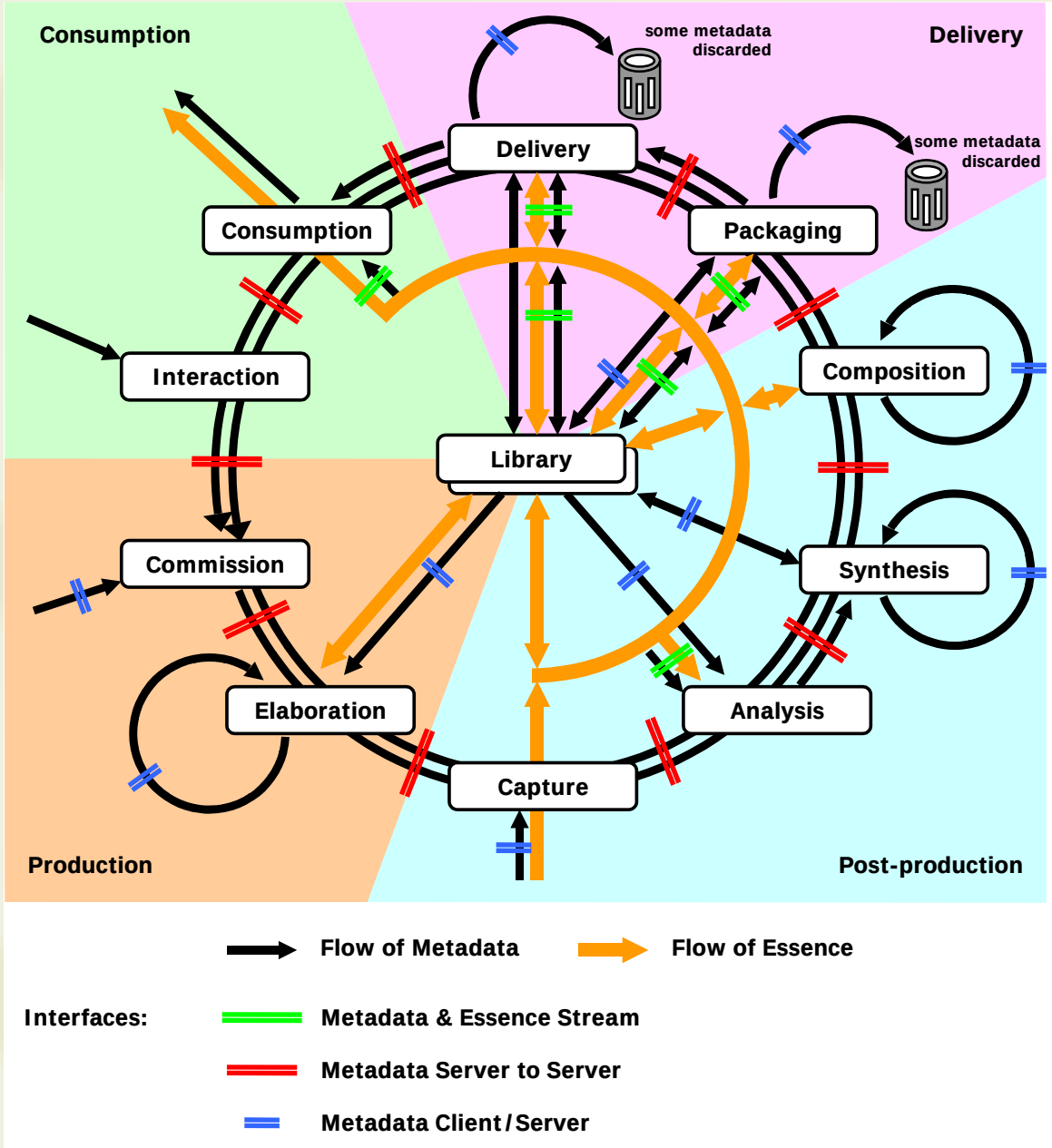
Push services

- Broadcast/streaming selection by agents
  - Automatic zapping
- Broadcast/streaming activation by agents
  - Background storage
  - Personalized composition

Indexing of AV Content

Contents

Metadata Lifecycle



Indexing of AV Content

Contents

- Introduction
- Metadata Lifecycle
- Classical features for indexing
  - Textual and basic semantic features
  - Content Management features
- AV features for indexing
  - Spatio-temporal structure features
  - Low-level features
  - Mid-level features
  - High-level features
- Bridging the Semantic Gap
- Collections of AV contents
- Metrics between descriptors and relevance feedback
- Conclusions

Classical features for indexing

Classical systems (archives, libraries, …) rely on textual annotations

- Descriptive semantic annotation (e.g., abstract, keywords, …)

They are aimed to Content Management

- Information about the media
- Information about the use
- Information allowing some classification, search, …

Several standards: MARC, Z39.50, Dublin Core, P/Meta, SMPTE,<???>ML (NewsML, Sports ML, …), ID3, EXIF, … but mainly “proprietary/stand-alone/domain-dependent” solutions

For completeness, MPEG-7 also covers *classicism*

Textual and Basic Semantic features

In order to describe things textually several options are available:

- Full free text
- Keywords
- Structured text (incomplete or complete)

In order to provide semantics ontologies/dictionaries/thesauri are needed

Some “classical” semantic entities are common

- Persons, organizations, places, …

Textual and Basic Semantic features: the MPEG-7 proposal

- Language description tools
  - xml:lang attribute (descriptions)
  - Language datatype (content related)
- Textual Annotation
  - Textual datatype (xml:lang)
  - TextAnnotation
    - free text
    - structured annotation (6w+how)
    - keyword annotation
    - dependency structure

Textual and Basic Semantic features: the MPEG-7

- Language des
  - o xml:lang attr
  - o Language datatype iated)
- Textual Annotation
  - o Textual datatyp ml:lang)
  - o TextAnnotati
    - free text
    - structured annotation (6w+how)
    - keyword annotation
    - dependency structure

```
<TextAnnotation>
        <FreeTextAnnotation xml:lang="en">
        Tanaka throws a small ball to Yamada.
        </FreeTextAnnotation>
</TextAnnotation>
```

Textual and Basic Sem
pr

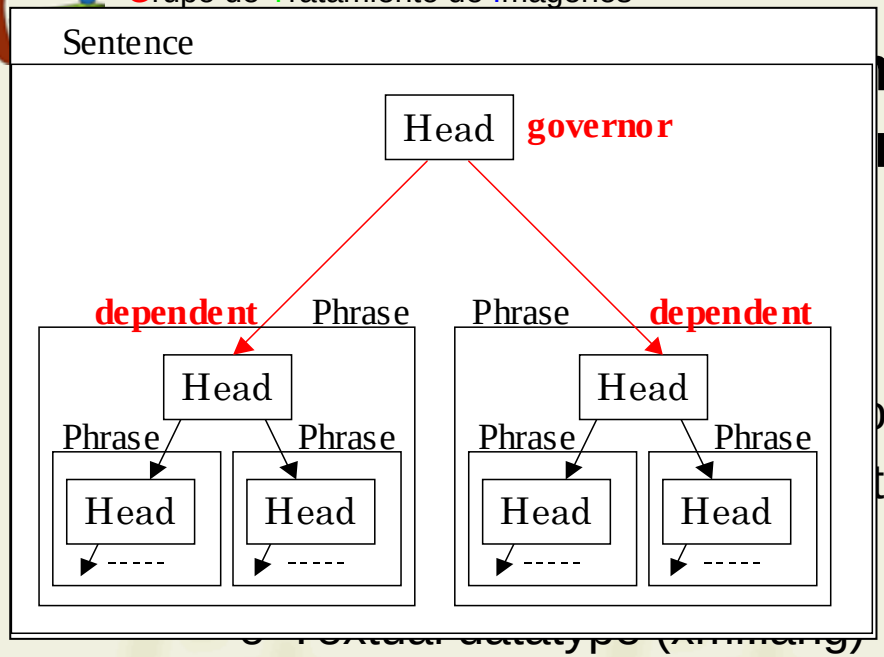
- Language description tools
  - xml:lang attribute (descriptio
  - Language datatype (content
- Textual Annotation
  - Textual datatype (xml:lang)
  - TextAnnotation
    - free text
    - structured annotation (6w
    - keyword annotation
    - dependency structure

```
<TextAnnotation>
<StructuredAnnotation>
<Who>
<ControlledTerm term="TANAKA100”
scheme="JapaneseName”
schemeLocation="http://PersonDic.com">
<Label xml:lang="en">Tanaka</Label>
</ControlledTerm>
</Who>
<Who>
<FreeTerm xml:lang="en">Yamada</FreeTerm>
</Who>
<WhatObject><FreeTerm xml:lang="en">
        A small ball</FreeTerm>
</WhatObject>
<WhatAction>
        <FreeTerm xml:lang="en">
        Tanaka throws a small ball to Yamada.
        </FreeTerm>
</WhatAction>
</StructuredAnnotation>
</TextAnnotation>
```

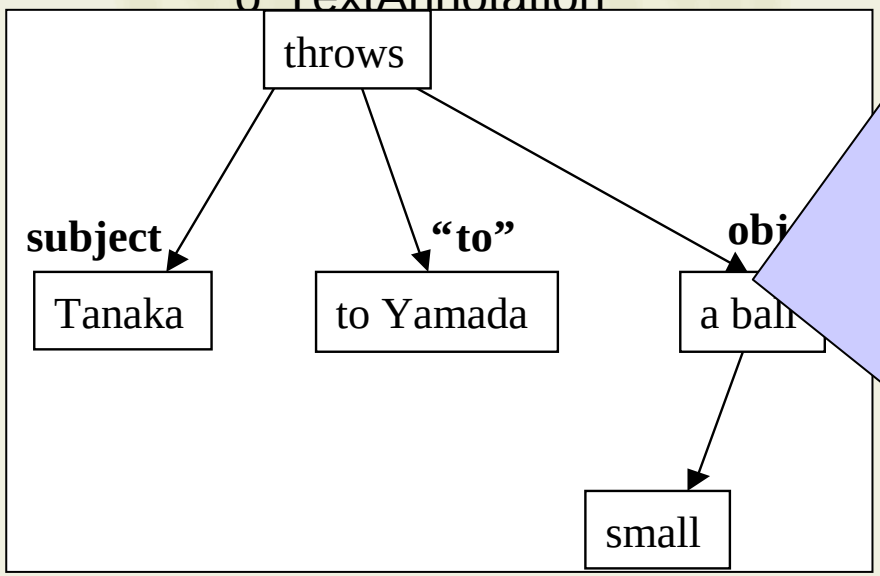
Textual and Basic Semantic features: the MPEG-7 proposal

- Language descri
  - xml:lang attribu
  - Language data
- Textual Annotatio
  - Textual datatyp
  - TextAnnotation
    - free text
    - structured annotation (
    - keyword annotation
    - dependency structure

```
<TextAnnotation>
    <KeywordAnnotation xml:lang="en">
            <Keyword>Tanaka</Keyword>
            <Keyword>Yamada</Keyword>
            <Keyword>ball</Keyword>
     </KeywordAnnotation>
</TextAnnotation>
```



o TextAnnotation




```
<TextAnnotation>
<DependencyStructure xml:lang="en">
<Sentence>
<Phrase operator="subject">
<Head type="noun”
baseForm="Tanaka">Tanaka</Head>
</Phrase>
<Phrase operator="object">
        <Phrase>
        <Head type="adjective"
baseForm="small">small</Head>
        </Phrase>
        <Head type="noun" baseForm="ball">a
ball</Head>
</Phrase>
<Phrase particle="to">
        <Head type="noun"
baseForm="Yamada">to Yamada</Head>
</Phrase>
<Head type="verb"
baseForm="throw">throws</Head>
</Sentence>
</DependencyStructure>
</TextAnnotation>
```

Textual and Basic Semantic features: the MPEG-7 proposal

- Language description tools
- Textual Annotation
- Controlled Vocabularies (thesauri, ontologies)
  - Classification Schemes
    - scheme identifier, version, CS reference/items
    - items: label (multilingual), definition (multilingual), CS reference/items, relation
  - Controlled Term
    - term identifier, scheme identifier, scheme locator (compact representation – by reference)
    - label, definition, term relation (complete representation – redundant)
  - Term
    - Free text or Controlled Term

Textual and Basic Semantic features: the MPEG-7 proposal

- Language description tools
- Textual Annotation
- Controlled Vocabularies (thesauri, ontologies)
- Agents
  - Person
    - Person Name, Affiliation (Organisation or Group), e-address
  - Person Group
    - Name, Members, Kind, Jurisdiction, e-address
  - Organisation
    - Name, Kind, Jurisdiction, Address, Contact, e-address
- Place
  - Names, role, country, region, GPS coordinates, Postal Address

Content Management features

- Content Management refers to the “classical” archival information
  - Content entity information (media, modality, quality, preservation, …)
  - Creation information (author, title, genre, …)
  - Usage information (availability, rights, …)
- Currently most of the existing AV archives are based on proprietary (and ad-hoc) Content Management data models
  - hard to be replaced (really personalized systems and “idiosyncratic” users)
- The problem is the interoperability of the archives, therefore some standards (de-jure and proprietary) have been proposed
  - “Commercial” product based: Virage, Tarsys, BBC model, audio jukeboxes (ID3 or proprietary ones), …
  - MPEG-7, SMPTE Metadata dictionary, P/META, JPSearch (under development), …

Content Management features: the MPEG-7 proposal

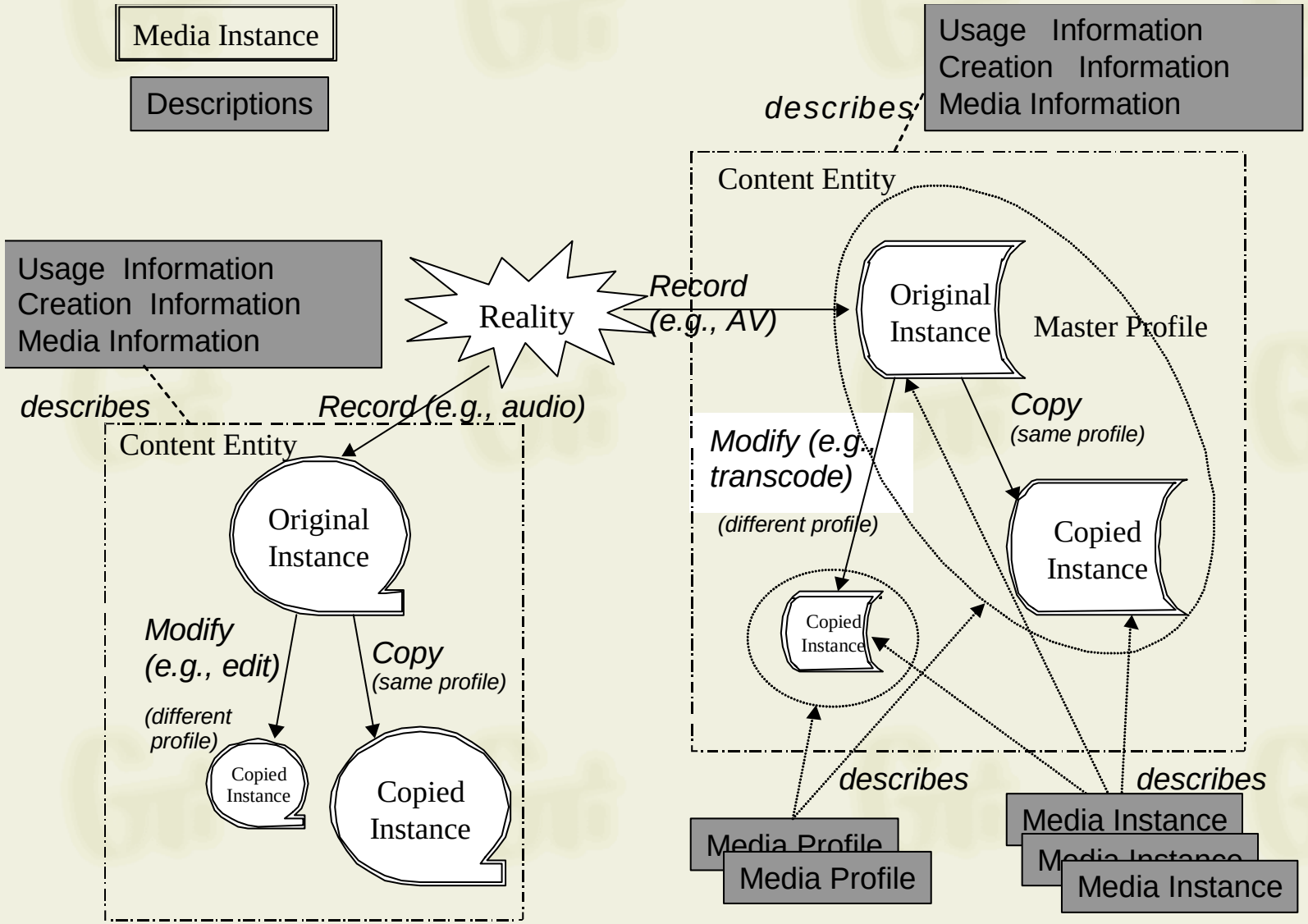
MPEG-7 seems to be the more generalist approach

- it targets archives of images, graphics, audios, videos, and combined collections.
- it covers the three main content management topics for a Content Entity: Media information, Creation information, Usage information
  - Nevertheless, it lacks preservation information and good support for “living updates” (description of content from drafts to “final” version, e.g., news coverage)

MPEG-7 Content Management description tools are grouped in:

- Media Information
- Creation Information
- Usage Information

Content Management features: the MPEG-7 proposal



Content Management features: the MPEG-7 proposal

Creation information:
Creation
Creator
Creation coordinates
Creation location
Creation date

Usage information:
Rights

Photographer: Seungyup
Place: Columbia University
Time: 19 September 1998

Columbia University,
All rights reserved

Media information:
Media profile
Media format
Media instance

master
704x480 pixels
jpg
http://www.ee.columbia.edu/~ana/alex&ana.jpg

Media information:
Media profile
Media format
Media instance

352x240 pixels
gif
http://www.ee.columbia.edu/~ana/alex&ana.gif
http://www.ee.columbia.edu/~alex/alex&ana.gif

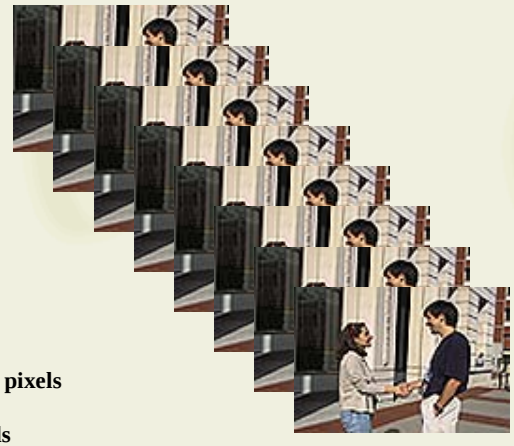
Image Content Entity
Two Profiles, Three Instances

Creation information:
Creation
Creator
Creation coordinates
Creation location
Creation date

Usage information:
Rights

Camera: José
Place: Columbia University
Time: 19 September 1998

Columbia University,
All rights reserved



master
640x480 pixels
mpg
5 seconds
http://www.ee.columbia.edu/~ana/alex&ana.mpg

Media information:
Media profile
Media format
Media instance

Video Content Entity
One Profile, One Instance

Content Management features: the MPEG-7 proposal

Media Information

- Media Identification
  - o entity identifier, domains
- Media Profiles

Name	*Definition*
`MediaIdentificationType`	Identifies the content entity independently of the different available profiles and instances.
`EntityIdentifier`	Identifies uniquely the particular and unique multimedia content entity. For example, ISO's ISAN.
`AudioDomain`	Describes the source, acquisition, and/or use of an audio content entity (optional). If a content entity is AV, AudioDomain and VideoDomain are used simultaneously. An example of CS is `AudioDomainCS`.
`VideoDomain`	Describes the source, acquisition, and/or use of a video content entity (optional). An example of CS is `VideoDomainCS`.
`ImageDomain`	Describes the source, acquisition, and/or use of an image content entity (optional). An example of CS is `ImageDomainCS`.

Content Management features: the MPEG-7 proposal

Media Information

- Media Identification
  - entity identifier, domain
- Media Profiles
  - Media Instances

Name	*Definition*
`MediaInstanceType`	Describes the identification and the location of Media Instances.
`InstanceIdentifier`	Identifies the Media Instance or copy.
`MediaLocator`	Describes the Media Locator of a Media Instance.
`LocationDescription`	Describes the location of a Media Instance not available via a Media Locator. For example, the location of tapes stored in the analogue archive of a broadcaster.

Content Management features: the MPEG-7 proposal

Name	*Definition*
`MediaFormatType`	Describes the format and coding parameter of the Media Profile.
`Content`	Describes the media present in the Media Profile (e.g., audio, image, scene definition, video, audiovisual). An example of CS is `ContentCS`.
`Medium`	Describes the physical storage medium on which the Media Profile is stored (optional). An example of CS is `MediumCS`.
`FileFormat`	Describes the file format of the Media Profile (optional). An example of CS is `FileFormatCS`.
`FileSize`	Indicates the size, in bytes, of the file where the Media Profile is stored (optional).
`System`	Describes the broad media format of the Media Profile (optional). An example of CS is `SystemCS`.
`Bandwidth`	Indicates the bandwidth range in Hz covered by the coded multimedia content (optional). Its value dependents on the acquisition filters or transcoding applied to the Media Profile.
`BitRate`	Indicates the nominal bit rate in bit/s of the Media Profile (optional).
`variable`	Indicates whether the bitrate is variable or fixed. If the bitrate is variable, three optional attributes can be used to specify the minimum, maximum and average bitrates. Default value of this attribute is *false*.
`minimum`	Indicates the minimum numerical value for the `BitRate` in case of variable bit rate (optional).
`maximum`	Indicates the maximum numerical value for the `BitRate` in case of variable bit rate (optional).
`average`	Indicates the average numerical value for the `BitRate` in case of variable bit rate (optional).
`TargetChannelBitRate`	Indicates the nominal bit rate in bits/s of the channel for which this MediaProfile is targeted (optional).
`ScalableCoding`	Indicates the scalability used in the coding (optional). The values of the scalable coding are defined as follows:. •*spatial* – The scalability is spatial. •*temporal* –- The scalability is temporal. •*snr* –- The scalability is SNR (Signal to Noise Ratio). •*fgs* –- The scalability is FGS (Fine Granularity Scalability). Other values that are datatype-valid with respect to `mpeg7:termReferenceType` are reserved.
`VisualCoding`	Describes the coding of the visual component of the Media Profile (optional). If a content entity is AV, `VisualCoding` and `AudioCoding` are used simultaneously.
`AudioCoding`	Describes the coding of the audio component of the Media Profile (optional). If a content entity is AV, `VisualCoding` and `AudioCoding` are used simultaneously.
`SceneCodingFormat`	Describes the coding format for a scene composition stream (optional), for example a BIFS stream in MPEG-4. An example of CS is `SceneCodingFormatCS`.
`GraphicsCodingFormat`	Describes the coding format for a graphics stream (optional), for example VRML. An example of CS is `GraphicsCodingFormatCS`.
`OtherCodingFormat`	Describes other coding formats that are not visual, audio, scene description or graphics such as HTML or Flash (optional). An example of CS is `OtherCodingFormatCS`.

Content Management features: the MPEG-7 proposal

Media Information

- Media Identification
  - entity identifier, domain
- Media Profiles
  - Media instances
  - Media Format
  - Component Media Profiles (multiplexed media)
  - Media Transcoding Hints
  - Media Quality

Creation Information

Usage Information

Name	*Definition*
`MediaTranscodingHintsType`	Describes transcoding hints of the Media Profile.
`MotionHint`	Describes the motion hints for a transcoder (optional).
`MotionRange`	Describe Motion range for a transcoder (optional).
`xLeft`	Indicates the recommended integer pel search range for horizontal motion vectors to the left.
`xRight`	Indicates the recommended integer pel search range for horizontal motion vectors to the right.
`yDown`	Indicates the recommended integer pel search range for vertical motion vectors to the bottom.
`yUp`	Indicates the recommended integer pel search range for vertical motion vectors to the top.
`uncompensability`	Describes the amount of new content, that is content that cannot be predicted, in the corresponding segment (this descriptor applies to descriptions attached to video segments). The `uncompensability` takes values from 0.0 to 1.0, where 0.0 indicates no new content and 1.0 indicates the highest change in content. 1.0 means that nothing can be predicted and a transcoder/encoder may choose a non-predictive encoding mode (optional).
`intensity`	Describes the motion intensity in a segment. The `motionIntensity` takes values from 0.0 to 1.0, where 0.0 indicates low motion intensity and 1.0 indicates the highest motion intensity (optional).
`ShapeHint`	Shape hints for the transcoder (optional).
`shapeChange`	Describes the amount of shape change in the corresponding segment (this descriptor applies to descriptions attached to moving regions). The `shapeChange` takes values from 0.0 to 1.0, where 0.0 indicates that no change has occurred and 1.0 indicates that all the pixels that define an object have been displaced (optional).
`numOfNonTranspBlocks`	Describes the average number of 16x16 blocks per frame containing at least one pixel with a non-zero alpha-map value (optional).
`CodingHints`	Coding hints for the transcoder (optional).
`avgQuantScale`	Describes the average quantization scale used to compress the media according to the compression format applied (optional).
`intraFrameDistance`	Describes the distance between Intra-coded Frames, also known as N (optional). A value of N=0 represents the case when N is infinite, for example, when the GOP has no I-frame (PBBPBBP..) or when there is only one I-frame at the start (IPPP..)
`anchorFrameDistance`	Describes the distance between anchor frames, also known as M, where an Anchor frame is defined as a frame that predictions are made from, for example, an I or P frame (optional).
`difficulty`	Describes the transcoding difficulty of the media in a segment. The difficultyHint takes values from 0.0 to 1.0, where 0.0 indicates the lowest difficulty and 1.0 indicates the highest difficulty. The difficulty is normalized within the contents. The maximum value of the difficulty is 1.0, which indicates the most difficult part to encode within the contents (optional).
`importance`	Describes the importance with respect to the semantic content of the media. The importance takes values from 0.0 to 1.0, where 0.0 indicates the lowest importance and 1.0 indicates the highest importance (optional).
`spatialResolutionHint`	Describes the maximum allowable spatial resolution reduction factor for perceptibility. The SpatialResolutionHint takes values from 0.0 to 1.0, where 0.5 indicates that the resolution can be reduced by half and 1.0 indicates the resolution cannot be reduced (optional).

Content Management features: the MPEG-7 proposal

Media Information

- Media Identification
  - o entity identifier, domain
- Media Profiles

Name	*Definition*
`MediaQualityType`	Describes the media quality of the Media Profile.
`QualityRating`	Describes the rating values and the criterion used to create the media quality ratings.
`type`	Describes the type of media quality rating. The types of quality rating are defined as follows. •*subjective* – The rating is subjective, that is, it is provided by human viewers. •*objective* – The rating is objective, that is, it is acquired using computational means.
`RatingSource`	Describes the source that provides the ratings (optional).
`RatingInformationLocator`	Indicates the locator for additional information about the quality rating method (optional).
`PerceptibleDefects`	Describes defects that are perceived in the media (optional).
`VisualDefects`	Describes the visual errors perceived in the media (optional). Errors are listed in the order of descending severity. An example of CS is `VisualDefectsCS`.
`AudioDefects`	Describes the audio errors perceived in the media (optional). Errors are listed in the order of descending severity. An example of CS is `AudioDefectsCS`.

Content Management features: the MPEG-7 proposal

Media Information

Creation Information

- Creation
- Classification
- RelatedMaterial

Usage Information

Content Management features: the MPEG-7 proposal

Media Information

Creation Information

Name	*Definition*
`CreationType`	Describes the creation of the content, including places, dates, actions, materials, staff (technical and artistic) and organizations involved.
`Title`	Describes one textual title of the multimedia content. Multiple titles are allowed. They may correspond to different types (indicated by the type attribute) or to different languages (indicated by the `xml:lang` attribute).
`TitleMedia`	Describes one multimedia title of the multimedia content (optional). It serves as a multimedia identifier of the multimedia content. It may contain a title in the form of an image, an audio clip, or a video.
`Abstract`	Describes a textual abstract of the multimedia content (optional). It is a summary, assigned during the creation process, of what is conveyed in the multimedia content.
`Creator`	Describes one creator of the multimedia content (optional). It allows the description of persons, organizations, groups, and so forth involved in the creation as well as their role.
`CreationCoordinates`	Describes the location and the date of creation of the multimedia content (optional).
`Location`	Describes the place where the multimedia content was created (optional).
`Date`	Describes the date or period when the multimedia content was created (optional).
`CreationTool`	Describes one device (and its settings) used in the creation of the multimedia content (optional).
`CopyrightString`	Describes one textual label indicating information that may be displayed or otherwise made known to the end user (optional). It is not a formal declaration of the usage rights of the multimedia content.

Content Management features: the MPEG-7 proposal

Name	*Definition*
`ClassificationType`	Describes the classification of the multimedia content.
`Form`	Describes the production type of the document, such as, film, news program, magazine, documentary, etc (optional). An example of CS is `FormatCS`.
`Genre`	Describes what the multimedia content is about (broad classification), such as sports, politics, economics, etc (optional). An example of CS is the `GenreCS`.
`type`	Indicates the type of the genre of the multimedia content. The types of genres are defined as follows. •*main* – The specified genre is the main, or primary. This is the default value. •*secondary* – The specified genre is a secondary genre, such as a subgenre.
`Subject`	Describes the subject (specific classification) of the multimedia content (optional). The subject allows a textual annotation to classify the multimedia content.
`Purpose`	Describes one purpose for which the multimedia content was created (optional). An example of CS is `IntentionCS`.
`Language`	Describes one language of the spoken audio of the program (optional).
`CaptionLanguage`	Describes one language of the caption information included with the program (optional). The type of the caption information associated with the program is denoted by the closed attribute. Closed captions can be turned on or off by the user, while open captions (or subtitles) are part of the picture itself and remain visible.
`SignLanguage`	Specifies the audio sign language provided for the multimedia content, and, optionally, qualifies the use of signing as a primary language or as a translation of the spoken dialogue (optional).
`Release`	Describes the release date and region of the multimedia content (optional).
`Region`	Indicates the countries or regions in which the multimedia content was first released (optional). This locator may be different than the location where it was created.
`date`	Indicates the date on which the multimedia content was first released. This date may be different than the date(s) when it was created (optional).
`Target`	Describes the target of the multimedia content in terms of market classification, age and country or region (optional).
`Market`	Describes one targeted market of the multimedia content (optional). An example of CS is `TargetGroupCS`.
`Age`	Describes the targeted age range of the multimedia content (optional).
`Region`	Describes one target country or region for the multimedia content (optional).
`ParentalGuidance`	Describes one parental guidance classification of the multimedia content (optional).
`MediaReview`	Describes one media review about the multimedia content (optional).

Content Management features: the MPEG-7 proposal

Media Information

C

U

Name	*Definition*
`RelatedMaterialType`	Describes material containing additional information about the multimedia content or related to it (e.g., extended reports of a news program or Web pages with information about topics covered in the news).
`DisseminationFormat`	Describes the publication medium or delivery mechanism of the related material, for example, terrestrial broadcast, web-cast, streaming, CD-ROM, and so forth. An example of CS is `DisseminationFormatCS`.
`MaterialType`	Describes the type of the related material (optional). For example, script of a movie, original book, lyrics of a song, or drafts of a graphic design.
`MediaLocator`	Describes the media location of the related material.
`MediaInformation`	Describes the media information about the related material. This descriptor provides information sufficient for locating the media.
`MediaInformationRef`	References the media information description of the related material. This descriptor provides information sufficient for locating the media.
`CreationInformation`	Describes the creation information about the related material.
`CreationInformationRef`	References the creation information description of the related material.
`UsageInformation`	Describes the usage information about the related material.
`UsageInformationRef`	References the usage information description of the related material.

Content Management features: the MPEG-7 proposal

Media Information

Creation Information

Usage Information

- Rights
- Financial Results

Name	*Definition*
`RightsType`	Describes a link to the right holders and to the access rights information. It gives an ID (such as specified in the MPEG-4 IP Identification Data Set) that enables access to the current Rights Owner information about the multimedia content or the description (making use of appropriate rights management databases).
`RightsID`	Identifies the link to the current Rights information about the content, including Rights Holders, Access Rights, and other related information.

Content Management features: the MPEG-7 proposal

Media Information

Creation Information

Usage Information

- Rights
- Financial Results

Name	*Definition*
`FinancialType`	Describes the costs and incomes generated by the multimedia content.
`AccountItem`	Describes an account item (cost or income).
`EffectiveDate`	Describes when the cost or income is made effective (optional). It can be the same as when it was generated or not.
`CostType`	Describes the type of a cost.
`IncomeType`	Describes the type of an income.
`currency`	Describes the currency of the price (see ISO 4217).
`value`	Describes the value of the price.

Content Management features: the MPEG-7 proposal

Media Information

Name	*Definition*
`AvailabilityType`	Describes the availability of the multimedia content. For example, broadcasting, on demand delivery, CD sales, and so forth.
`InstanceRef`	Describes a reference to the `MediaInstance` for which the availability is described.
`Dissemination`	Describes how the multimedia content is disseminated (optional).
`Financial`	Describes the financial information related to the particular use described in the `Availability` description (optional). For example, it allows the description of the price of using the content and additional costs due to the availability of the content.
`Rights`	Describes information about the owners of the rights corresponding to the multimedia content, and how the multimedia content can be used (optional). Its appearance at this level precludes its appearance in the `Rights` instance of the same `UsageInformation` instance.
`AvailabilityPeriod`	Describes one period (date, time, and duration) of availability of the multimedia content (optional). For example, time and date of the broadcast, or time and date of availability for an on-demand content. In the case of availability on a persistent medium (e.g, DVD), it specifies the date of publication. If there is no duration description, the duration of the availability is unbounded.
`type`	Indicates the types of availability of the period (optional): •*live* – Indicates the multimedia content is made available (e.g, broadcast) live. •*repeat* – Indicates the multimedia content is a repeat of multimedia content that has been made available earlier (e.g, broadcast). •*firstShowing* – Indicates the given availability period is the first showing of the multimedia content. •*lastShowing* – Indicates the given availability period is the final showing of the multimedia content. •*conditionalAccess* – Indicates the access to the multimedia content is restricted by a conditional access mechanism. •*encrypted* – Indicates the multimedia content is encrypted for restricted viewing. •*payPerUse* – Indicates the multimedia content is pay-per-use (e.g. pay-per-view). Other values that are datatype-valid with respect to `mpeg7:termReferenceType` are reserved.

Content Management features: the MPEG-7 proposal

Media Information

Creation Information

Usage Information

- Rights
- Financial Results
- Availability
- Usage Record

Name	*Definition*
`UsageRecordType`	Describes past use of the multimedia content.
`AvailabilityRef`	Describes a reference to the content availability's description.
`Audience`	Specifies the number of users who used the multimedia content in the particular use described in the referenced `Availability` description (optional).
`Financial`	Describes the financial information related to the particular use that is described in the referenced availability description (optional).

Fin Unidad 1-2

Algo menos de las dos hroas (2*50)

Indexing of AV Content

Contents

- Introduction
- Metadata Lifecycle
- Classical features for indexing
  - Textual and basic semantic features
  - Content Management features
- AV features for indexing
  - Spatio-temporal structure features
  - Low-level features
  - Mid-level features
  - High-level features
- Bridging the Semantic Gap
- Collections of AV contents
- Metrics between descriptors and relevance feedback
- Conclusions

Audiovisual Features for Indexing

- Spatio-temporal structure features
- Low-level features
- Mid-level features
- High-level features

Spatio-temporal structure features

Multimedia material can be decomposed at different levels:

- Synchronized component media
  - Can be decomposed at a “high-level” authoring point of view
  - Mostly “presentation” driven
- Static Images
  - Can be decomposed in spatial regions
    - Grid based (regular regions)
    - Content based (homogeneous regions)
- Graphics
  - Can be decomposed in pieces of the object
  - Can be decomposed in “components” (e.g., triangles, splines, …)
    - Point of views
- Audio
  - Temporal segmentation
  - Spatialization segmentation (channels)
- Video
  - Temporal segmentation
  - Spatial segmentation
  - Spatio-temporal segmentation

Spatio-temporal structure

Spatio-temporal structure (Segments)

- *Types:*
  - *Still Region,*
  - *Video Segment,*
  - *Moving Region,*
  - *Audio Segment*
- *Textual annotations (elementary semantics), Content Management, ...*
- *Relations among segments*
- *Hooks for Audio and Visual description tools (depending on segment type)*

```
<Mpeg7 …>
<DescritpionMetadata>…</DescriptionMetadata>
<Description xsi:type="ContentEntity">
  <MultimediaContent xsi:type=”VideoType”>
    <Video id=”video_example”>
    <MediaInformation>...</MediaInformation>
    <TemporalDecomposition gap=”false” overlap=”false”>
      <VideoSegment id=”VS1”>
        <MediaTime>
          <MediaTimePoint>T00:00:00</MediaTimePoint>
          <MediaDuration>PT2M</MediaDuration>
        </MediaTime>
        <VisualDescriptor
          xsi:type=”GoFGoPColorType”  aggregation=”average”>
          <ScalableColor
            numOfCoef=”8” numOfBitplanesDicarded=”0”>
            <Coeff>1 2 3 4 5 6 7 8</Coeff>
          </ScalableColor>
        </VisualDescriptor>
      </VideoSegment>
      <VideoSegment id=”VS2”>
        <MediaTime>
          <MediaTimePoint>T00:02:00</MediaTimePoint>
          <MediaDuration>PT2M</MediaDuration>
        </MediaTime>
        <VisualDescriptor
          xsi:type=”GoFGoPColorType”  aggregation=”average”>
          <ScalableColor
            numOfCoef=”8” numOfBitplanesDicarded=”0”>
            <Coeff>8 7 6 5 4 3 2 1</Coeff>
          </ScalableColor>
        </VisualDescriptor>
      </VideoSegment>
    </TemporalDecompostion>
    </Video>
  </MultimediaContent>
</Description>
</Mpeg7>
```

Image description: Still Regions and Spatial Relations

No gap, no overlap

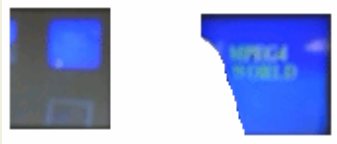
SR1:
- Media, Creation, & Usage information
- Spoken Content
- Textual annotation
- Color histogram, Texture

SR7:
- Color Histogram
- Textual annotation

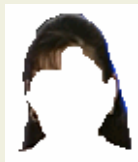
Gap, no overlap

No gap, no overlap

SR2:
- Shape
- Color Histogram
- Textual annotation



Gap, no overlap



SR5:
- Shape
- Color Histogram
- Textual annotation

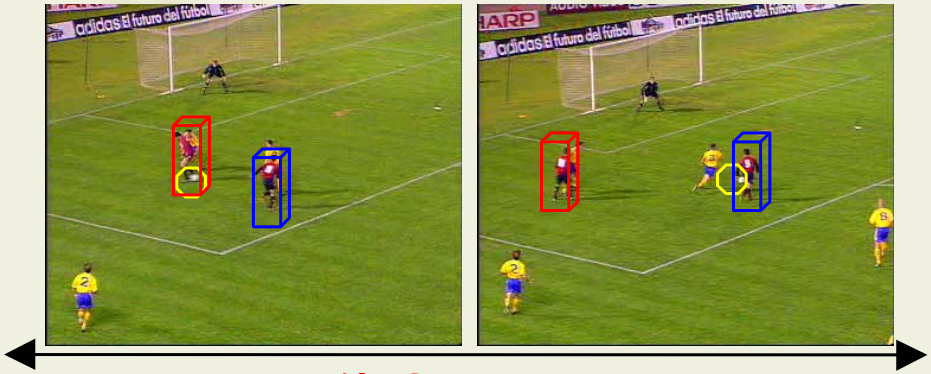
SR8:
- Color Histogram
- Textual annotation

SR3:
- Shape
- Color Histogram
- Textual annotation

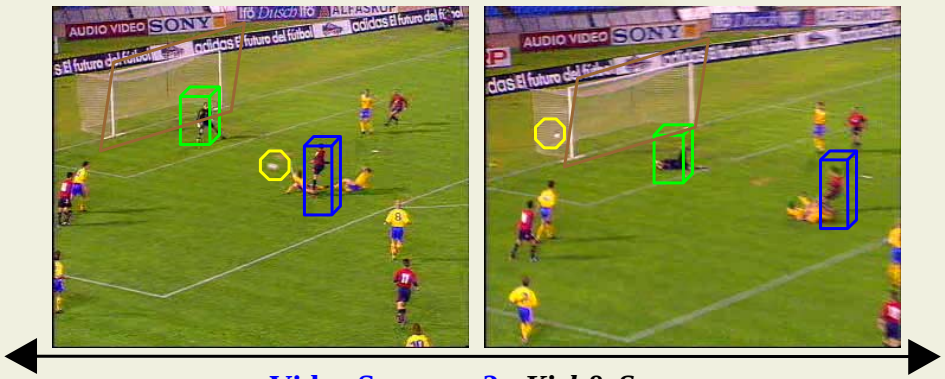
SR6:
- Shape
- Textual annotation

SR4:
- Shape
- Color Histogram
- Textual annotation

Video Description: Regions, Segments and Relations



Video Segment 1: ***Pass***



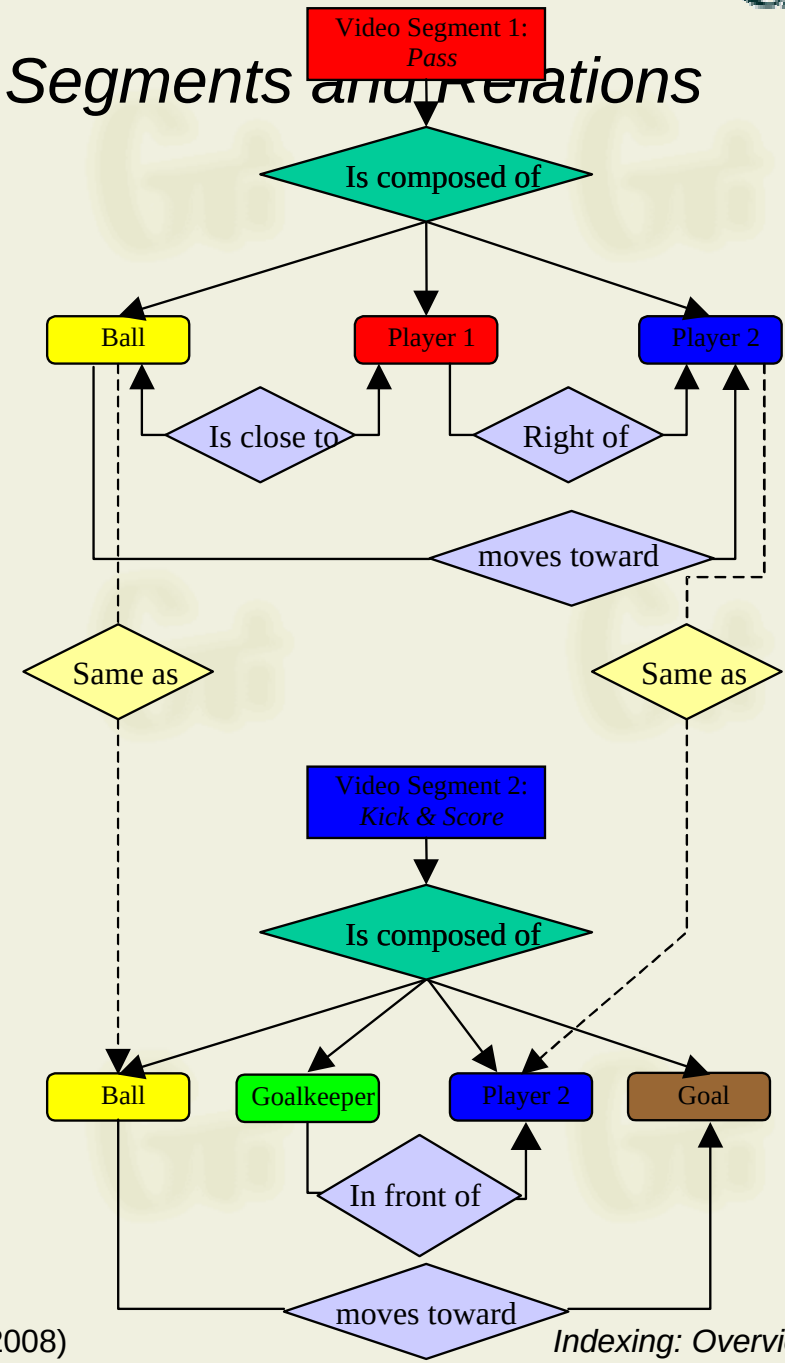
Video Segment 2: ***Kick& Score***

Moving Region: Players

Moving Region: Ball

Moving Region: Goalkeeper

Still Region: Goal



Spatio-temporal structure features: the MPEG-7 proposal

MPEG-7 is the only standard supporting description tools for spatio-temporal structure beyond the “Synchronized component media” (could be supported by many approaches) and “temporal segmentation” (SMPTE) features

Nevertheless it lacks support for graphics and has limited support for audio “spatialization” segmentation.

- Somehow supported by graphic and audio representation formats

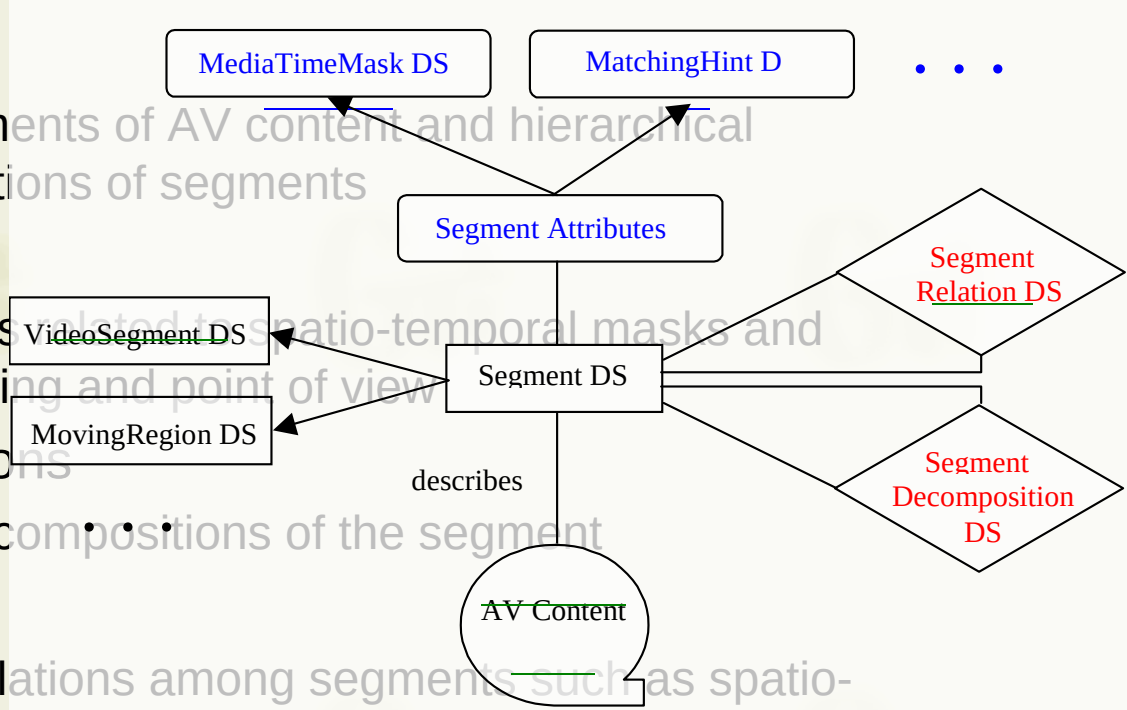
The main related description tools are:

- Segment entities
- Segment Attributes
- Segment decomposition
- Structural relation

Spatio-temporal structure features: the MPEG-7 proposal

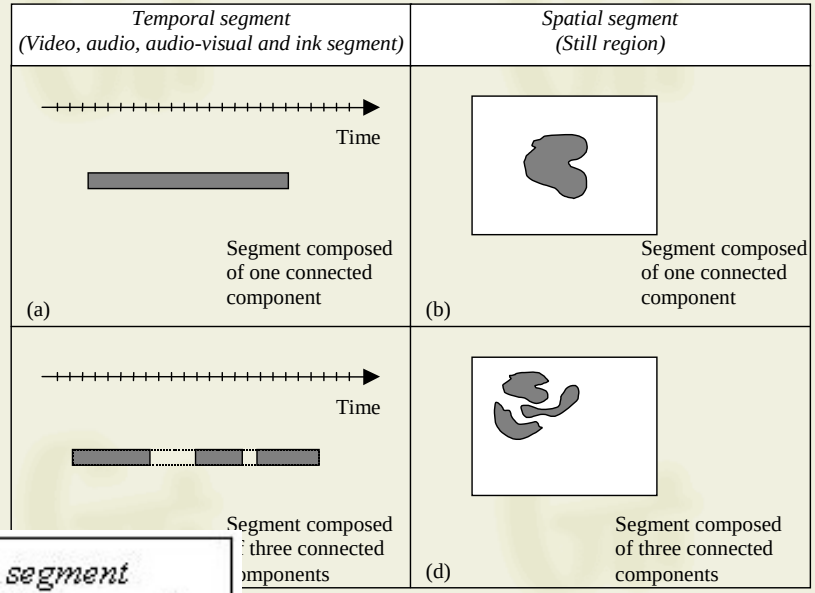
Structure of Content

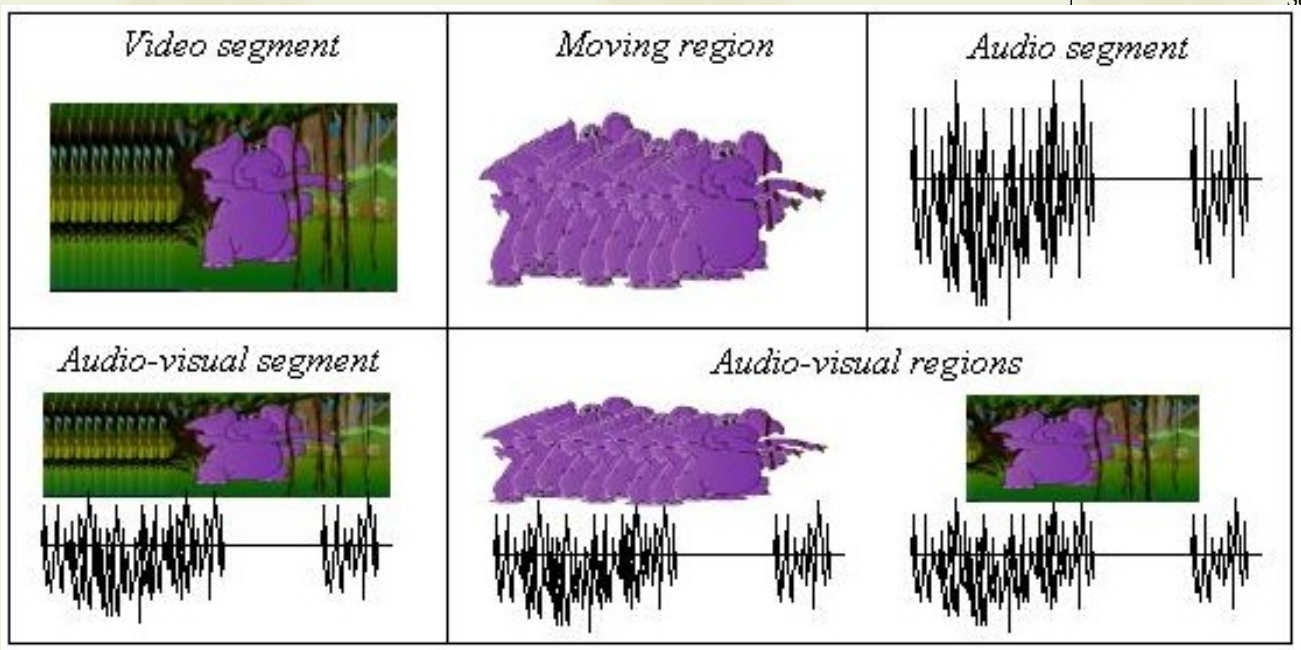
- Segment Entities
  - spatio-temporal segments of AV content and hierarchical structural decompositions of segments
- Segment Attributes
  - attributes of segments related to spatio-temporal masks and importance for matching and point of view
- Segment Decompositions
  - Describe different decompositions of the segment
- Segment Relations
  - describe structural relations among segments such as spatio-temporal relations



Spatio-temporal structure features: the MPEG-7 proposal

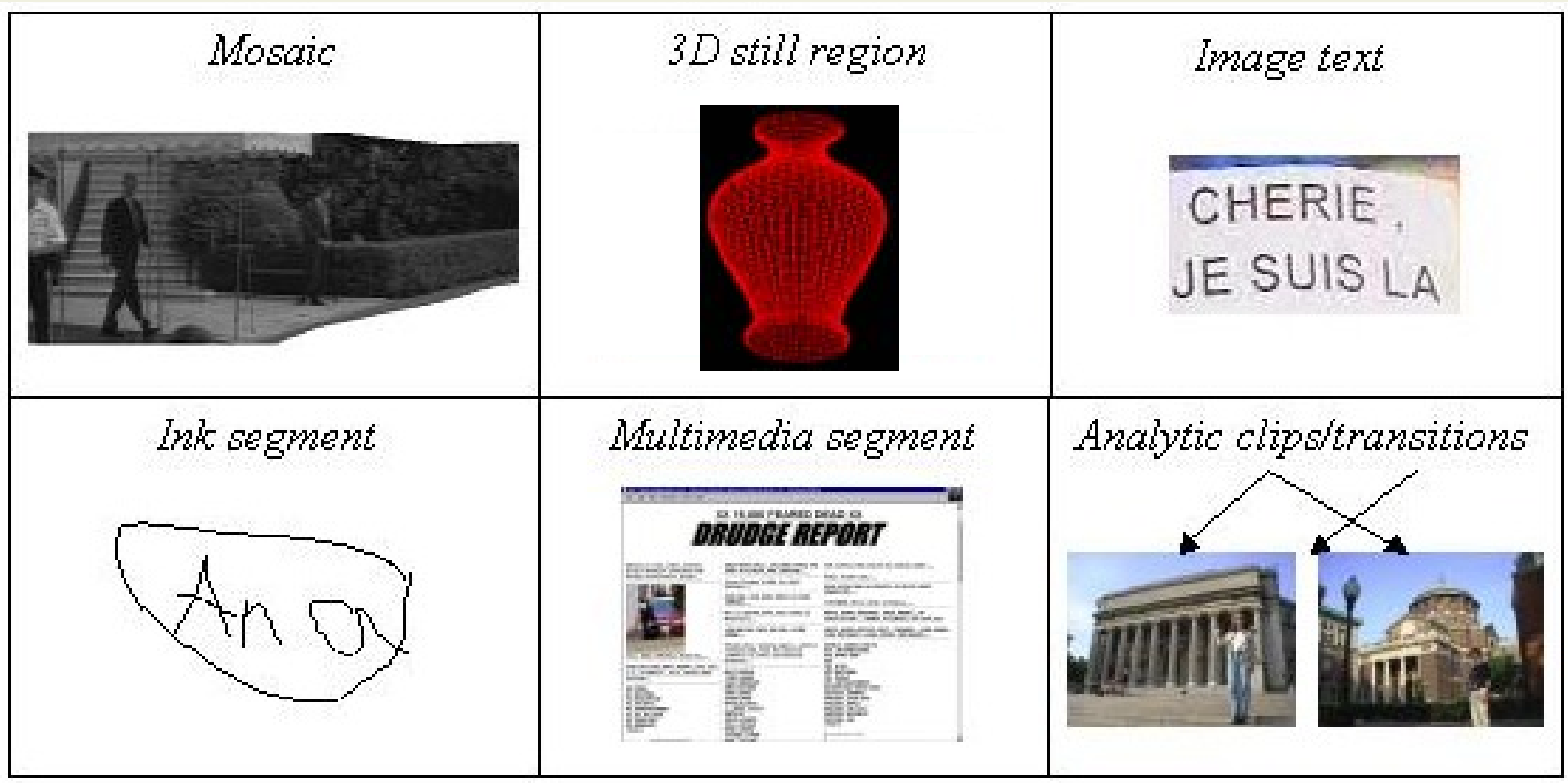
Generic Segment entities





Spatio-temporal structure features: the MPEG-7 proposal

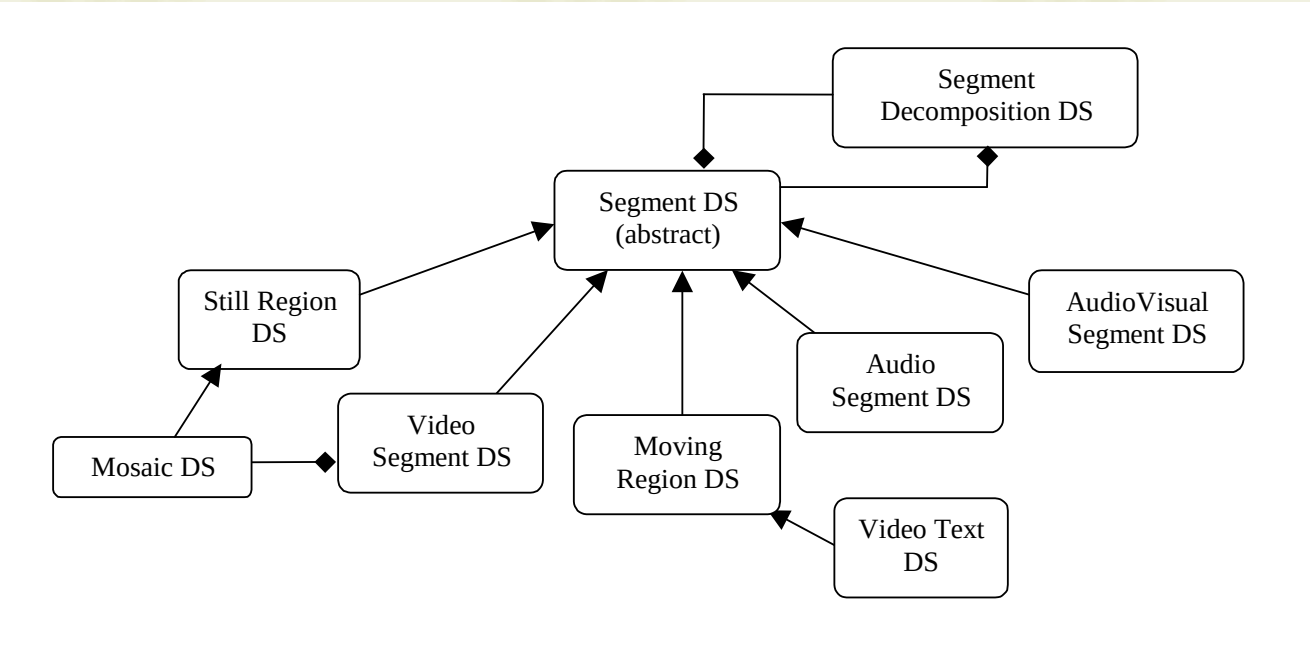
Specific Segment entities



Spatio-temporal structure features: the MPEG-7 proposal

Segment DS

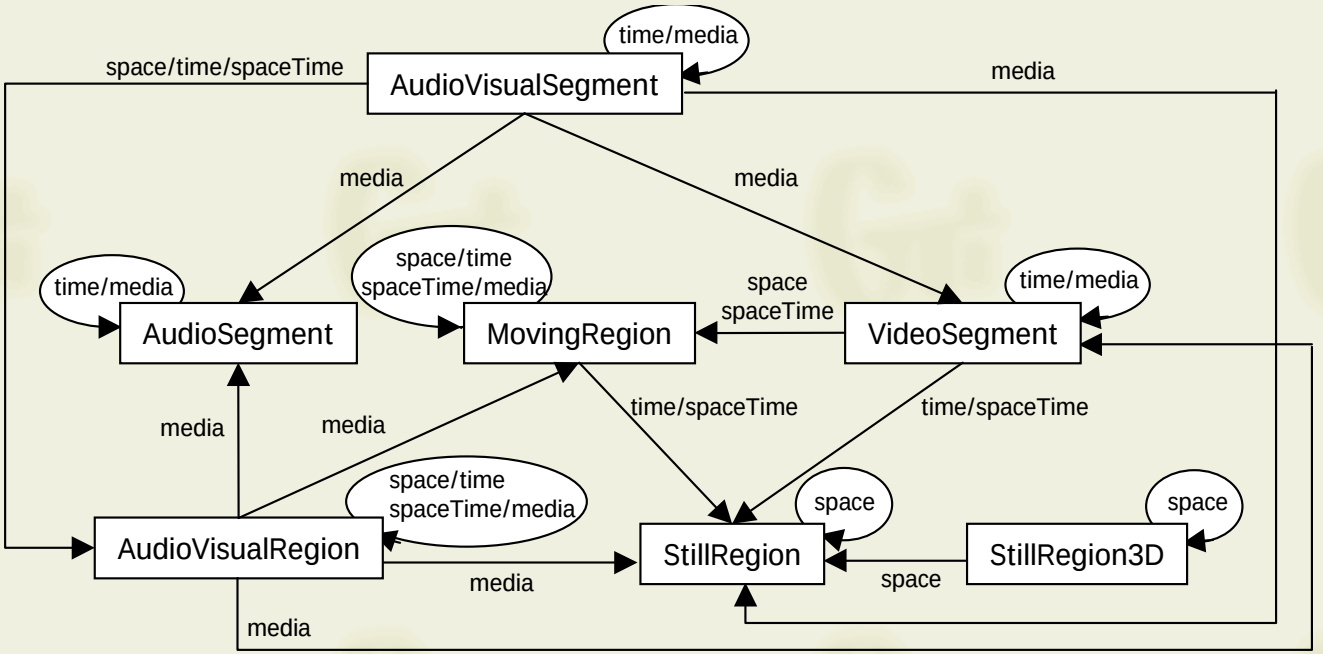
- MediaInfo
- MediaLocator
- CreationInfo
- UsageInfo
- TextAnnotation
- MatchingHint
- PointOfView
- Mask
- SegmentDecomposition
- Relation
- Audio-visual hooks (low-level features)



Spatio-temporal structure features: the MPEG-7 proposal

Segment decomposition

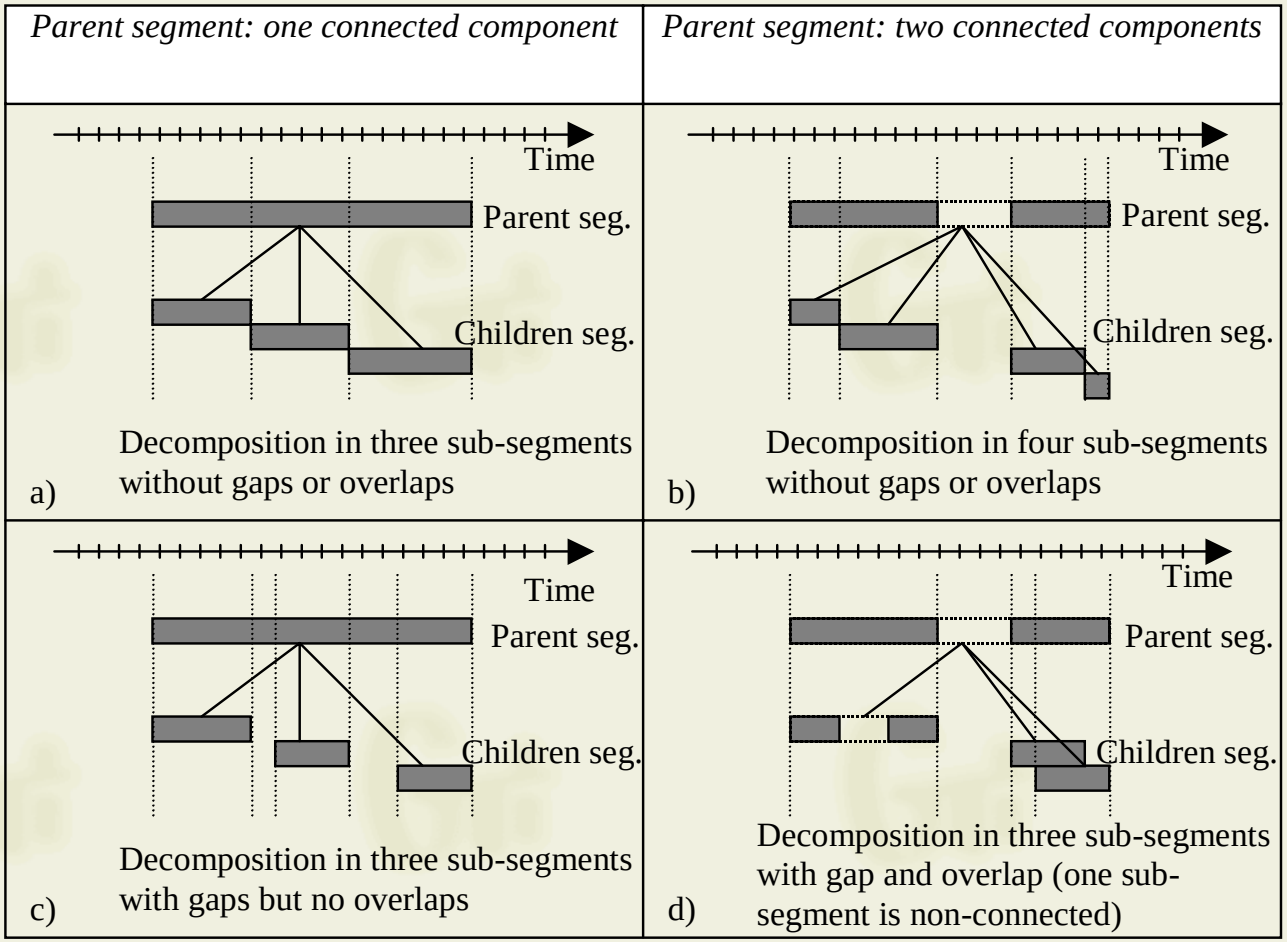
- Hierarchical structure (support for annotation of ToC, indexes, …)
- Time, space, spacetime, media sources decomposition



Cannot be the result of any segment decomposition

Spatio-temporal structure features: the MPEG-7 proposal

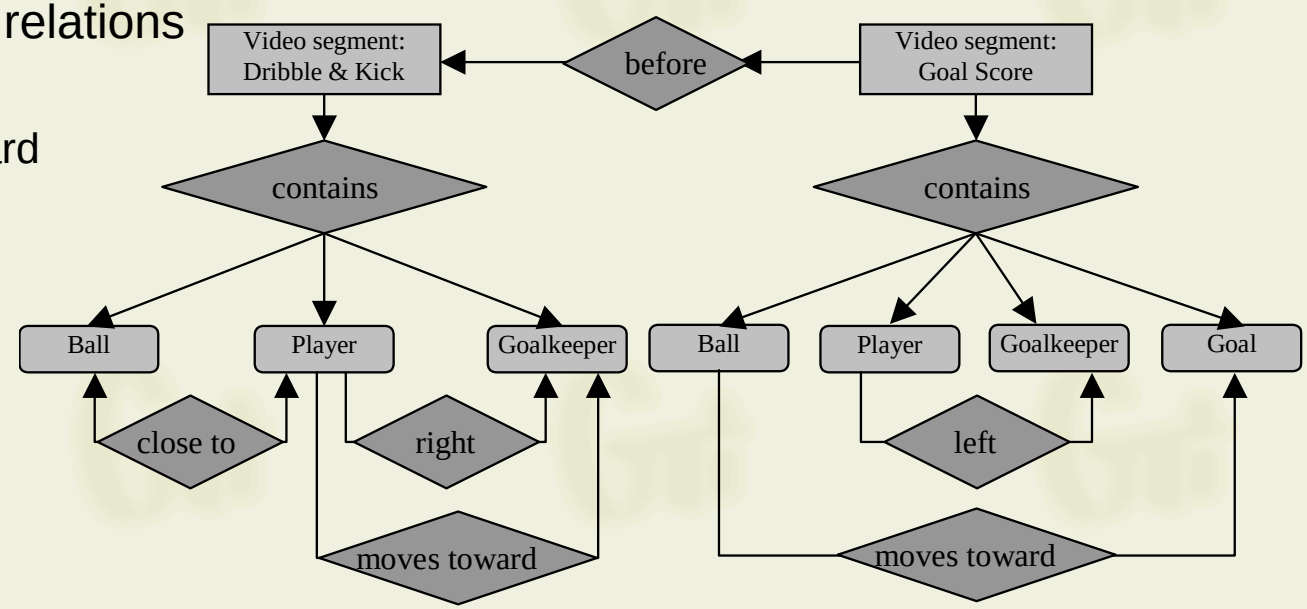
Segment decomposition



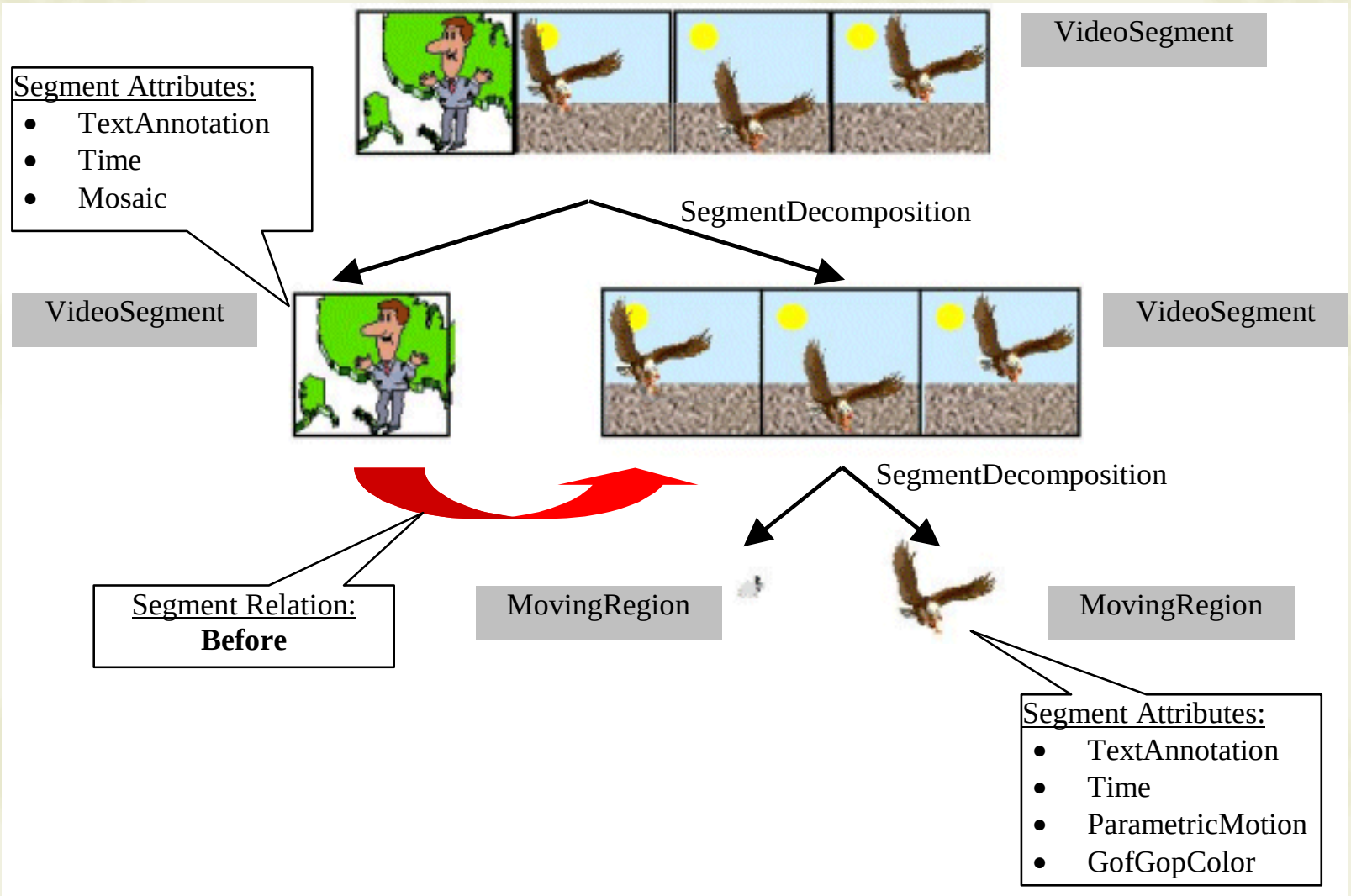
Spatio-temporal structure features: the MPEG-7 proposal

Segment decomposition

- Normative relations (Classification Scheme)
  - Spatial: south, north, west, east, northwest, northeast, southwest, southeast, left, right, below, above, over, under
  - Temporal: before, after, meets, metBy, overlaps, overlappedBy, during, contains, strictDuring, strictContains, starts, startedBy, finishes, finishedBy, equal, contiguous, sequential, cobegin, coned, parallel, overlapping
  - Generic relations: e.g., identity, union, disjoint
- Non-normative relations
  - Close to
  - Moves forward



Spatio-temporal structure features: the MPEG-7 proposal



Audiovisual Features for Indexing

- Spatio-temporal structure features
- Low-level features
- Mid-level features
- High-level features

Low-level features

Low-level features refers commonly to audio and visual features

Some structural features can be seen as a low-level feature, but they are differentiated from audio and visual features because these are “extracted” from segments…

- The first step in any indexing is (must be) the segmentation

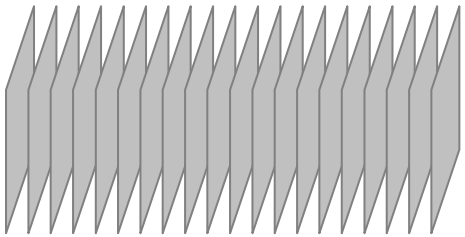
… and they are “hooked” to segments

- Depending on the segment type

Low-level features

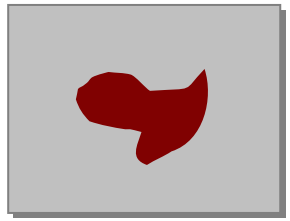
Low-level features refers commonly to audio and visual features

Video segments



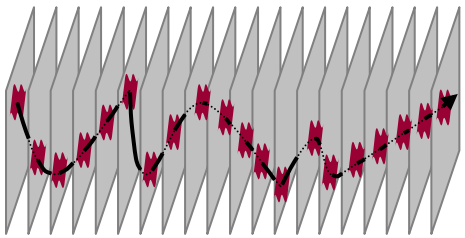
- Color
- Camera motion
- Motion activity
- Mosaic

Still regions



- Color
- Shape
- Position
- Texture

Moving regions



- Color
- Motion trajectory
- Parametric motion
- Spatio-temporal shape

Audio segments

- Spoken content
- Spectral characterization
- Music: timbre, melody

Low-level features: the MPEG-7 proposal

Several features are proposed/used

- Lot of technology around

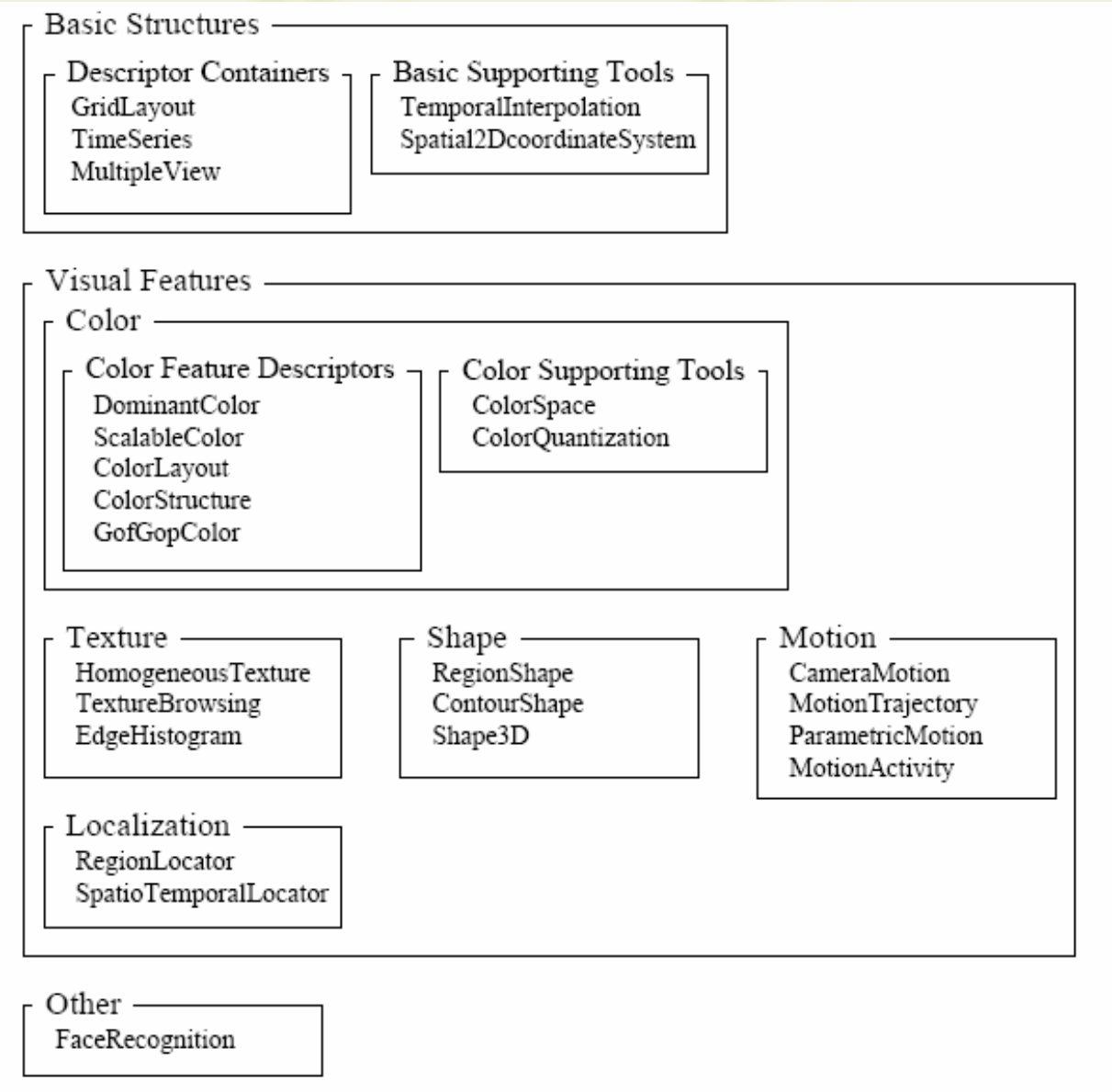
but only one standard

- MPEG-7

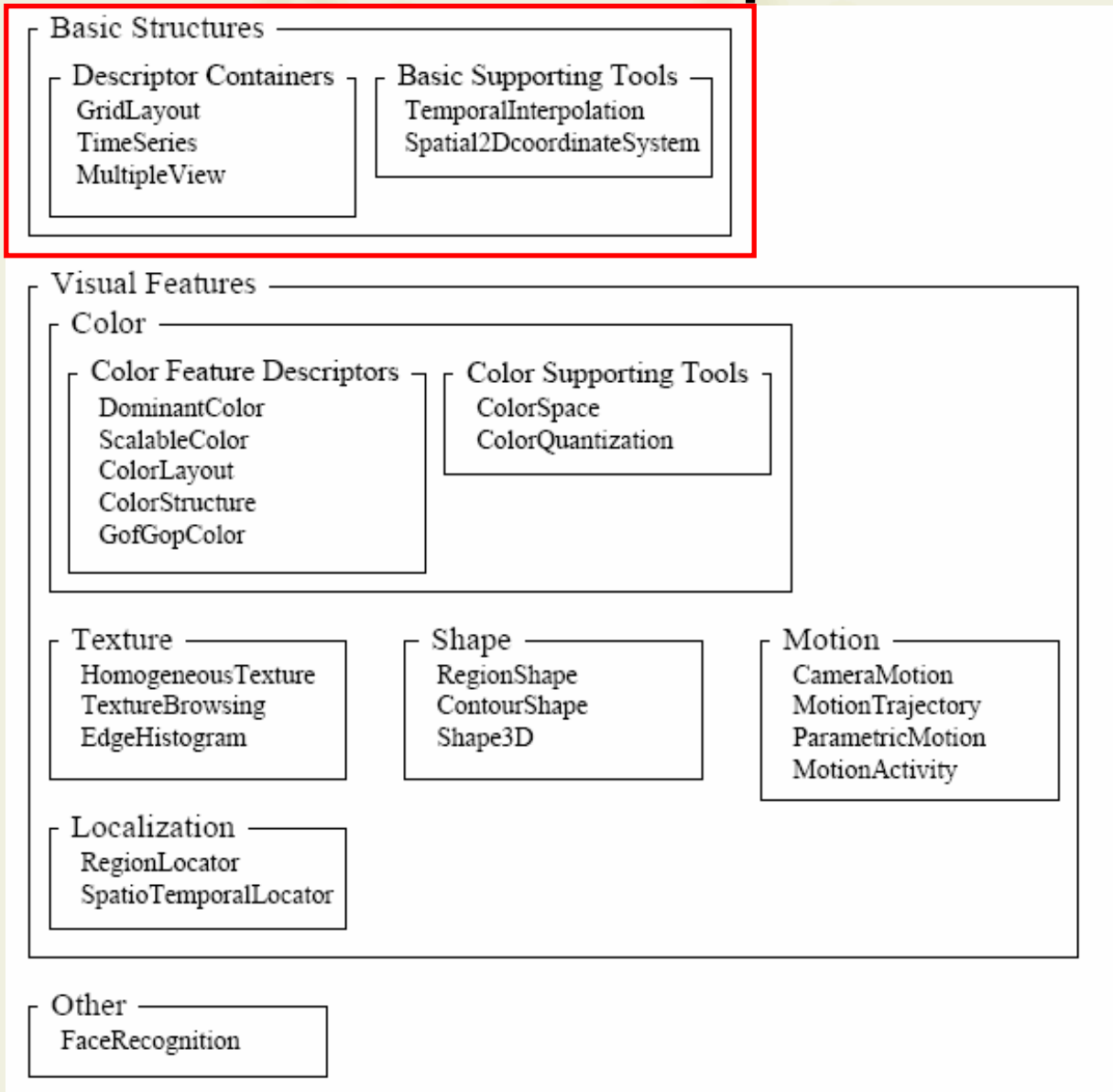
MPEG-7 Visual and MPEG-7 Audio Description tools

- No support (currently) for synthetic content

MPEG-7 Visual Description tools



MPEG-7 Visual Description tools



MPEG-7 Visual Description tools
Basic Structures – Grid Layout (I)

Grid Layout: splitting of the image in equally sized rectangular regions where other visual descriptors (of the same type) are attached

- Not all visual descriptors are allowed (e.g., ColorSpace, motion ones)
- If applied to video, all frames have the same grid

```
<GridLayout numOfPartX="2" numOfpartY="2" descriptorMask="0110">
    <!--instance at (0 1) -->
    <Descriptor xsi:type="ColorLayoutType">
        <YDCCoeff>50</YDCCoeff>
        <CbDCCoeff>34</CbDCCoeff>
        <CrDCCoeff>30</CrDCCoeff>
        <YACCoeff5>16 12 15 12 17</YACCoeff5>
        <CbACCoeff2>12 17</CbACCoeff2>
        <CrACCoeff2>12 14</CrACCoeff2>
    </Descriptor>
    <!--instance at (1 0) -->
    <Descriptor xsi:type="ColorLayoutType">
        <YDCCoeff>48</YDCCoeff>
        <CbDCCoeff>34</CbDCCoeff>
        <CrDCCoeff>32</CrDCCoeff>
        <YACCoeff5>12 10 13 9 10</YACCoeff5>
        <CbACCoeff2>14 15</CbACCoeff2>
        <CrACCoeff2>16 12</CrACCoeff2>
    </Descriptor>
</GridLayout>
```

MPEG-7 Visual Description tools
Basic Structures – Grid Layout (II)

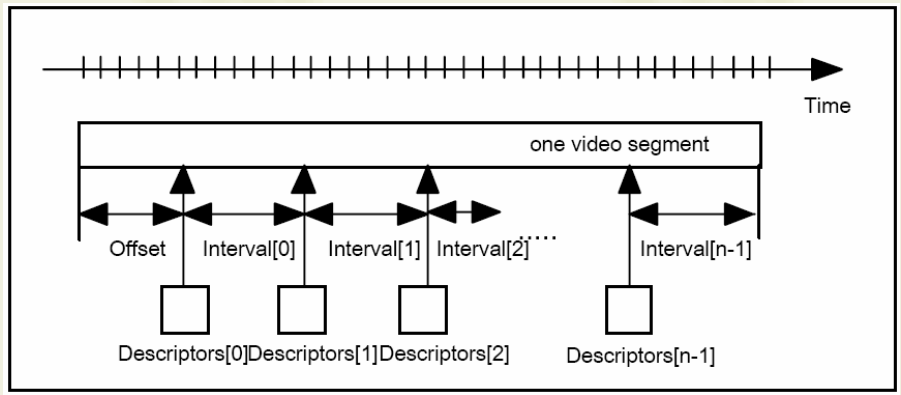
Grid Layout: Binary representation

GridLayout {	Number of bits	Mnemonic
DescriptorID	8	uimsbf
numOfPartX	8	uimsbf
numOfPartY	8	uimsbf
DescriptorMaskPresent	1	bslbf
if(DescriptorMaskPresent) {		
descriptorMask	partNumX*partNumY	bslbf
}		
for(k=0;k<partNumX*partNumY; k++) {		
if(DescriptorMaskPresent) {		
if(descriptorMask[k]) {		
Descriptor[k]		Descriptor instance specified by descriptorID
}		
} else {		
Descriptor[k]		Descriptor instance specified by descriptorID
}		
}		
}		

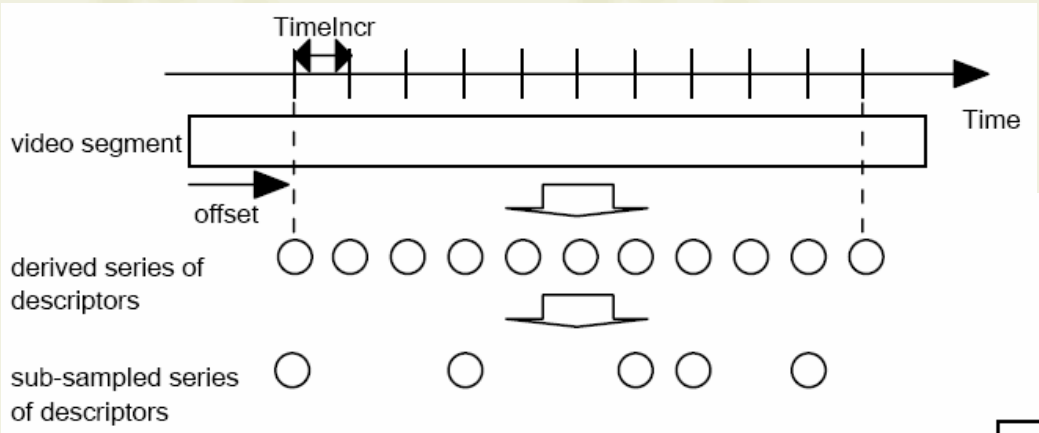
MPEG-7 Visual Description tools
Basic Structures – Time Series (I)

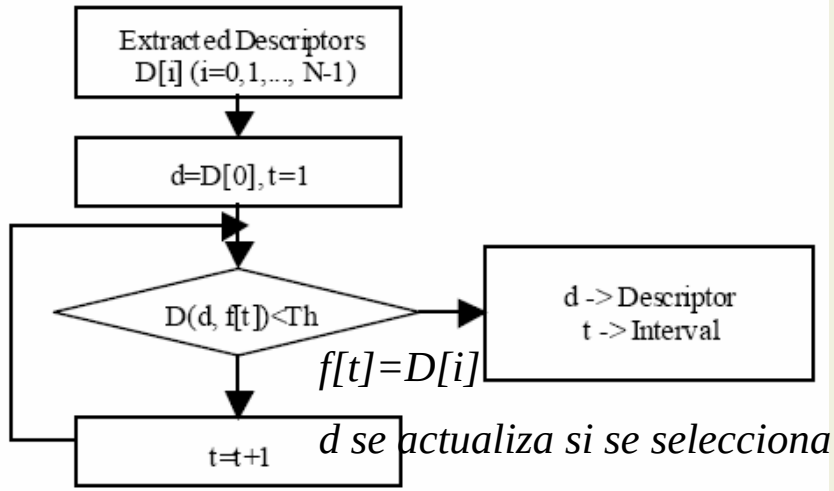
Time Series: regular or irregular series of Ds in a video segment

- VisualTimeSeries (base type/abstract element)
- RegularVisualTimeSeries
- IrregularVisualTimeSeries



- Extraction





- Similarity matching
  - MPEG-7 part 8 specifies the similarity matching rules, including when sequence are of different length

MPEG-7 Visual Description tools
Basic Structures – Time Series (II)

RegularVisualTimeSeries {	Number of bits	Mnemonic
DescriptorID	8	bslbf
NumOfDescriptors	32	uimsbf
IsRandomAccess	1	bslbf
if(IsRandomAccess) {		
DescriptorLength	16	uimsbf
}		
TimeIncr	See annex B	
IsOffset	1	
if(IsOffset) {		
offset	See annex B	
}		
if(IsRandomAccess) {		
BitStuffing	0-7	
}		
for(k=0; k<NumOfDescriptors; k++) {		
Descriptor[k]		
if(IsRandomAccess) {		
BitStuffing	0-8*DescriptorLength-1	
}		
}		
}		

IrregularVisualTimeSeries {	Number of bits	Mnemonic
DescriptorID	8	bslbf
NumOfDescriptors	32	uimsbf
IsRandomAccess	1	bslbf
if(IsRandomAccess) {		
DescriptorLength	16	uimsbf
}		
TimeIncr	See annex B	mediaDurationType
IsOffset	1	bslbf
if(IsOffset) {		
offset	See annex B	mediaDurationType
}		
IsShortInterval	1	bslbf
if(IsRandomAccess) {		
BitStuffing	0-7	vlclbf
}		
for(i=0; i<NumOfDescriptors; i++) {		
Descriptor[i]		descriptor instance specified by DescriptorID
if(IsRandomAccess) {		
BitStuffing	0-8*DescriptorLength-1	vlclbf
}		
if (IsShortInterval) {		
ShortInterval[i]	8	uimsbf
} else {		
LongInterval[i]	32	uimsbf
}		
}		
}		

MPEG-7 Visual Description tools
Basic Structures – Time Series (III)

```
<RegularVisualTimeSeries offset="PT1N10F">
   <TimeIncr>PT3N10F</TimeIncr>
   <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>50</YDCCoeff>
      <CbDCCoeff>34</CbDCCoeff>
      <CrDCCoeff>30</CrDCCoeff>
      <YACCoeff5>16 12 15 12 17</YACCoeff5>
      <CbACCoeff2>12 17</CbACCoeff2>
      <CrACCoeff2>12 14</CrACCoeff2>
   </Descriptor>
   <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>48</YDCCoeff>
      <CbDCCoeff>34</CbDCCoeff>
      <CrDCCoeff>32</CrDCCoeff>
      <YACCoeff5>12 10 13 9 10</YACCoeff5>
      <CbACCoeff2>14 15</CbACCoeff2>
      <CrACCoeff2>16 12</CrACCoeff2>
   </Descriptor>
</RegularTimeSeries>
```

```
<IrregularVisualTimeSeries>
   <TimeIncr>PT1N10F</TimeIncr>
   <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>50</YDCCoeff>
      <CbDCCoeff>34</CbDCCoeff>
      <CrDCCoeff>30</CrDCCoeff>
      <YACCoeff5>16 12 15 12 17</YACCoeff5>
      <CbACCoeff2>12 17</CbACCoeff2>
      <CrACCoeff2>12 14</CrACCoeff2>
   </Descriptor>
   <Interval>4</Interval>
   <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>46</YDCCoeff>
      <CbDCCoeff>34</CbDCCoeff>
      <CrDCCoeff>30</CrDCCoeff>
      <YACCoeff5>16 12 15 12 17</YACCoeff5>
      <CbACCoeff2>12 17</CbACCoeff2>
      <CrACCoeff2>12 14</CrACCoeff2>
   </Descriptor>
   <Interval>1</Interval>
   <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>40</YDCCoeff>
      <CbDCCoeff>34</CbDCCoeff>
      <CrDCCoeff>30</CrDCCoeff>
      <YACCoeff5>16 12 15 12 17</YACCoeff5>
      <CbACCoeff2>12 17</CbACCoeff2>
      <CrACCoeff2>12 14</CrACCoeff2>
   </Descriptor>
   <Interval>3</Interval>
   <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>48</YDCCoeff>
      <CbDCCoeff>34</CbDCCoeff>
      <CrDCCoeff>32</CrDCCoeff>
      <YACCoeff5>12 10 13 9 10</YACCoeff5>
      <CbACCoeff2>14 15</CbACCoeff2>
      <CrACCoeff2>16 12</CrACCoeff2>
   </Descriptor>
   <Interval>1</Interval>
   <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>48</YDCCoeff>
      <CbDCCoeff>34</CbDCCoeff>
      <CrDCCoeff>32</CrDCCoeff>
      <YACCoeff5>15 11 13 9 8</YACCoeff5>
      <CbACCoeff2>14 15</CbACCoeff2>
      <CrACCoeff2>16 12</CrACCoeff2>
   </Descriptor>
   <Interval>1</Interval>
</IrregularVisualTimeSeries>
```

MPEG-7 Visual Description tools
Basic Structures – Multiple View (I)

Multiple View: a tool for representing a 3D object via 2D Ds extracted from different view angles of the 3D objects.

- Different views are needed (3,7,16)
  - MPEG-7 part 8 specifies the way of obtaining the different viewing directions

    If fixedViewsFlag true (else arbitrary views)
  - There is a flag for indicating if the view is visible in the actual content

    Allowing to retrieve content with the same view or any

```
<MultipleView fixedViewsFlag="true">
   <IsVisible>true</IsVisible>
   <Descriptor xsi:type="mpeg7:Contour3hape">...</Descriptor>
   <IsVisible>false</IsVisible>
   <Descriptor xsi:type="mpeg7:Contour3hape">...</Descriptor>
   <IsVisible>false</IsVisible>
   <Descriptor xsi:type="mpeg7:Contour3hape">...</Descriptor>
</MultipleView>
```

MPEG-7 Visual Description tools
Basic Structures – Multiple View (II)

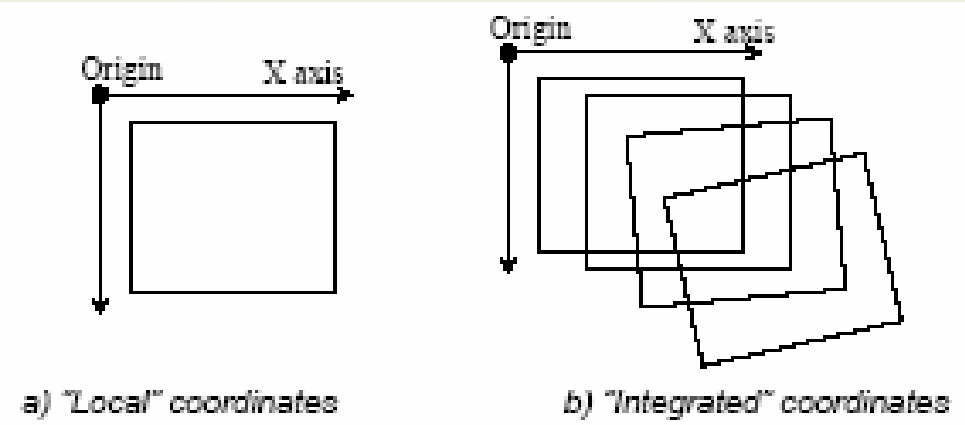
Multiple View binary representation

MultipleView{	Number of bits	Mnemonic
DescriptorID	8	uimsbf
fixedViewsFlag	1	bslbf
NumOfViews	4	uimsbf
for(k=0;k<NumOfViews;k++) {		
IsVisible[k]	1	bslbf
Descriptor[k]		Description instance specified by DescriptorID
}		
}		

MPEG-7 Visual Description tools
Basic Structures – Spatial 2D coordinates (I)

Spatial 2D coordinates: a 2D spatial coordinates system allowing to refer a descriptor extracted in an original image to an adapted image or a set of frames (the original image being the first frame). Allows to map the descriptor each time it is needed without the need of recalculating it and storing it

- Local coordinates (image scaling)
- Integrated coordinates (mosaics, video)



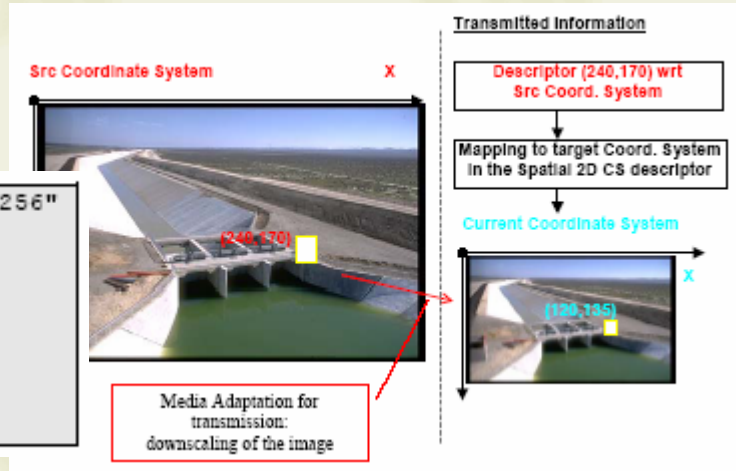
(Confusing)

MPEG-7 Visual Description tools
Basic Structures – Spatial 2D coordinates (II)

Local coordinates (image scaling)

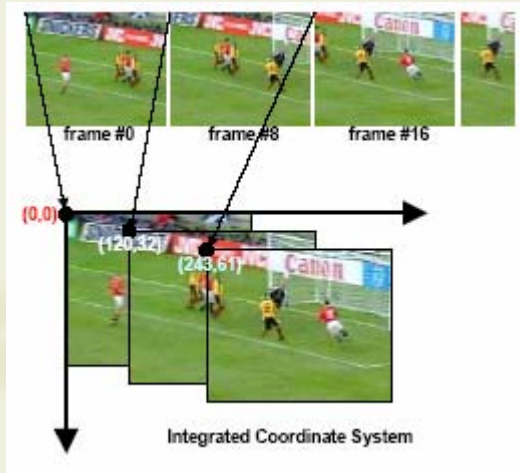
- (incorrect)

```
<Spatial2DCoordinateSystem " xRepr="16" yRepr="16" xSrcSize="384" ySrcSize="256"
id="SCSNewImg1">
   <LocalCoordinateSystem name="PixelCoords">
      <CurrentPixel>0 0</CurrentPixel>
      <SrcPixel>0 0</SrcPixel>
      <CurrentPixel>150 100</CurrentPixel>
      <SrcPixel>300 200</SrcPixel>
   </LocalCoordinateSystem>
</Spatial2DCoordinateSystem>
```



Integrated coordinates (mosaics, video)

```
<Spatial2DCoordinateSystem xRepr="16" yRepr="16">
   <!-- LocalCoordinateSystem is omitted -->
   <!-- (default local coordinate sytem is used) -->
   <IntegratedCoordinateSystem modelType="translational"
         xOrigin="0.0" yOrigin="0.0">
      <!-- description of the frame #8 -->
      <TimeIncr> ... </TimeIncr> <!-- see MediaIncrDuration -->
      <MotionParams>120.0</MotionParams>
      <MotionParams>32.0</MotionParams>
      <!-- description of the frame #16 -->
      <TimeIncr> ... </TimeIncr> <!-- see MediaIncrDuration -->
      <MotionParams>243.0</MotionParams>
      <MotionParams>61.0</MotionParams>
      <!-- description of other frames follows -->
      ...
   </IntegratedCoordinateSystem>
</Spatial2DCoordinateSystem>
```



MPEG-7 Visual Description tools
Basic Structures – Spatial 2D coordinates (III)

Spatial2DCoordinateSystem {	Number of bits	Mnemonic
id	See ISO 10646	UTF-8
xRepr	8	uimsbf
yRepr	8	
XSrcSizeDefined	1	
if (XSrcSizeDefined) {		
xSrcSize		
}		
YSrcSizeDefined	1	
if (YSrcSizeDefined) {		
ySrcSize		
}		
UnitDefined	1	
LocalCoordinatesDefined	1	
IntegratedCoordinatesDefined	1	
if (UnitDefined) {		
Unit	3	
}		

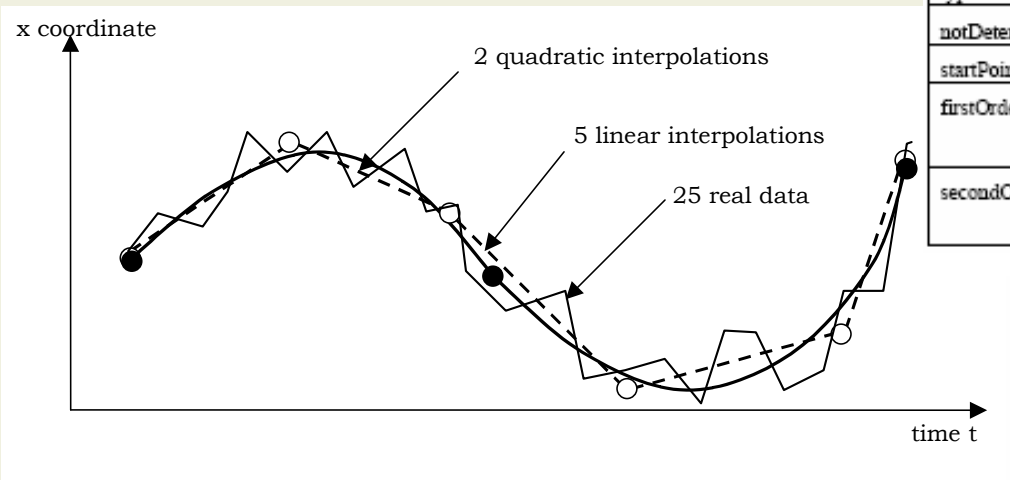
if(LocalCoordinatesDefined) {		
NameLength		vluimsbf5
name	8*NameLength	bslbf
DataSetDefined	1	bslbf
if(DataSetDefined) {		
DataSetLength		vluimsbf5
dataSet	8*DataSetLength	bslbf
}		
Coord	1	bslbf
NumOfPoints	2	uimsbf
for(k=0; k<NumOfPoints; k++) {		
if(!Coord) {		
CurrPixelX	xRepr	usimsbf
CurrPixelY	yRepr	usimsbf
SrcPixelX	16	simsbf
SrcPixelY	16	simsbf
} else {		
PixelX	xRepr	usimsbf
PixelY	yRepr	usimsbf
CoordPointX	32	fsbf
CoordPointY	32	fsbf
}		
}		
NumOfMappingFuncts	2	uimsbf
for(l=0; l<NumOfMappingFuncts; l++) {		
MappingFunctLength		
MappingFunct		

}		
}		
if(IntegratedCoordinatesDefined) {		
modelType	3	uimsbf
xOrigin	32	fsbf
yOrigin	32	fsbf
NumOfMotionParamSets	16	uimsbf
for(k=0; k<NumOfMotionParamSets; k++) {		
TimeIncr	See annex B	MediaIncrDurationType
for(l=0; l<NumOfParams; l++) {		
MotionParams	32	fsbf
}		
}		
}		
}		

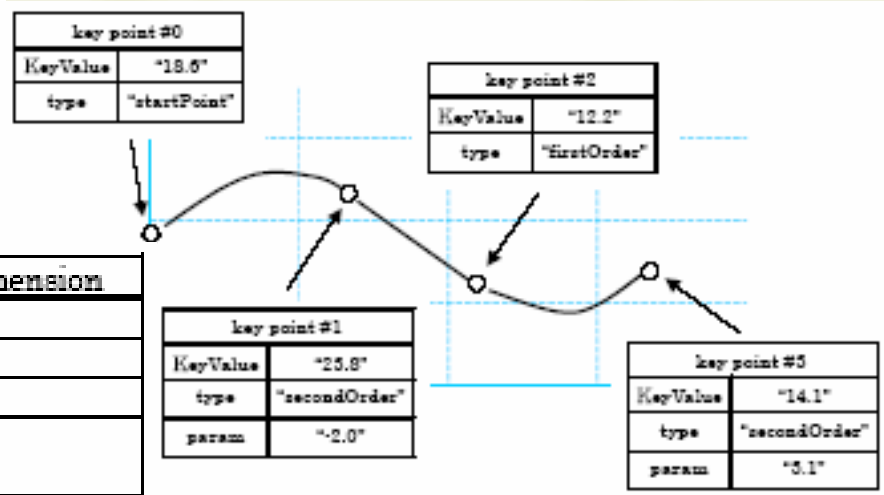
MPEG-7 Visual Description tools
Basic Structures – Temporal Interpolation (I)

Temporal Interpolation: approximation of variable values of a D using connected polynomials

- Reduces the number of required D values



type	Interpolation Function Form	param	Constraint
notDetermined	(none)	(none)	(not applicable)
startPoint	(none)	(none)	(not applicable)
firstOrder	$f(t) = f_a + c_1(t - t_a)$	(none)	$c_1 = \frac{f_b - f_a}{t_b - t_a}$
secondOrder	$f(t) = f_a + c_1(t - t_a) + c_2(t - t_a)^2$	c_2	$c_1 = \frac{f_b - f_a}{t_b - t_a} - c_2(t_b - t_a)$



D or DS using TemporalInterpolation	Value of Dimension
2D MotionTrajectory	2
3D MotionTrajectory	3
ParameterTrajectory (Translational Model)	2
ParameterTrajectory (Affine Transformation Model)	6
ParameterTrajectory (Parabolic Model)	12

MPEG-7 Visual Description tools
Basic Structures – Temporal Interpolation (II)

```
<TwoDimMotionTrajectory>
    <!-- time of 4 key points -->
    <KeyTimePoint>
        <MediaTimePoint>T00:00:00:0F10</MediaTimePoint>
        <MediaTimePoint>T00:00:02:0F10</MediaTimePoint>
        <MediaTimePoint>T00:00:10:5F10</MediaTimePoint>
        <MediaTimePoint>T00:00:15:00F10</MediaTimePoint>
    </KeyTimePoint>

    <!-- X values of 4 key points -->
    <InterpolationFunctions>
        <KeyValue type="startPoint>118.9</KeyValue>
        <KeyValue type="secondOrder" param="2.1">102.1</KeyValue>
        <KeyValue type="firstOrder">82.35</KeyValue>
        <KeyValue type="secondOrder" param="0.2">85.5</KeyValue>
    </InterpolationFunctions>
    <!-- Y values of 4 key points -->
    <InterpolationFunctions>
        <KeyValue>210.0</KeyValue>
        <KeyValue>220.8</KeyValue>
        <KeyValue>228.9</KeyValue>
        <KeyValue>215.1</KeyValue>
    </InterpolationFunctions>
</TwoDimMotionTrajectory>
```

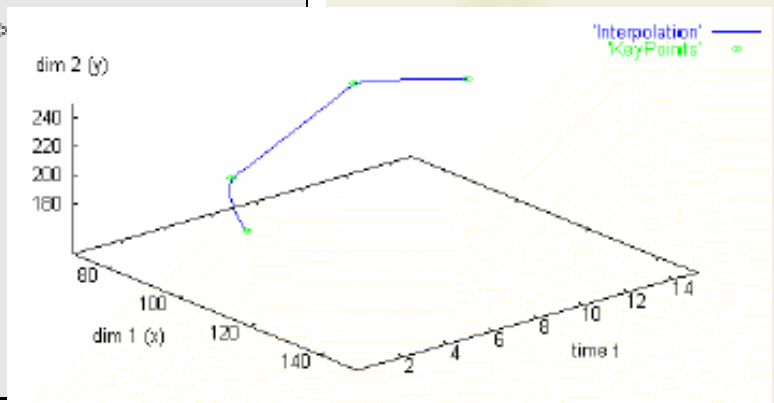


Table 21 - Key points and interpolation functions used in this example. Red colored numbers, which are necessary to derive interpolation functions, can be found in the above description.

t	0.0	0.0 <= t <= 2.0	2.0	2.0 <= t <= 10.5	10.5	10.5 <= t <= 15.0	15.0
x	118.9	$x=2.1t^2-12.6t+118.9$	102.1	$x=-2.3t+106.73$	82.35	$x=0.2t^2-4.4t+106.5$	85.5
y	210.0	$y=5.4t+210.0$	220.8	$y=0.95t+218.89$	228.9	$y=3.07t+201.13$	215.1

MPEG-7 Visual Description tools
Basic Structures – Temporal Interpolation (III)

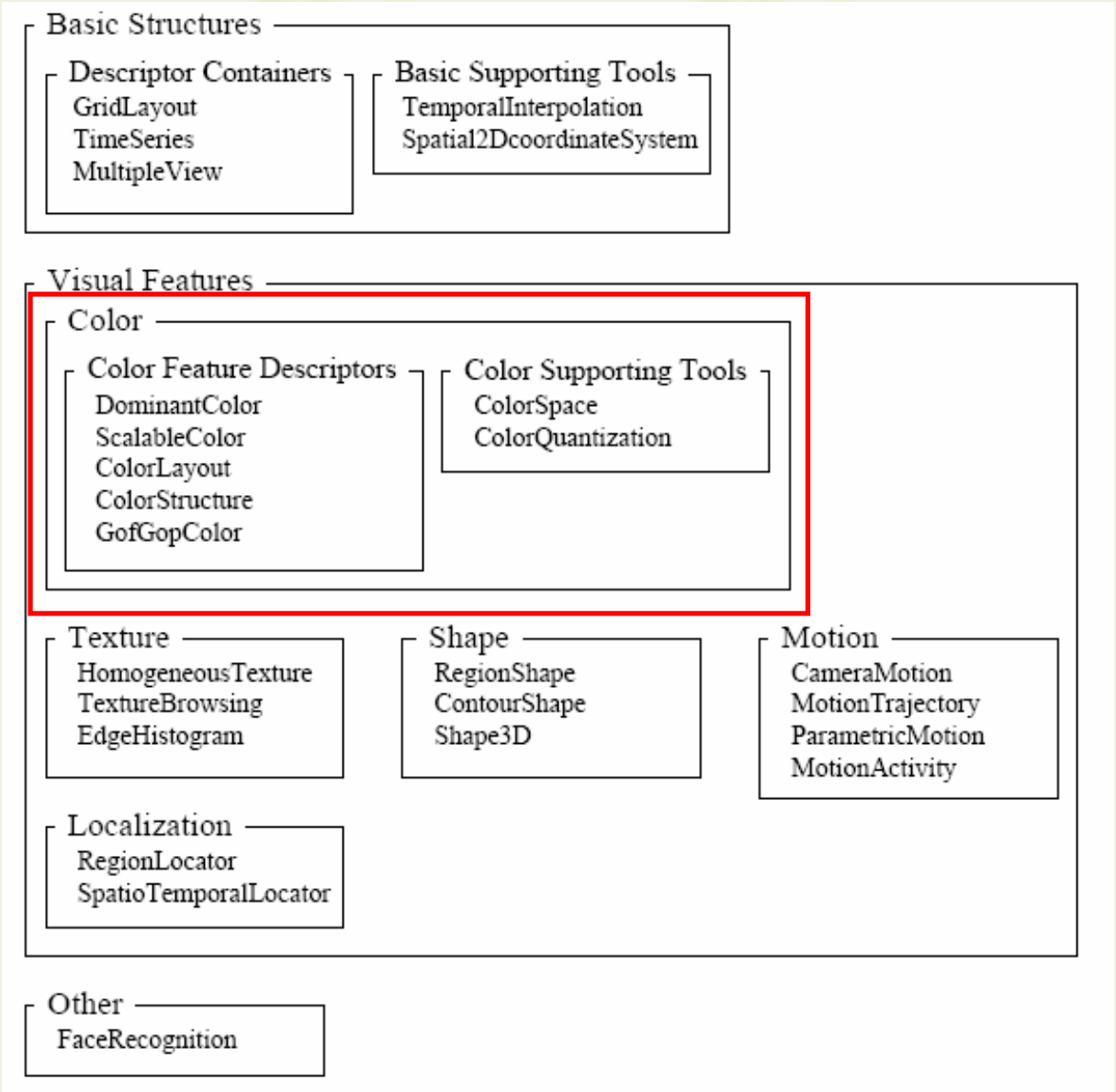
TemporalInterpolation {	Number of bits	Mnemonic
NumOfKeyPoints	16	uimsbf
ConstantTimeInterval	1	bslbf
QuantizationFlag	1	bslbf
if (ConstantTimeInterval) {		
WholeIntervaleDataType	1	bslbf
if (!WholeIntervalDataType) {		
MediaDuration	See annex B	mediaDurationType
} else {		
MediaIncrDuration	See annex B	MediaIncrDurationType
} else {		
KeyTimePointDataType	2	bslbf
if(KeyTimePointDataType==00) {		
for(j=0; j<NumOfKeyPoints; j++)		
MediaTimePoint	See annex B	mediaTimePointType
} else if(KeyTimePointDataType==01) {		
for(j=0; j<NumOfKeyPoints; j++)		
MediaRelTimePoint	See annex B	
} else if (KeyTimePointDataType==10) {		
for(j=0; j<NumOfKeyPoints; j++)		
MediaRelIncrTimePoint	See annex B	
}		
}		

Dimension	4	uimsbf
for(j=0; j<Dimension; j++) {		
DefaultFunction	1	bslbf
for(k=0; k<NumOfKeyPoints; k++) {		
if(!DefaultFunction) {		
type	2	bslbf
if(type==10) {		
param	32	fsbf
}		
}		
if(!QuantizationFlag) {		
KeyValue	32	fsbf
} else {		
QuantizedKeyValue	if (j==0) XRepr else if (j==1) YRepr	uimsbf
}		
}		
}		
}		

Fin Unidad 3

60 aprox (se compensa con la 4)

MPEG-7 Visual Description tools



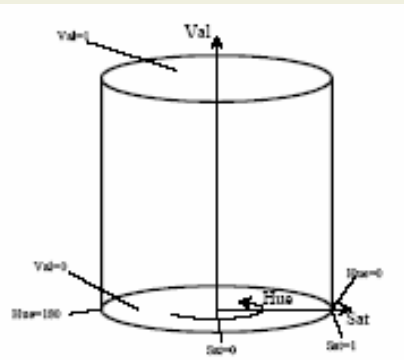
MPEG-7 Visual Description tools
Colour – Color Space (I)

Color space: a datatype for specifying the color space in which the color descriptors are expressed.

The supported color spaces are:

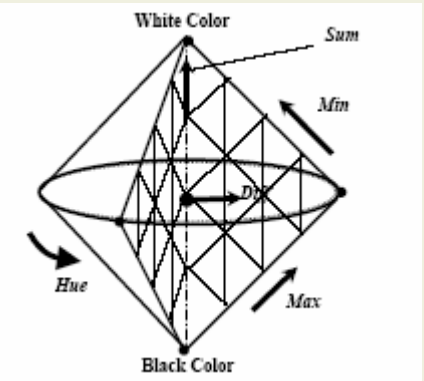
- RGB
  - With or without reference primaries
- YCbCr
- HSV
- HMMD
- Linear transformation matrix with reference to RGB
- Monochrome (Y)

$$
\begin{aligned}
Y &= 0.299*R + 0.587*G + 0.114*B \\
Cb &= -0.169*R - 0.331*G + 0.500*B \\
Cr &= 0.500*R - 0.419*G - 0.081*B
\end{aligned}
$$




```
<ColorSpace type = "HMMD"/>
```

```
<ColorSpace type = "LinearMatrix">
   <ColorTransMat>
   4898 9617 2359 -2769 -5424 8192 8192 -6865 -1328
   </ColorTransMat>
</ColorSpace>
```



MPEG-7 Visual Description tools
Colour – Color Space (II)

ColorSpace {	Number of bits	Mnemonic
colorReferenceFlag	1	bslbf
type	4	bslbf
if (type=='LinearMatrix') {		
for(j=0; j<3; j++) {		
for(k=0; k<3; k++) {		
ColorTransMat[j][k]	16	uimsbf
}		
}		
}		
}		

Table 8 — CIE chromaticities for reference RGB primaries and illuminant white.

	Red	Green	Blue	White point (D65)
x	0.6400	0.3000	0.1500	0.3127
y	0.3300	0.6000	0.0600	0.3290
z	0.0300	0.1000	0.7900	0.3583

MPEG-7 Visual Description tools
Colour – Color Quantization

Colour Quantization: this descriptor defines the uniform quantization of a color component

- 1 component for monochrome
- 3 for the other color spaces
  - HDDM: either (H,Max,Min) or (H,Sum,Diff)

```
<ColorQuantization>
    <Component> H </Component>
    <NumOfBins> 256 </NumOfBins>
    <Component> Diff </Component>
    <NumOfBins> 256 </NumOfBins>
    <Component> Sum </Component>
    <NumOfBins> 256 </NumOfBins>
</ColorQuantization>
```

ColorQuantization {	Number of
for(k=0; k<NumOfComponents; k++) {	
Component[k]	5
NumOfBins[k]	12
}	
}	

Component	Meaning
00000	R
00001	G
00010	B
00011	Y
00100	Cb
00101	Cr
00110	H
00111	S
01000	V
01001	Max
01010	Min
01011	Diff
01100	Sum
01101-11111	Reserved

MPEG-7 Visual Description tools
Colour – Dominant Color (I)

Dominant Color: dominant colours in an arbitrary shaped region

- Up to 8 colours extracted using the Generalized Lloyd Algorithm in the RGB colour space
- DC = {{ci, pi, vi}, s}
  - ci : representative colour
  - pi : percentage
  - vi : variance (optional)
  - s : spatial coherency (weighted si) [if 0 not computed]
- By default RGB colour space and 5 bits colour quantization

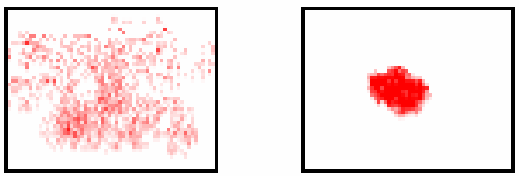


Figure 11 — Examples of low (a) and high (b) spatial coherency of color.

```
- <VisualDescriptor xsi:type="DominantColorType">
    <SpatialCoherency>0</SpatialCoherency>
  - <Value>
      <Percentage>9</Percentage>
      <Index>3 2 2</Index>
    </Value>
  - <Value>
      <Percentage>13</Percentage>
      <Index>8 6 5</Index>
    </Value>
  - <Value>
      <Percentage>3</Percentage>
      <Index>13 11 12</Index>
    </Value>
  - <Value>
      <Percentage>0</Percentage>
      <Index>23 20 18</Index>
    </Value>
  - <Value>
      <Percentage>3</Percentage>
      <Index>12 9 6</Index>
    </Value>
  - <Value>
      <Percentage>1</Percentage>
      <Index>18 17 19</Index>
    </Value>
  </VisualDescriptor>
```

MPEG-7 Visual Description tools
Colour – Dominant Color (II)

DominantColor {	Number of bits	Mnemonic
Size	3	uimsbf
ColorSpacePresent	1	bslbf
if(ColorSpacePresent) {		
ColorSpace	See subclause 6.2.3	ColorSpaceType
}		
ColorQuantizationPresent	1	bslbf
if(ColorQuantizationPresent) {		
ColorQuantization	See subclause 6.3.3	ColorQuantizationType
}		
VariancePresent	1	bslbf
SpatialCoherency	5	uimsbf
for(k=0; k<Size; k++) {		
Percentage	5	uimsbf
for(m=0; m<3; m++) {		
Index	1-12	uimsbf
if(VariancePresent) {		
ColorVariance	1	uimsbf
}		
}		
}		
}		

MPEG-7 Visual Description tools
Colour – Dominant Color (III)

Work proposal 1: Dominant Color (clause 6.4 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the Dominant Color D from a set of images in a folder
  - Couple of slides describing the algorithms and the program
  - clause 4.2.3.1 (ISO/IEC TR 15938-8)
- Similarity matching:
  - Matlab program that selects an image from an image folder, extracts the Dominant Color D and provides a ranked list of most similar images in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.2.3.2 (ISO/IEC TR 15938-8)

MPEG-7 Visual Description tools
Colour – Scalable Color (I)

Scalable Color: scalable (bin numbers and bits per bin) colour histogram

- retrieval accuracy increases with number of bits of D
- histogram in HSV colour space encoded by a Haar transform

ScalableColor {	Number of bits	Mnemonic
numOfCoeff	3	bslbf
numOfBitplanesDiscarded	3	bslbf
for(k=0; k<numOfCoeff; k++) {		
CoefficientSign	1	bslbf
}		
for(k=0; k<8-numOfBitplanesDiscarded; k++) {		
Bitplane[k]	BitplaneSize	bslbf
}		
}		

numOfCoeff	Meaning
000	16
001	32
010	64
011	128
100	256
101-111	Reserved

numOfBitplanesDiscarded	Meaning
000	0
001	1
010	2
011	3
100	4
101	6
110	8
111	Reserved

MPEG-7 Visual Description tools
Colour – Scalable Color (II)

Work proposal 2: Scalable color (clause 6.5 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the Scalable Color D from a set of images in a folder
  - Number of coeeficients and bitplanes are gives as parameters
  - Couple of slides describing the algorithms and the program
  - clause 4.2.4.1 (ISO/IEC TR 15938-8)
- Similarity matching:
  - Matlab program that selects an image from an image folder, extracts the Scalable Color D (with a number of coefficients and bitplanes) and provides a ranked list of most similar images in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.2.4.2 (ISO/IEC TR 15938-8)

“not too clearly explained”

May be done as group work together with Work proposal 4

MPEG-7 Visual Description tools
Colour – Color Layout (I)

Color Layout: spatial distribution of colours for high speed retrieval and browsing

- Usefull for image-to-image matching and layout based retrieval (sketch-to-image)
- Can be applied to the whole image (or selected frame) or to an arbitrarily shaped region (or selected region from a moving region)
- DCT coefficients of a 8x8 matrix of local representative colours
  - How to calculate the local representative colours is non-normative (e.g., average colour)

```
<ColorLayout>
    <YDCCoeff>50</YDCCoeff>
    <CbDCCoeff>34</CbDCCoeff>
    <CrDCCoeff>30</CrDCCoeff>
    <YACCoeff5>16 12 15 12 17</YACCoeff5>
    <CbACCoeff2>12 17</CbACCoeff2>
    <CrACCoeff2>12 14</CrACCoeff2>
</ColorLayout>
```

MPEG-7 Visual Description tools
Colour – Color Layout (II)

ColorLayout {	Number	
CoeffPattern	1-2	
if(CoeffPattern==11) {		
NumOfYCoeffIndex	3	
NumOfCCoeffIndex	3	
}		
YDCCoeff	6	uimsbf
CbDCCoeff	6	uimsbf
CrDCCoeff	6	uimsbf
for(k=1; k<NumOfYCoeff; k++) {		
YACCoeff	5	uimsbf
}		
for(k=1; k<NumOfCCoeff; k++) {		
CbACCoeff	5	
}		
for(k=1; k<NumOfCCoeff; k++) {		
CrACCoeff	5	
}		
}		

CoefficientPattern	Meaning	
	NumOfYCoeff	NumOfCCoeff
0	6	3
10	6	6
11	Specified by NumOfYCoeffIndex	Specified by NumOfCCoeffIndex

NumOfYCoeffIndex/NumOfCCoeffIndex	NumOfYCoeff, NumOfCCoeff
000	reserved
001	3
010	6
011	10
100	15
101	21
110	28
111	64

MPEG-7 Visual Description tools
Colour – Color Layout (III)

Feature extraction:

- Calculate a 8x8 matrix of local representative colors (e.g., average)
- Apply DCT
- Quantize (DC 6 bits, AC 5 bits)
- Zig-zag scanning

```
YCoeff[zigzag[i,j]] = YC[i][j]
CbCoeff[zigzag[i,j]] = CbC[i][j]
CrCoeff[zigzag[i,j]] = CrC[i][j]
```

	i							
j	0	1	5	6	14	15	27	28
	2	4	7	13	16	26	29	42
	3	8	12	17	25	30	41	43
	9	11	18	24	31	40	44	53
	10	19	23	32	39	45	52	54
	20	22	33	38	46	51	55	60
	21	34	37	47	50	56	59	61
	35	36	48	49	57	58	62	63

```
int quant_Y_DC(int i) {
  int j;
  i = i/8;
  if(i>192) j=112+(i-192)/4;
  else if(i>160) j=96+(i-160)/2;
  else if(i>96) j=32+i-96;
  else if(i>64) j=16+(i-64)/2;
  else j=i/4;
  return j>>1;
}
int quant_CbCr_DC(int i) {
  int j;
  i = i/8;
  if(i>191) j=63;
  else if(i>160) j=56+(i-160)/4;
  else if(i>144) j=48+(i-144)/2;
  else if(i>112) j=16+i-112;
  else if(i>96) j=8+(i-96)/2;
  else if(i>64) j=(i-64)/4;
  else j=0;
  return j;
}
int quant_Y_AC(int i) {
  int j;
  i = i/2;
  if(i>255) i=255;
  if(i<-256) i=-256;
  if(abs(i)>127) j=64+abs(i)/4;
  else if(abs(i)>63) j=32+abs(i)/2;
  else j=abs(i);
  j = (i<0)?-j:j;
  return (int)trunc(((double)j+128.0)/8.0+0.5);
}
int quant_CbCr_AC(int i) {
  int j;
  if(i>255) i=255;
  if(i<-256) i=-256;
  if(abs(i)>127) j=64+abs(i)/4;
  else if(abs(i)>63) j=32+abs(i)/2;
  else j=abs(i);
  j = (i<0)?-j:j;
  return (int)trunc(((double)j+128.0)/8.0+0.5);
}
```

MPEG-7 Visual Description tools
Colour – Color Layout (IV)

Similarity matching

$$D = \sqrt{\sum_{i=0}^{Max\{NumberOf(Y)Coeff\}-1} \lambda_{Y_i}(YCoeff[i]-YCoeff''[i])^2} + \sqrt{\sum_{i=0}^{Max\{NumberOf(C)Coeff\}-1} \lambda_{Cr_i}(CrCoeff[i]-CrCoeff''[i])^2} + \sqrt{\sum_{i=0}^{Max\{NumberOf(C)Coeff\}-1} \lambda_{Cb_i}(CbCoeff[i]-CbCoeff''[i])^2}$$

Table 22 - An example of weighting values for the default descriptor

(X)	Coefficient Order					
	0	1	2	3	4	5
Y	2	2	2	1	1	1
Cb	2	1	1			
Cr	4	2	2			

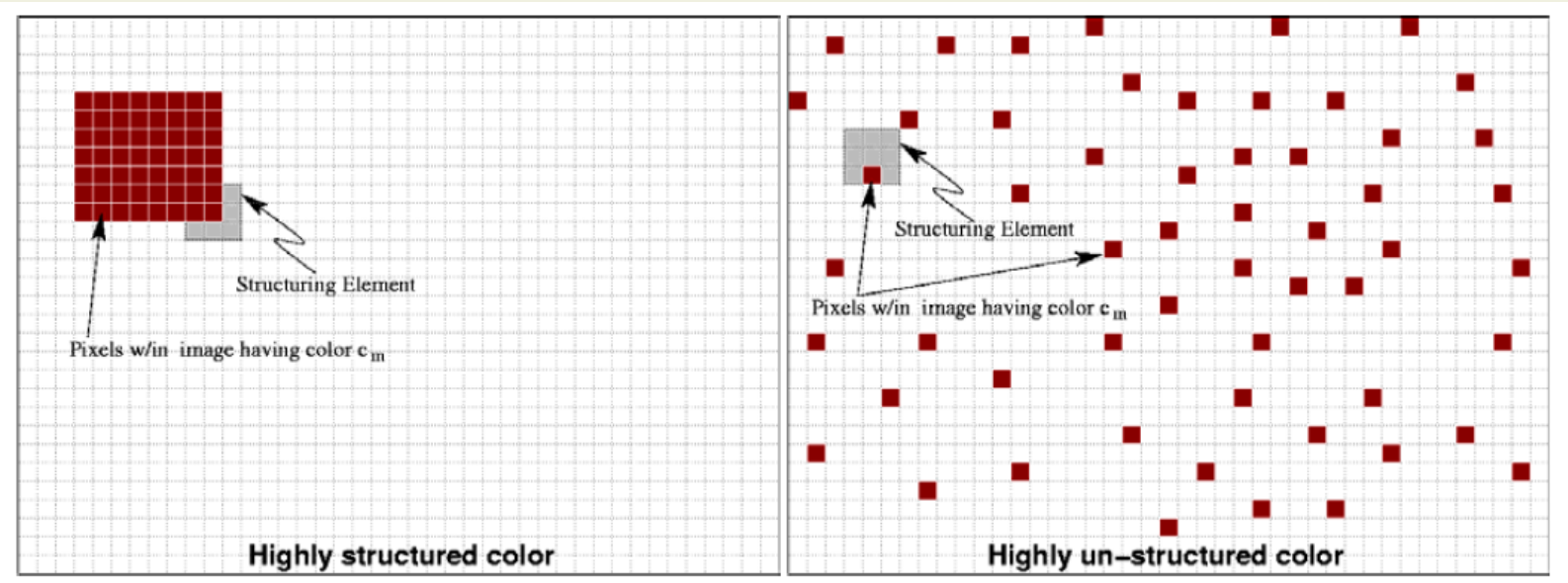
MPEG-7 Visual Description tools
Colour – Color Layout (V)

Work proposal 3: Color layout (clause 6.6 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the Color Layout D from a set of video sequences in a folder
  - Couple of slides describing the algorithms and the program
  - clause 4.2.5.1.2 (ISO/IEC TR 15938-8)
- Similarity matching:
  - Matlab program that selects a video sequence from a folder, extracts the Color Layout D and provides a ranked list of most similar sequences in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.2.5.2.2 (ISO/IEC TR 15938-8)

MPEG-7 Visual Description tools
Colour – Color Structure (I)

Colour Structure: description of colour content and structure of the content



- 256 bins values obtained from the image in the HMMD colour space
  - Resampled to 128, 64, 32

```
<ColorStructure colorQuant="1">
    <Values>
        0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31
    </Values>
</ColorStructure>
```

MPEG-7 Visual Description tools
Colour – Color Structure (II)

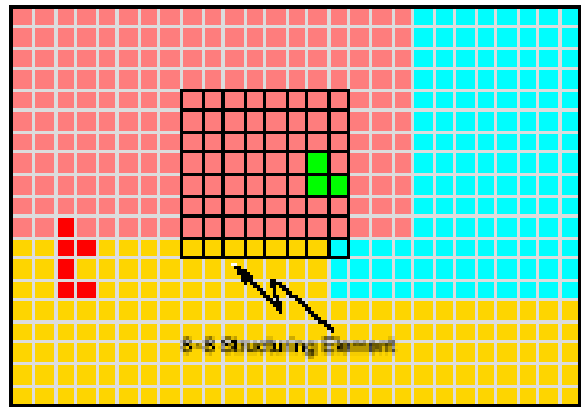
ColorStructure{	Number of bits	Mnemonic
colorQuant	3	uimsbf
NumOfValuesCode	8	uimsbf
for (m=0; m<M; m++) {		
Values[m]	8	uimsbf
}		
}		

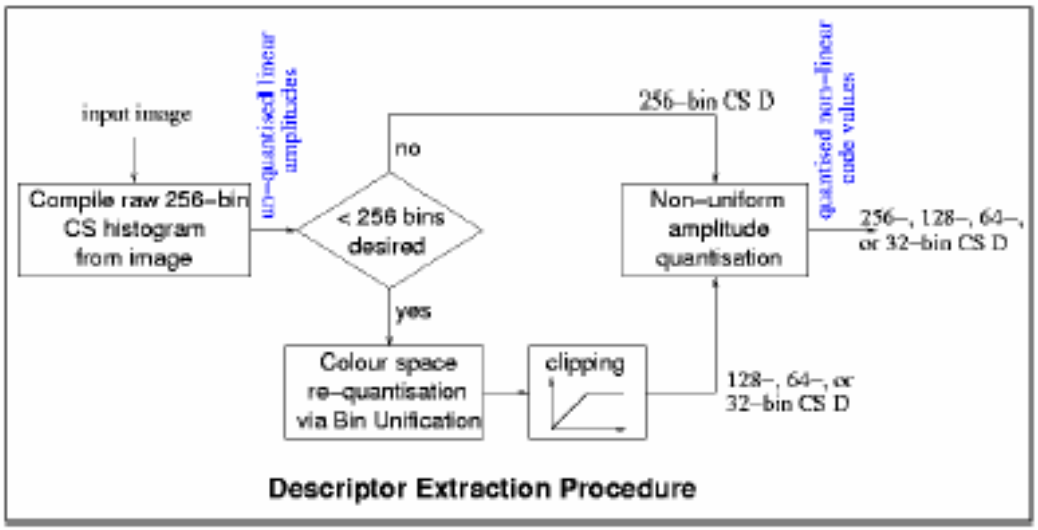
colorQuant	operating point
0	Forbidden
1	32 (HMMD)
2	64 (HMMD)
3	128 (HMMD)
4	256 (HMMD)
5-7	Reserved

MPEG-7 Visual Description tools
Colour – Color Structure (III)

Feature extraction:

- Convert image to HMMD
- Scan the image using the structuring element (8x8 for CIF, for others the S.E. may be resampled –always 64 elements but may include “holes”-)
- The standard specifies the re-quantifications of the bin and the non-uniform quantisation



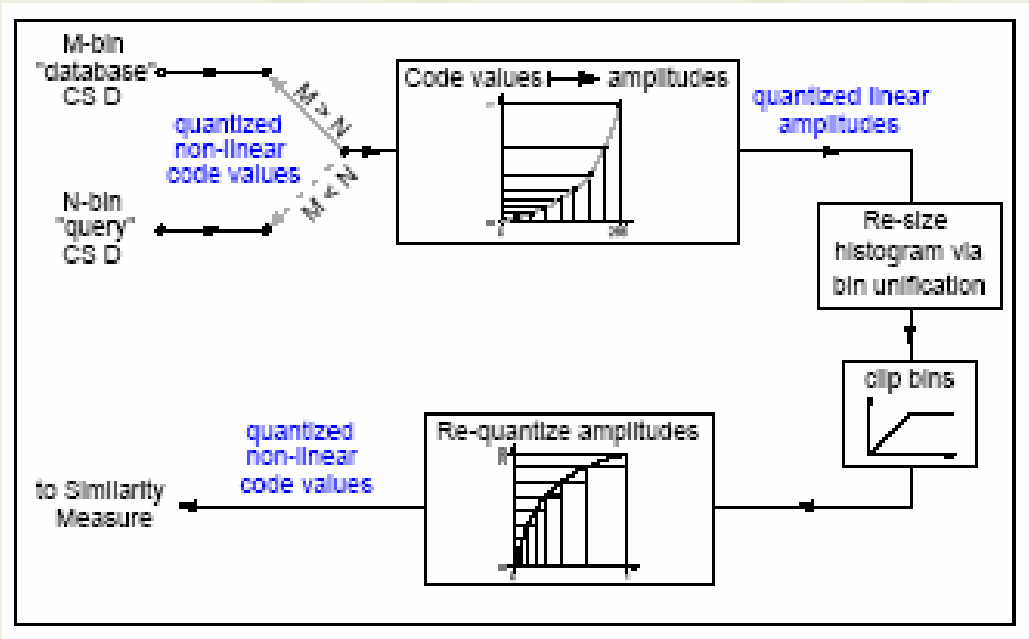


MPEG-7 Visual Description tools
Colour – Color Structure (IV)

Similarity matching

$$dist(A,B) \equiv \left| \underline{h}_A - \underline{h}_B \right|_1 = \sum_n \left| \underline{h}_A(n) - \underline{h}_B(n) \right|$$

- If the histograms are of different number of bins, the larger should be re-normalized before comparision



MPEG-7 Visual Description tools
Colour – GoF/GoP Color (I)

GoF/GoP Color: Scalable Color of a group of images or video frames

- average, median or intersection of individual histograms before Haar Transform (Scalable Color)

$$Avg\_Histogram\_value(j) = \frac{1}{N}\sum_{i=0}^{N} Histogram\_value_i(j),$$
$$j = 0,\ldots,number\_histogram\_bins - 1$$

$$Med\_Histogram\_value(j) = \text{median}(Histogram\_value_0(j), \ldots, Histogram\_value_{N-1}(j)), \quad j = 0, \ldots, 255.$$

$$Int\_Histogram\_value(j) = \min(Histogram\_value_i(j)), \quad j = 0, \ldots, 255.$$

GofGoPColor {	Number of bits	Mnemonic
aggregation	2	bslbf
ScalableColor	See subclause 6.5.3	ScalableColorType
}		

aggregation	Meaning
00	Average
01	Median
10	Intersection
11	Reserved

MPEG-7 Visual Description tools
Colour – GoF/GoP Color (II)

Work proposal 4: GoF/GoP Color (clause 6.8 ISO/IEC FDIS 15938-3)

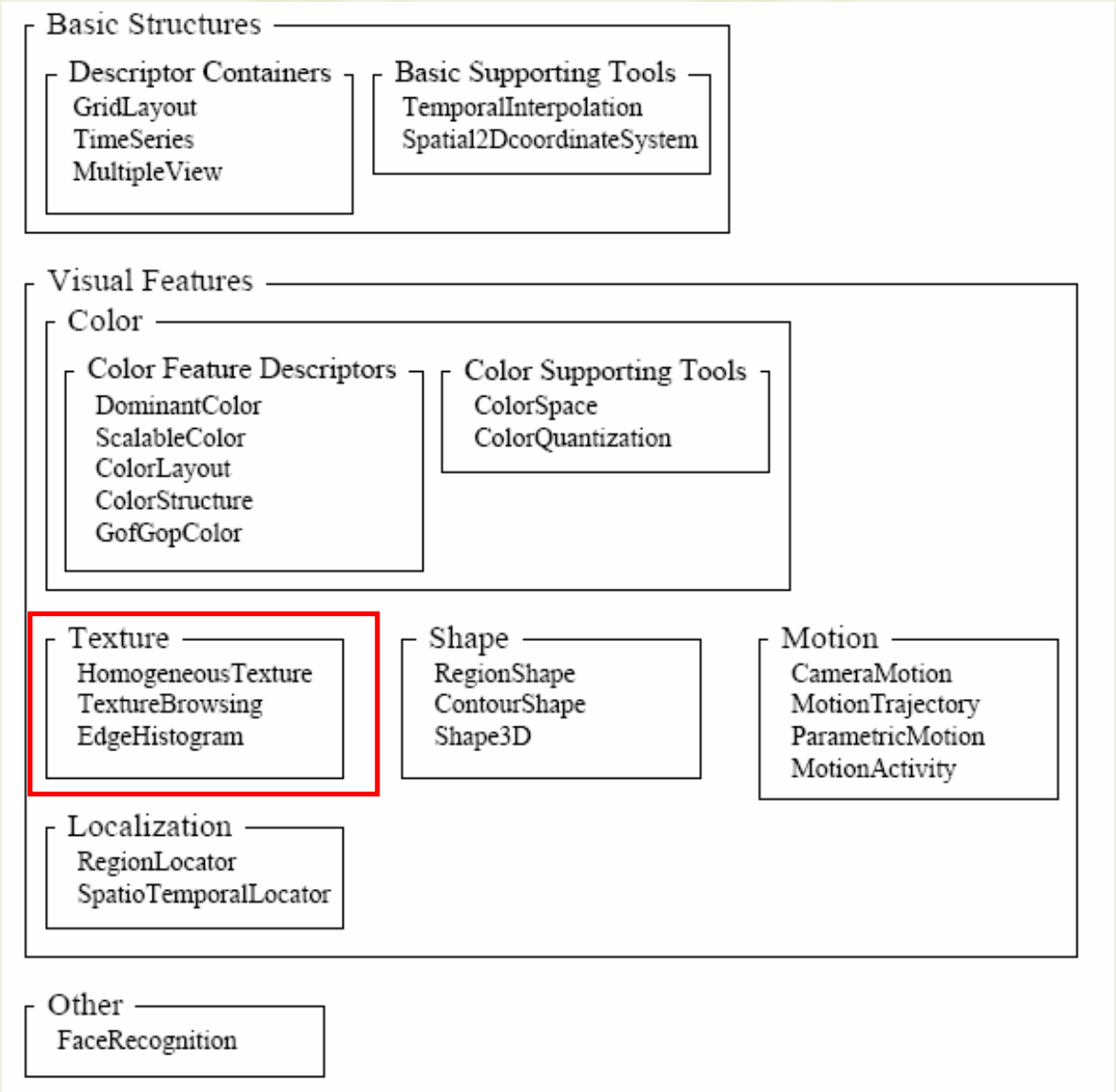
- Feature extraction:
  - Matlab program that extracts the GoF/GoP Color D from a set of video sequences in a folder
  - Couple of slides describing the algorithms and the program
  - clause 4.2.7.1 (ISO/IEC TR 15938-8)
- Similarity matching and use cases:
  - Matlab program that selects a video sequence from a folder, extracts the GoF/GoP D and provides a ranked list of most similar sequences in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.2.7.2 and 4.2.7.3 (ISO/IEC TR 15938-8)

May be done as group work together with Work proposal 2

Fin Unidad 4

40 aprox (compensa la 3)

MPEG-7 Visual Description tools



MPEG-7 Visual Description tools
Texture – HomogeneousTexture (I)

Homogeneous texture: descriptor based on the energy and energy deviation in a frequency layout: 30 channels: 6 directions x 5 scales

- Gabor functions (to avoid discontinuities in the frontiers)
- 62 floats
  - Mean and standard deviation of intensity and 30 energy (and optionally 30 energy deviation)
  - All are uniform quantized to 8 bits, within a predefined range (min/max values defined in the standard)

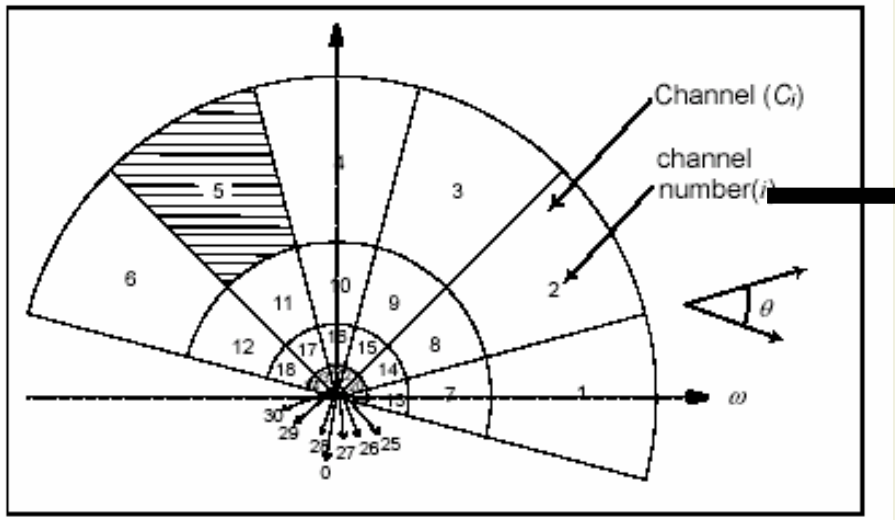


Fig. 1. Frequency region division with HVS filter.

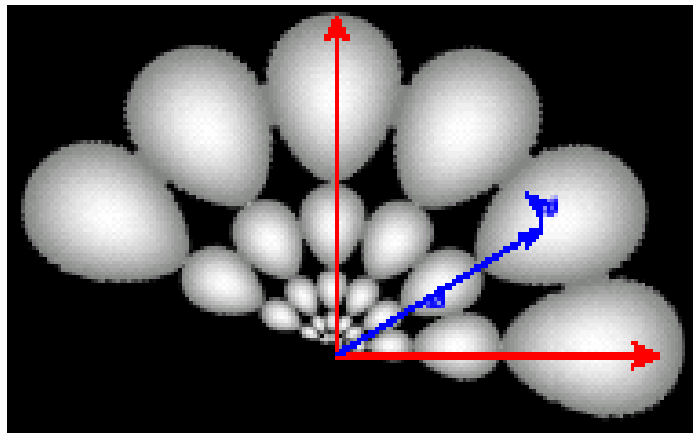


Fig. 5. S66 Gabor filters in polar coordinate system.

MPEG-7 Visual Description tools
Texture – HomogeneousTexture (II)

HomogeneousTexture {	Number of bits	Mnemonic
EnergyDeviationFlag	1	bslbf
Average	8	uimsbf
StandardDeviation	8	uimsbf
for(k=0; k<30; k++) {		
Energy[k]	8	uimsbf
}		
if (energyDeviationFlag) {		
for(k=0; k<30; k++) {		
EnergyDeviation[k]	8	uimsbf
}		
}		
}		

```
<<HomogeneousTexture>
    <Average>125</Average>
    <StandardDeviation>10</StandardDeviation>
    <Energy>51 32 23 42 25 67 81 48 79 110 16 42 73 84 75 66 57 48 39 18 16 25
34 44 52 66 8 82 96 71</Energy>
    <EnergyDeviation>39 18 16 25 34 34 52 36 48 52 26 31 51 32 23 42 25 37 31 48
19 11 16 42 43 14 35 46 9 48</EnergyDeviation>
</HomogeneousTextureg>
```

MPEG-7 Visual Description tools
Texture – HomogeneousTexture (III)

Work proposal 5: Homogeneous Texture (clause 7.2 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the Homogeneous Texture D from a set of images in a folder (see clause 4.3.1.4 ISO/IEC TR 15938-8)
  - Couple of slides describing the algorithms and the program
  - clause 4.3.1.1 (ISO/IEC TR 15938-8)
- Similarity matching and use cases:
  - Matlab program that selects an image from a folder, extracts the HomogeneousTexture D and provides a ranked list of most similar images in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.3.1.2 and 4.3.1.4 (ISO/IEC TR 15938-8)

MPEG-7 Visual Description tools
Texture – Texture Browsing (I)

Texture Browsing: efficient (12 bits) human-like perceptual characterisation

- regularity, coarness and directionality (enumerated types)
- browsing and coarse classification

Table 29 — Semantics of Regularity.

Regularity	Semantics
00	irregular
01	slightly regular
10	regular
11	highly regular

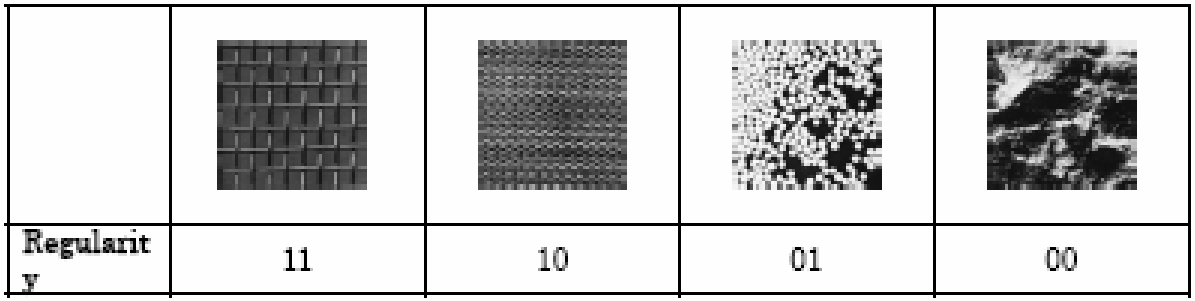


Figure 20 — Examples of Regularity.

```
<TextureBrowsing>
          <Regularity>highlyregular</Regularity>
          <Direction>0degree</Direction>
          <Scale>medium</Scale>
          <Direction>90degree</Direction>
          <Scale>medium</Scale>
</TextureBrowsing>
```

MPEG-7 Visual Description tools
Texture – Texture Browsing (II)

TextureBrowsing {	Number of bits	Mnemonic
NumOfComponentsFlag	1	bslbf
Regularity	2	bslbf
for(k=0; k<=NumOfComponents; k++) {		
Direction	3	bslbf
Scale	2	bslbf
}		
}		

Regularity	Semantics
00	irregular
01	slightly regular
10	regular
11	highly regular

Direction	Semantics
000	no directionality
001	0 degree
010	30 degree
011	60 degree
100	90 degree
101	120 degree
110	150 degree
111	Reserved

Scale	Semantics
00	fine (0)
01	medium (1)
10	coarse (2)
11	very coarse (3,4)

MPEG-7 Visual Description tools
Texture –Texture Browsing (III)

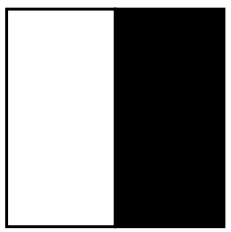
Work proposal 6: Texture Browsing (clause 7.3 ISO/IEC FDIS 15938-3)

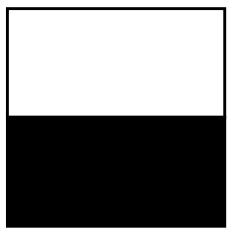
- Feature extraction:
  - Matlab program that extracts the TextureBrowsing D from a set of images in a folder
  - Couple of slides describing the algorithms and the program
  - clause 4.3.2.1 (ISO/IEC TR 15938-8)
- Similarity matching and use cases:
  - Matlab program that selects an image from a folder, extracts the TextureBrowsing D and provides a ranked list of most similar images in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.3.2.2 and 4.3.2.4 (ISO/IEC TR 15938-8)

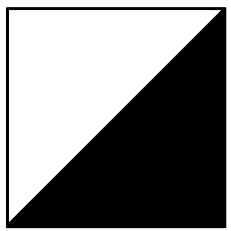
MPEG-7 Visual Description tools
Texture –Edge histogram (I)

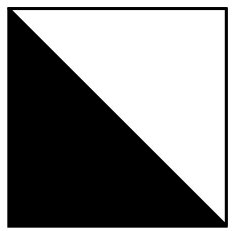
Edge histogram: spatial distribution of 5 types of edges in 4x4 subimages

- For each subimage (16) the number of types of edges (5) is provided (80 bin values)
- Size of image blocks depends on image size
  - Number of image blocks fixed (default 1100 desired in the whole image)
  - Number of edges in each image block depend on image size: bin values are normalized and quantified (tables in the standard) to 3 bits
- image-to-image and sketch-to-image matching


a) vertical edge


b) horizontal edge


c) 45 degree edge


d) 135 degree edge


e)non-directional edge

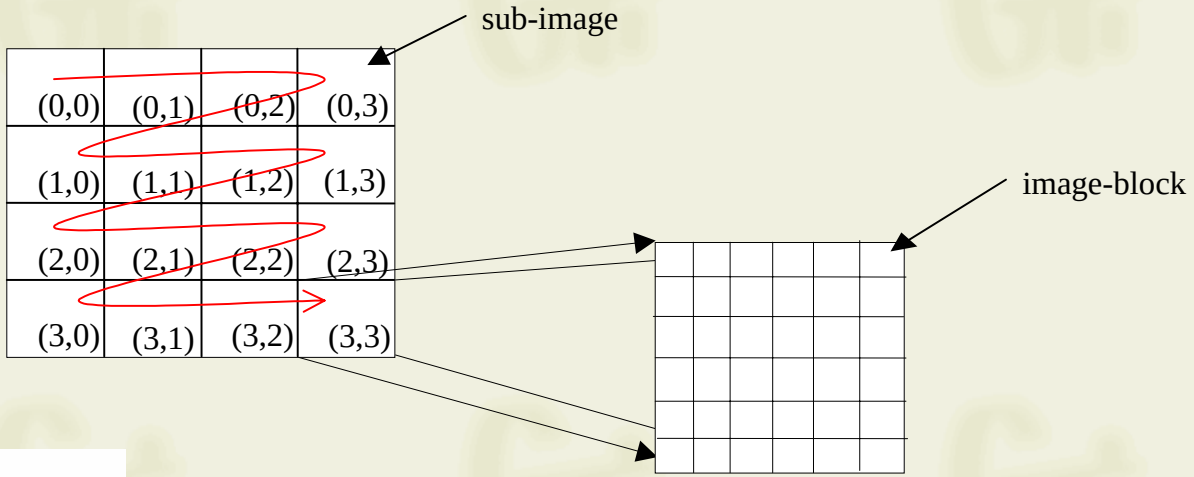
MPEG-7 Visual Description tools
Texture –Edge histogram (II)

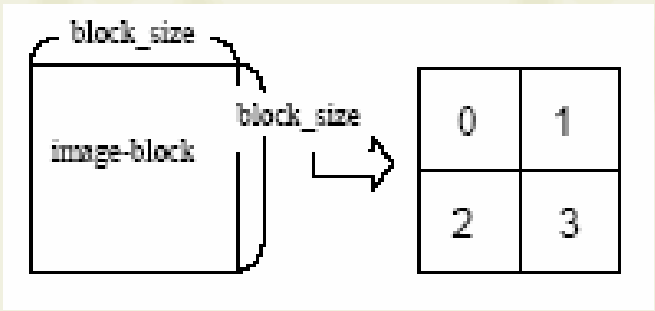
EdgeHistogram {	Number of bits	Mnemonic
for(k=0; k<80; k++) {		
BinCounts[k]	3	uimsbf
}		
}		

BinCounts[k]	Semantics
BinCounts[0]	Vertical edges in sub-image (0,0)
BinCounts[1]	Horizontal edges in sub-image (0,0)
BinCounts[2]	45 degree edges in sub-image (0,0)
BinCounts[3]	135 degree edges in sub-image (0,0)
BinCounts[4]	Non-directional edges in sub-image (0,0)
BinCounts[5]	Vertical edges in sub-image (0,1)
*	*
BinCounts[74]	Non-directional edges in sub-image (3,2)
BinCounts[75]	Vertical edges in sub-image (3,3)
BinCounts[76]	Horizontal edges in sub-image (3,3)
BinCounts[77]	45 degree edges in sub-image (3,3)
BinCounts[78]	135 degree edges in sub-image (3,3)
BinCounts[79]	Non-directional edges in sub-image (3,3)

MPEG-7 Visual Description tools
Texture –Edge histogram (III)

Feature extraction





$$A_0(i,j) = \frac{4}{\text{block\_size} \times \text{block\_size}} \sum_{m=0}^{\text{block\_size}/2-1} \sum_{n=0}^{\text{block\_size}/2-1} I_0(m,n) \quad (3)$$

$$A_1(i,j) = \frac{4}{\text{block\_size} \times \text{block\_size}} \sum_{m=\text{block\_size}/2}^{\text{block\_size}-1} \sum_{n=0}^{\text{block\_size}-1} I_0(m,n) \quad (4)$$

$$A_2(i,j) = \frac{4}{\text{block\_size} \times \text{block\_size}} \sum_{m=0}^{\text{block\_size}/2-1} \sum_{n=\text{block\_size}/2}^{\text{block\_size}-1} I_0(m,n) \quad (5)$$

$$A_3(i,j) = \frac{4}{\text{block\_size} \times \text{block\_size}} \sum_{m=\text{block\_size}/2}^{\text{block\_size}-1} \sum_{n=\text{block\_size}/2}^{\text{block\_size}-1} I_0(m,n) \quad (6)$$

MPEG-7 Visual Description tools
Texture –Edge histogram (IV)

1	-1	1	1	$\sqrt{2}$	0	0	$\sqrt{2}$	2	-2
1	-1	-1	-1	0	$-\sqrt{2}$	$-\sqrt{2}$	0	-2	2

a) ver_edge_filter() b) hor_edge_filter() c) dia45_edge_filter() d) dia135_edge_filter() e) nond_edge_filter()

$$ver\_edge\_stg(i,j) = \left| \sum_{k=0}^{3} A_k(i,j) \times ver\_edge\_filter(k) \right|$$

$$hor\_edge\_stg(i,j) = \left| \sum_{k=0}^{3} A_k(i,j) \times hor\_edge\_filter(k) \right|$$

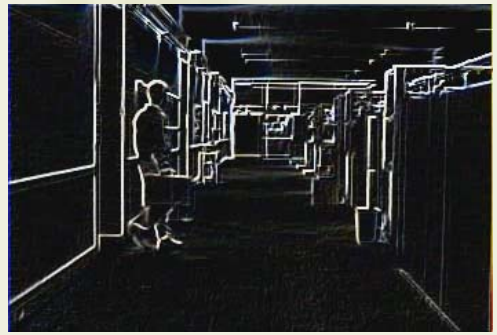
$$dia45\_edge\_stg(i,j) = \left| \sum_{k=0}^{3} A_k(i,j) \times dia45\_edge\_filter(k) \right|$$

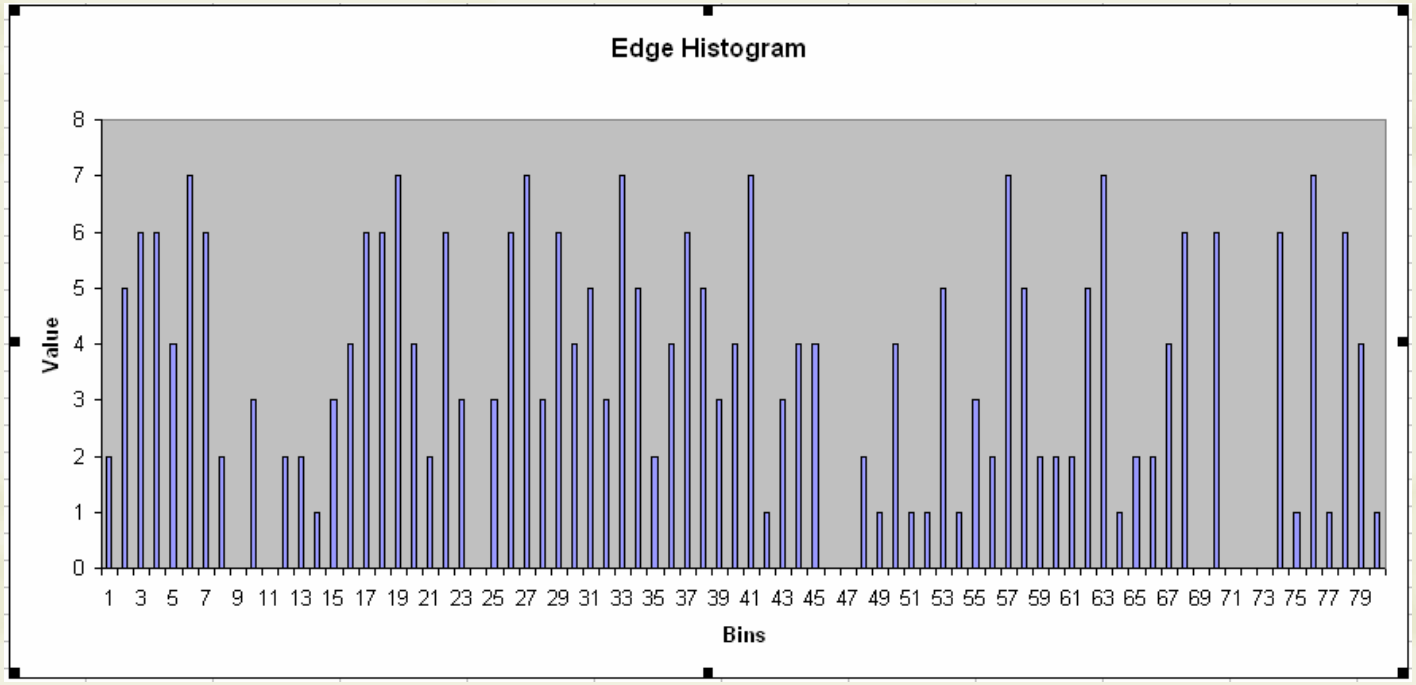
$$dia135\_edge\_stg(i,j) = \left| \sum_{k=0}^{3} A_k(i,j) \times dia135\_edge\_filter(k) \right|$$

$$nond\_edge\_stg(i,j) = \left| \sum_{k=0}^{3} A_k(i,j) \times nond\_edge\_filter(k) \right|$$

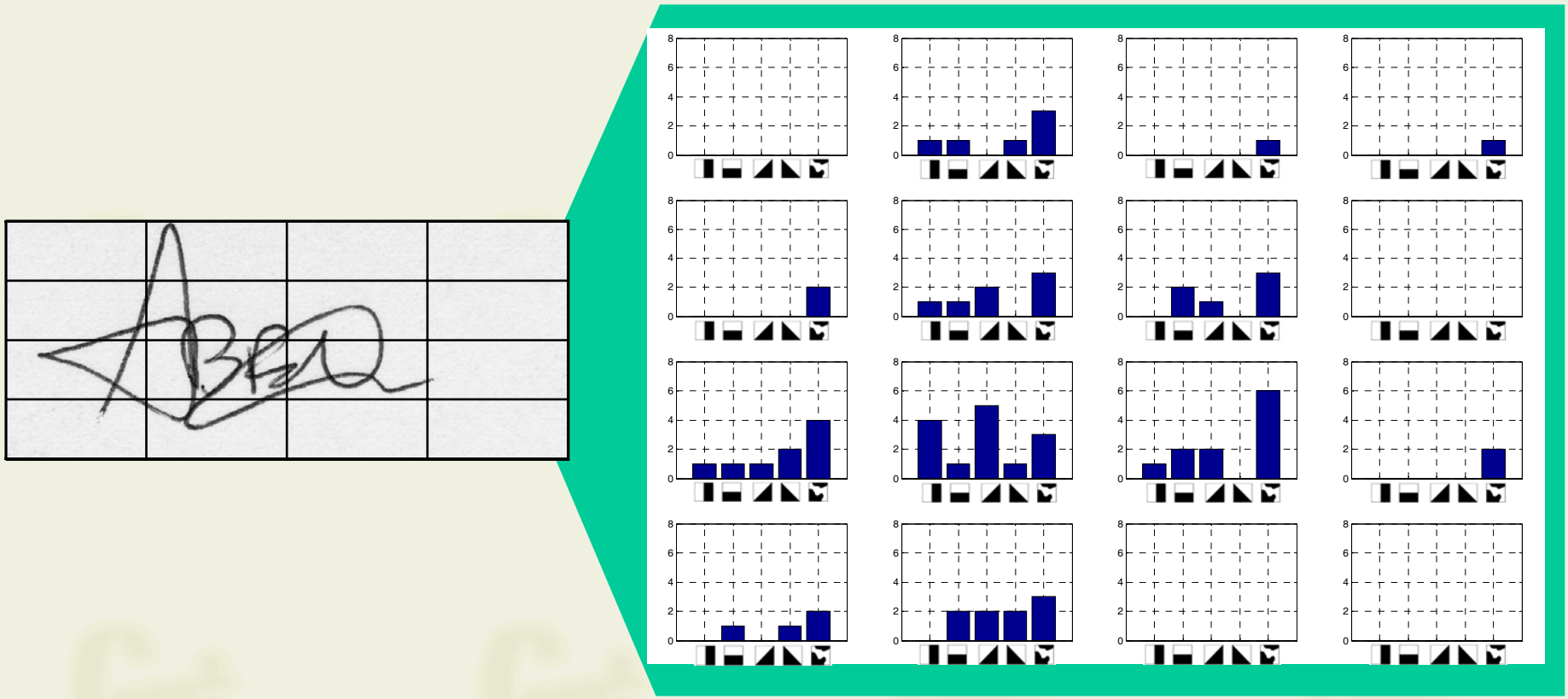
Max(ver_edge_stg(i,j), hor_edge_stg(i,j),
dia45_edge_stg(i,j), dia135_edge_stg(i,j), nond_edge_stg(i,j)) > Thedge

MPEG-7 Visual Description tools
Texture –Edge histogram (V)





MPEG-7 Visual Description tools
Texture –Edge histogram (VI)

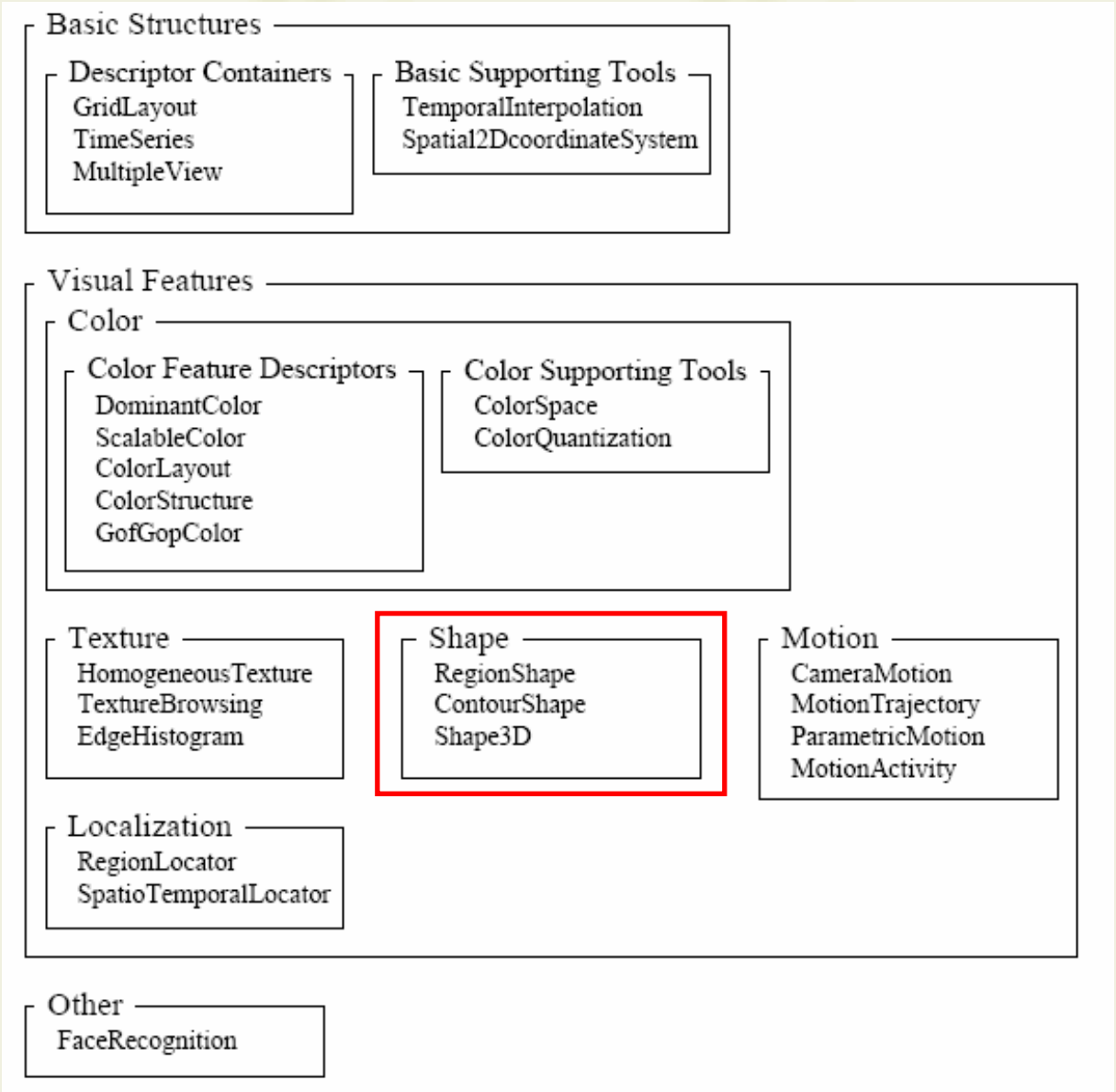


MPEG-7 Visual Description tools
Texture – Edge Histogram (VI)

Work proposal 7: Edge Histogram (clause 7.4 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the Edge Histogram D from a set of images in a folder
  - Couple of slides describing the algorithms and the program
  - clause 4.3.3.1 (ISO/IEC TR 15938-8)
- Similarity matching and use cases:
  - Matlab program that selects an image from a folder, extracts the EdgeHistogram D and provides a ranked list of most similar images in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.3.3.2 and 4.3.3.4 (ISO/IEC TR 15938-8)

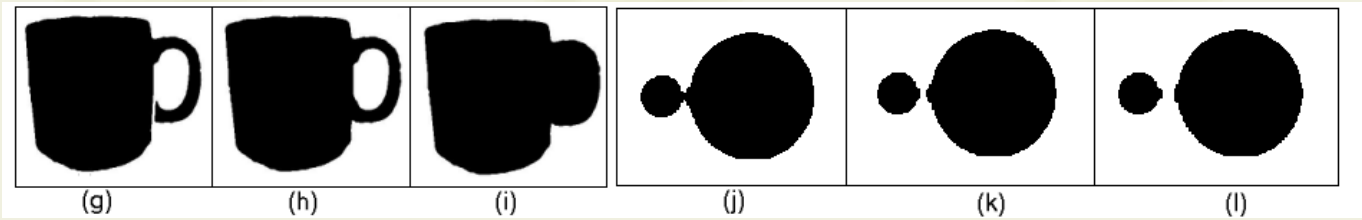
MPEG-7 Visual Description tools

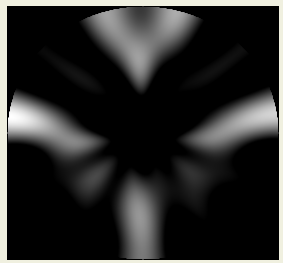


MPEG-7 Visual Description tools
Shape – Region shape (I)

Region Shape: region-based shape of an object, making use of all the object pixels

- Allows to model simple (single region) or complex (set of regions as well as holes) object shapes





MPEG-7 Visual Description tools
Shape – Region shape (II)

RegionShape {	Number of bits	Mnemonic
for(k=0; k<35; k++) {		
MagnitudeOfART[k]	4	uimsbf
}		
}		

k	0	1	2	3	4	…	30	31	32	33	34
n	1	2	0	1	2	…	1	2	0	1	2
m	0	0	1	1	1	…	10	10	11	11	11

The extracted coefficients are normalized and quantized to 4 bits

MPEG-7 Visual Description tools
Shape – Region shape (III)

Feature extraction

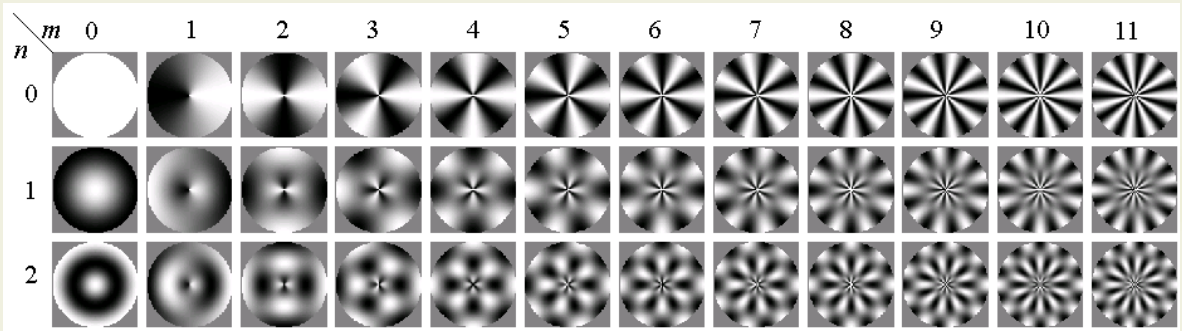
- F_{nm}: 35 ART (Angular Radial Transform) coefficients
  - 2-D transform defined on a unit disk in polar coordinates

$$F_{nm} = \langle V_{nm}(\rho,\theta), f(\rho,\theta) \rangle = \int_0^{2\pi} \int_0^1 V_{nm}^*(\rho,\theta), f(\rho,\theta) \rho d\rho d\theta$$

  - F_{nm} ART coeficient of order *n* and *m*, *f* is the function defininf the image in polar coordinates, and V_{nm} is the base function. ART base functions are separable in angular and radial direcctions as shown below:

$$V_{nm}(\rho,\theta) = A_m(\theta) R_n(\rho) \qquad A_m(\theta) = \frac{1}{2\pi} \exp(jm\theta) \qquad R_n(\rho) = \begin{cases} 1 & n = 0 \\ 2\cos(\pi n \rho) & n \neq 0 \end{cases}$$

  - ART basis functions (real part)



MPEG-7 Visual Description tools
Shape – Region shape (IV)

Similarity matching

- L1 norm after reconstruction

$$D(Q,D) = \sum_{i=0}^{34} \left| \begin{array}{l} InverseQuantize(MagnitudeOfART_Q[i]) \\ - InverseQuantize(MagnitudeOfART_D[i]) \end{array} \right|$$

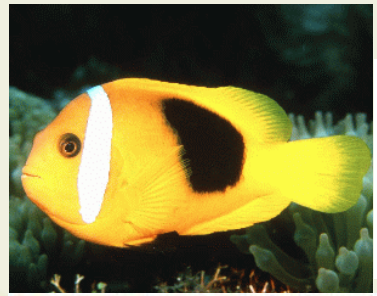
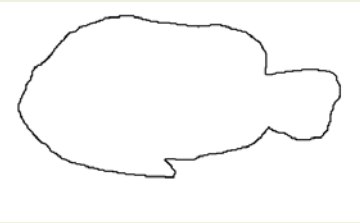
Table 27 - Reconstruction value for ART coefficients.

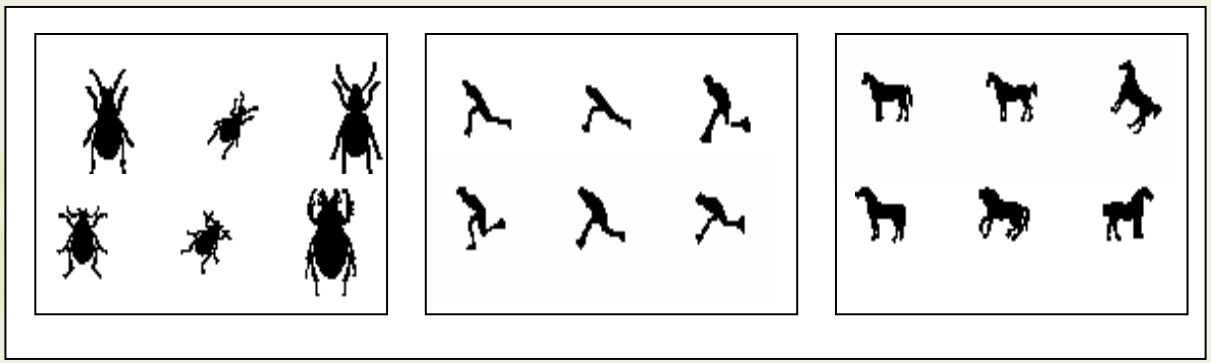
MagnitudeOfART	Reconstruction value
0000	0.001763817
0001	0.005468893
0010	0.009438835
0011	0.013714449
0100	0.018346760
0101	0.023400748
0110	0.028960940
0111	0.035140141
1000	0.042093649
1001	0.050043696
1010	0.059324478
1011	0.070472849
1100	0.084434761
1101	0.103127662
1110	0.131506859
1111	0.192540857

MPEG-7 Visual Description tools
Shape – Contour shape (I)

Contour-based Shape: closed contour of a 2D object or region

- Curvature Scale Space (CSS) representation
- Very compact (below 14 bytes in average)
- Properties
  - shape generalisation (perceptual similarity)
  - robustness to non-rigid motion
  - robustness to partial occlusion

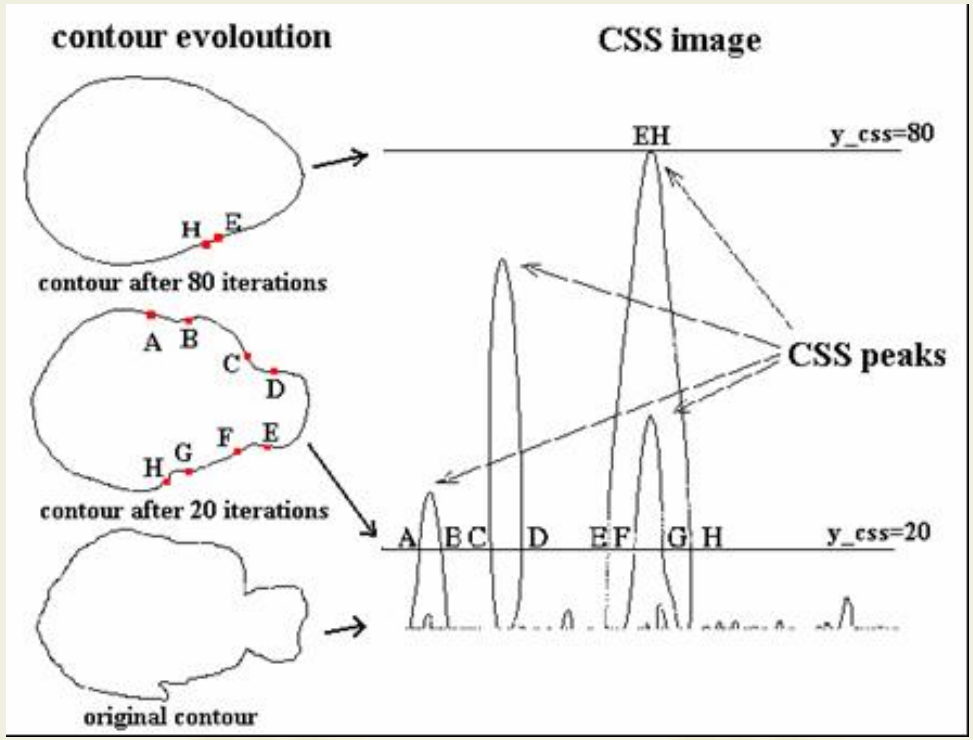





MPEG-7 Visual Description tools
Shape – Contour shape (II)

CSS Image

- x_css: 1..NSamples
  - NSamples == countour points (equidistant)
- y_css: number of filtering steps
  - Low pass filtering with kernel [0.25, 0.5, 0.25]
  - Until convex contour
- For each row (y_css) the x_css points (binary pixel[x_css,y_css]) are set to 1 (black) when the curvature function crosses zero at that point (curvature change)
  - Twice each concave segment



MPEG-7 Visual Description tools
Shape – Contour shape (III)

Feature extraction

- From the CSS image the “prominent” peaks are extracted
  - Points (x_css,y_css)
- The points/peaks are ordered by decreasing value of y_css, transformed using a non-linear transformation and quantizied.
  - Up to 64 peaks are used
  - As non-normative guidance, peaks less than 0.05 the maximum are removed
- The eccentricy and circularity of the original contour are calculated and quantizied, as well as the ones from the final contour (convex)

$$ypeak[0] = 3.8 * \left(\frac{y\_css[0]}{Nsamples^2} \right)^{0.6}$$

$$ypeak[k] = 3.8 * \left(\frac{y\_css[k]}{Nsamples^2} \right)^{0.6}$$

$$circularity = \frac{perimeter^2}{area}$$

$$eccentricity = \sqrt{\frac{i_{20} + i_{02} + \sqrt{{i_{20}}^2 + {i_{02}}^2 - 2 i_{20} i_{02} + 4 {i_{11}}^2}}{i_{20} + i_{02} - \sqrt{{i_{20}}^2 + {i_{02}}^2 - 2 i_{20} i_{02} + 4 {i_{11}}^2}}}$$

$$i_{02} = \sum_{k=1}^{N} (y_k - y_c)^2$$

$$i_{11} = \sum_{k=1}^{N} (x_k - x_c)(y_k - y_c)$$

$$i_{20} = \sum_{k=1}^{N} (x_k - x_c)^2$$

MPEG-7 Visual Description tools
Shape – Contour shape (IV)

ContourShape {	**Number of bits**	**Mnemonic**
NumOfPeaks	6	uimsbf
GlobalCurvature	2*6	uimsbf
if (NumOfPeaks != 0) {		
PrototypeCurvature	2*6	uimsbf
}		
HighestPeakY	7	uimsbf
for (k=1; k<NumOfPeaks; k++) {		
peakX[k]	6	uimsbf
peakY[k]	3	uimsbf
}		
}		

MPEG-7 Visual Description tools
Shape – Contour shape (V)

Similarity matching

- First the global parameters are compared.

$$\frac{\left|c_q[0]-c_r[0]\right|}{MAX(c_q[0],c_r[0])} \le Th\_e \qquad \frac{\left|c_q[1]-c_r[1]\right|}{MAX(c_q[1],c_r[1])} \le Th\_c$$

- If they are similar the others are also computed

$$M = 0.4 \times \frac{\left|c_q[0]-c_r[0]\right|}{MAX(c_q[0],c_r[0])} + 0.3 \times \frac{\left|c_q[1]-c_r[1]\right|}{MAX(c_q[1],c_r[1])} + Mcss$$

$$Mcss = \sum_i \left((xpeak[i]-xpeak[j])^2 + (ypeak[i]-ypeak[j])^2\right) + \sum_i (ypeak[i])^2$$

  - The first term deal with “matched” peaks (x1-x2 aound 0.1) and the second one penalizes not matched ones.

MPEG-7 Visual Description tools
Shape – Contour shape (VI)

Work proposal 8: Contour Shape (clause 8.3 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the ContourShape D from a set of images in a folder
  - Couple of slides describing the algorithms and the program
    Contour should be firstly obtained (contour images can be used)
  - clause 4.4.2.1 (ISO/IEC TR 15938-8)
- Similarity matching and use cases:
  - Matlab program that selects an 8contour) image from a folder, extracts the ContourShape D and provides a ranked list of most similar (countour) images in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.4.2.2 and 4.4.2.3 (ISO/IEC TR 15938-8)

MPEG-7 Visual Description tools
Shape – Shape 3D (I)

3D Shape: intrinsic shape description of 3D mesh models exploiting some local attributes of the 3D surface

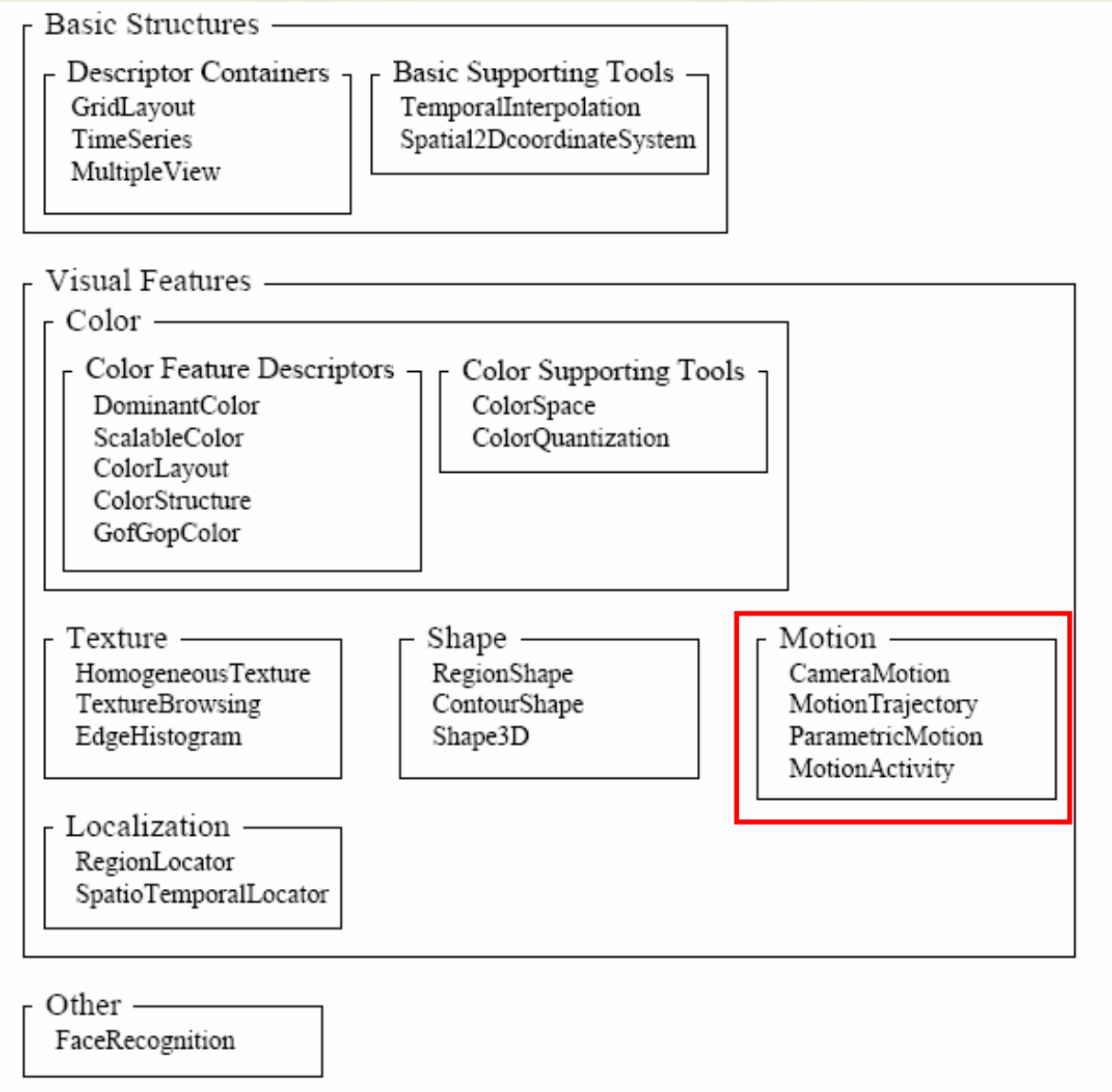
- shape spectrum: histogram of shape indexes
  - shape index: function of the two principal curvatures (Koenderik)

$$I_p = \frac{1}{2} - \frac{1}{\pi}\arctan\frac{k_p^1 + k_p^2}{k_p^1 - k_p^2}, \text{ with } k_p^1 \geq k_p^2.$$

  - Mesh based curvature calculation

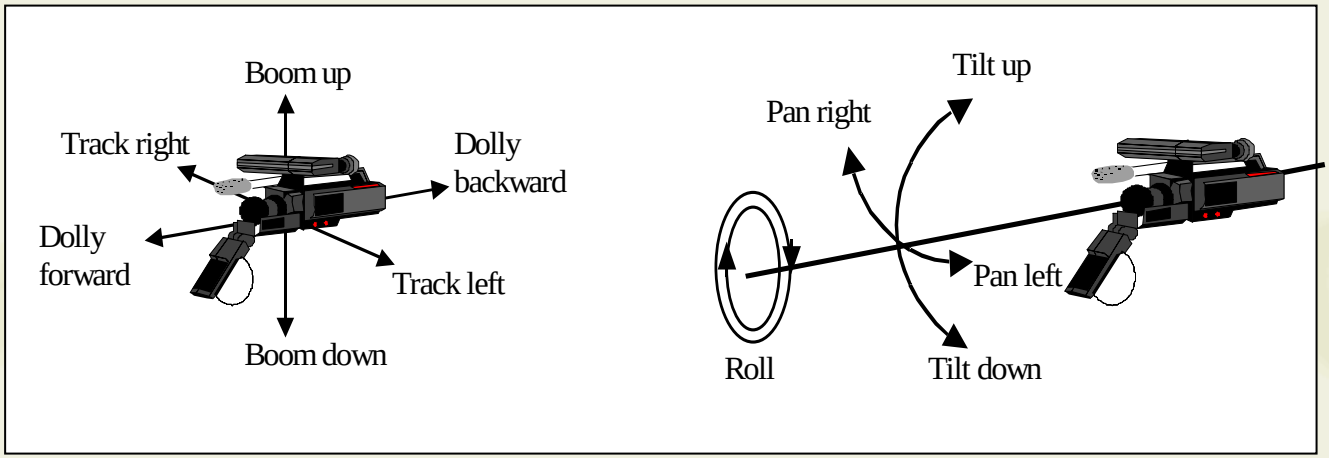
Shape3D {	Number of bits	Mnemonic
NumOfBins	8	uimsbf
bitsPerBin	4	uimsbf
for(k=0; k<NumOfBins; k++) {		
Spectrum[k]	1-12	uimsbf
}		
PlanarSurfaces	1-12	uimsbf
SingularSurfaces	1-12	uimsbf
}		

MPEG-7 Visual Description tools



MPEG-7 Visual Description tools
Motion – Camera Motion

Camera Motion: describes 3D camera motion parameters, plus zoom



- For each segment it indicates the start time and duration and the amount of movement of the different possible parameters
  - Mixed movements can be described
- Parameters can be extracted from a sequence via analysis or annotated during camera operation (near future?)
- Poorly described in TR

MPEG-7 Visual Description tools
Motion – Motion Trajectory

Motion Trajectory: describes the motion trajectory of a moving object via the spatio-temporal localisation of one representative point of the object (e.g., the centroid)

- key points and interpolation (see TemporalInterpolation)

MotionTrajectory {	Number of bits	Mnemonic
cameraFollows	2	bslbf
CoordFlag	1	bslbf
if(CoordFlag) {		
ref		UTF-8
spatialRef	1	bslbf
}		
else {		
units	2	bslbf
CoordCodingLength	1	bslbf
if(CoordCodingLength) {		
xRepr	8	uimsbf
yRepr	8	uimsbf
}		
}		
Params	See subclause 5.6.3	TemporalInterpolationType
}		

- The TR provide some clues for extraction and similarity measures

MPEG-7 Visual Description tools
Motion – Parametric Motion (I)

Parametric Motion: specifies motion of objects as well as global motion

- motion associated to regions of image over a time interval
  - If several feature points it can describe non-rigid objects movements
- parameters of motion models: translation, rotation/scaling, affine, planar perspective, parabolic (quadratic)

Translational model:

$$x' = a_1 + x$$
$$y' = a_2 + y$$

Rotation/Scaling model:

$$x' = a_1 + a_3 x + a_4 y + x$$
$$y' = a_2 - a_4 x + a_3 y + y$$

Affine model:

$$x' = a_1 + a_3 x + a_4 y + x$$
$$y' = a_2 + a_5 x + a_6 y + y$$

Planar perspective model:

$$x' = [(a_1 + a_3 x + a_4 y) / (1 + a_7 x + a_8 y)] + x$$
$$y' = [(a_2 + a_5 x + a_6 y) / (1 + a_7 x + a_8 y)] + y$$

Parabolic model:

$$x' = a_1 + a_3 x + a_4 y + a_7 xy + a_9 x^2 + a_{10} y^2 + x$$
$$y' = a_2 + a_5 x + a_6 y + a_8 xy + a_{11} x^2 + a_{12} y^2 + y$$

Example 1: Translational – 2 parameters

```
<ParametricMotion motionModel = "translational">
   <CoordDef originX = 3.0 originY = 4.0/>
   <MediaDuration> 1340 </MediaDuration>
   <Params> 3.5 2.7 </Params>
</ParametericMotion>
```

Example 2: Rotation/scaling – 4 parameters

```
<ParametricMotion motionModel = "rotationOrScaling">
   <CoordDef originX = 2.0 originY = 4.0/>
   <MediaDuration> 1783 </MediaDuration>
   <Params> 3.5 2.7 6.7 9.6 </Params>
</ParametericMotion>
```

Example 3: Affine – 6 parameters

```
<ParametricMotion motionModel = "affine">
   <CoordDef originX = 3.0 originY = 5.0/>
   <MediaDuration> 3402 </MediaDuration>
   <Params> 3.5 2.7 6.7 9.6 4.5 2.1 </Params>
</ParametericMotion>
```

Example 4: Perspective – 8 parameters

```
<ParametricMotion motionModel = "perspective">
   <CoordDef originX = 3.0 originY = 2.0/>
   <MediaDuration> 2334 </MediaDuration>
   <Params> 3.5 2.7 6.7 9.6 4.5 2.1 3.6 8.5 </Params>
</ParametericMotion>
```

Example 5: Quadratic – 12 parameters

```
<ParametricMotion motionModel = "quadratic">
   <CoordDef originX = 3.0 originY = 6.0/>
   <MediaDuration> 1673 </MediaDuration>
   <Params>   3.5 2.7 6.7 9.6 4.5 2.1 3.6 8.5 1.2 4.3 5.6 7.7</Params>
</ParametericMotion>
```

MPEG-7 Visual Description tools
Motion – Parametric Motion (II)

ParametricMotion {	Number of bits	Mnemonic
motionModel	3	bslbf
CoordFlag	1	bslbf
if(CoordFlag) {		
ref		UTF-8
spatialRef	1	bslbf
} else {		
originX	32	fsbf
originY	32	fsbf
}		
MediaDuration	See annex B	MediaIncrDurationType
for(k=0; k<NumOfParams; k++) {		
Params[k]	32	fsbf
}		
}		

motionModel	NumOfParams	Meaning
000	2	Translational
001	4	Rotation/scaling
010	6	Affine
011	8	Perspective
100	12	Quadratic
101-111	reserved	Reserved

MPEG-7 Visual Description tools
Motion – Motion Activity (I)

Motion Activity: intuitive notion of “intensity of action” in a video segment

- intensity of activity
- direction of activity
- spatial distribution and localisation
- temporal localisation
- Extraction via gross motion characteristics (e.g., motion vectors)
  - avoid object segmentation, tracking etc.
  - See TR

MPEG-7 Visual Description tools
Motion – Motion Activity (II)

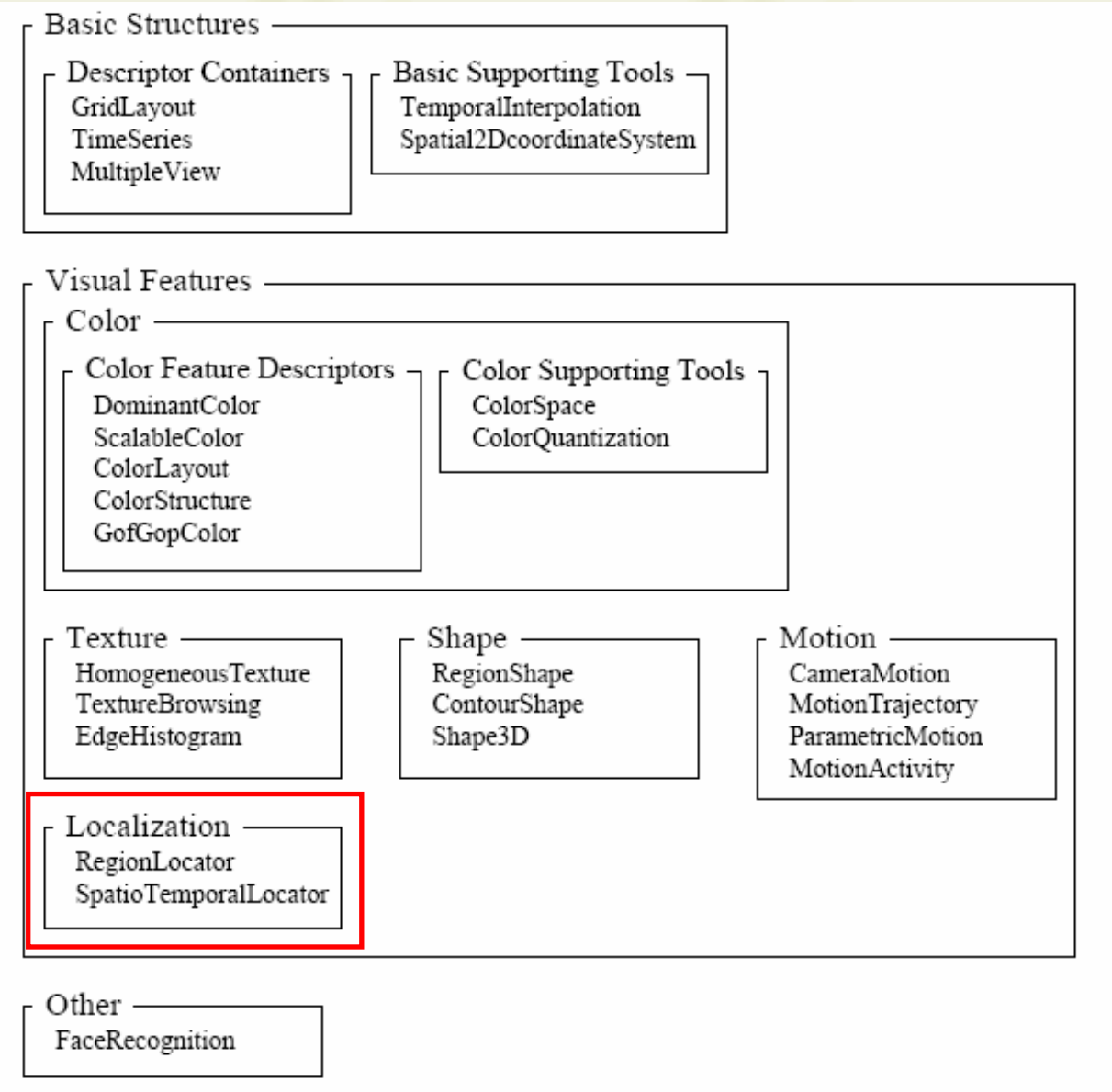
MotionActivity {	Number of bits	Mnemonic
Intensity	3	uimsbf
DirectionFlag	1	bslbf
SpatialDistributionFlag	1	bslbf
SpatialLocalizedDistributionFlag	1	bslbf
TemporalDistributionFlag	1	bslbf
if(DirectionFlag) {		
DominantDirection	3	uimsbf
}		
if(SpatialDistributionFlag) {		
NumOfShortRuns	6	uimsbf
NumOfMediumRuns	5	uimsbf
NumOfLongRuns	5	uimsbf
}		
if(SpatialLocalizedDistributionFlag) {		
SpaLocNumber	2	uimsbf
for(k=0; k<SpaLocNumber; k++) {		
SpatialLocalizationParams	3	uimsbf
}		
}		
if(TemporalDistributionFlag) {		
for(k=0; k<5; k++) {		
TemporalParams[k]	6	uimsbf
}		
}		
}		

MPEG-7 Visual Description tools
Motion – Motion activity (III)

Work proposal 9: Motion activity (clause 9.5 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the Motion activity D from a set of video sequences in a folder
  - Couple of slides describing the algorithms and the program
  - clause 4.5.4.1 (ISO/IEC TR 15938-8)
- Similarity matching and use cases:
  - Matlab program that selects a video from a folder, extracts the Motion Activity D and provides a ranked list of most similar videos in the folder
  - Couple of slides describing the algorithms and the program
  - clause 4.5.4.2 and 4.5.4.3 (ISO/IEC TR 15938-8)

MPEG-7 Visual Description tools

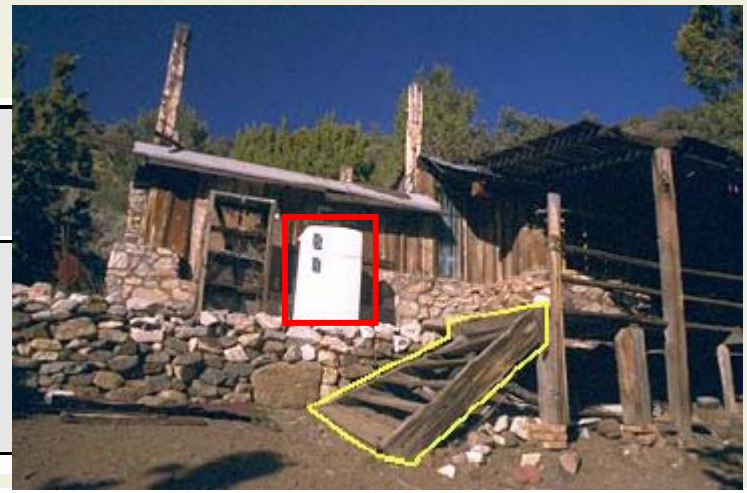


MPEG-7 Visual Description tools
Localization – Region Locator (I)

Region Locator: localisation of regions of images with a brief description of boxes and polygons

- If unlocatedRegion is true the region is excluded

```
<RegionLocator>
   <Polygon unlocatedRegion = "false">
      <Coords dim = "7 2">
         230  50  0  -68  -19  -36  73
         166  -9  21  62  -4  -24  -38
      </Coords>
   </Polygon>
</RegionLocator>
```



```
<RegionLocator>
   <Box unlocatedRegion = "false" dim = "2 2">
         148  185
         114  173
   </Box>
</RegionLocator>
```

```
<RegionLocator>
   <Box unlocatedRegion = "true" dim = "2 2">
         148  185
         114  173
   </Box>
</RegionLocator>
```

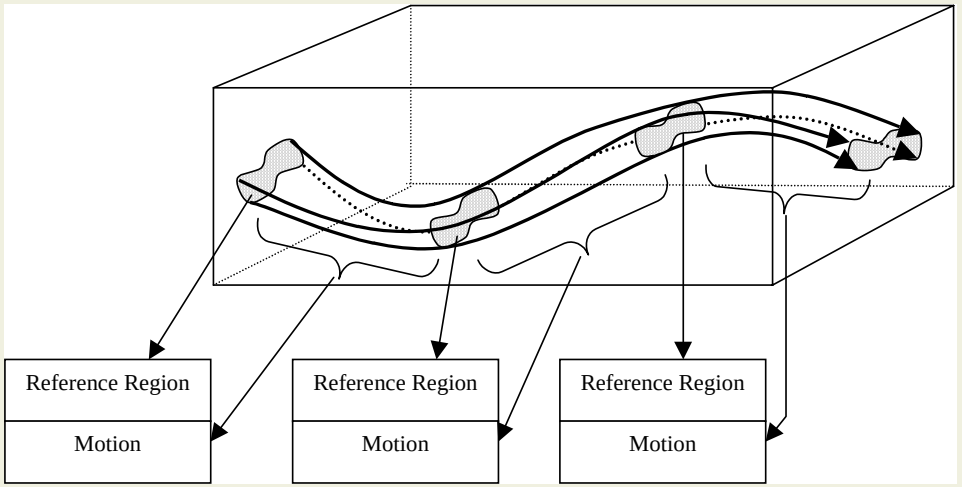
MPEG-7 Visual Description tools
Localization – Region Locator (II)

RegionLocator {	Number of bits	Mnemonic
CoordFlag	1	bslbf
if(CoordFlag) {		
ref	See ISO 10646	UTF-8
spatialRef	1	bslbf
} else {		
XRepr	8	uimsbf
YRepr	8	uimsbf
}		
ContainedLocatorTypes	2	bslbf
if(ContainedLocatorTypes&1) {		
NumOfBoxes		vluimsbf5
for(j=0;j<NumOfBoxes;j++) {		
unlocatedRegion	1	bslbf
Use3P	1	bslbf
for(k=0;k<2+Use3P;k++) {		
PixelX[k]	if(CoordFlag) ceil(ld(xSrcSize)) else XRepr	uimsbf
PixelY[k]	if(CoordFlag) ceil(ld(ySrcSize)) else YRepr	uimsbf
}		
}		
}		
if(ContainedLocatorTypes&2) {		
NumOfPolygons		vluimsbf5
for(j=0;j<NumOfPolygons;j++) {		
unlocatedRegion	1	bslbf
NumOfVertices		vluimsbf5
FirstVertexX	if(CoordFlag) ceil(ld(xSrcSize)) else XRepr	uimsbf
FirstVertexY	if(CoordFlag) ceil(ld(ySrcSize)) else YRepr	uimsbf
XDynamicRange	4	uimsbf
YDynamicRange	4	uimsbf
for(k=0;k<NumOfVertices;k++) {		
Octant	3	bslbf
MajorComponent[k]	XDynamicRange or YDynamicRange	uimsbf
MinorComponent[k]	ld(min(MajorComponent[k], DynamicRange(MinorComponent)))	uimsbf
}		
}		
}		
}		

MPEG-7 Visual Description tools
Localization – Spatio Temporal Locator (I)

Spatio Temporal Locator: specifies the spatio-temporal regions in a video (for hypermedia applications)

- localisation of moving regions
- reference regions and motion description
  - Figure Trajectory (non-rigid object): vertices and motion
  - Parameter Trajectory (rigid object): region and motion



MPEG-7 Visual Description tools
Localization – Spatio Temporal Locator (II)

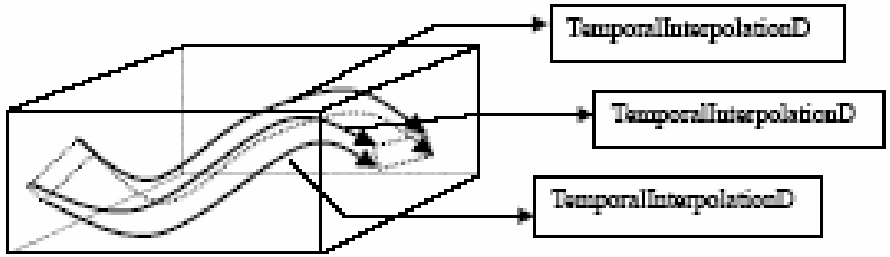


Figure 117 - Motion Descriptions for FigureTrajectory

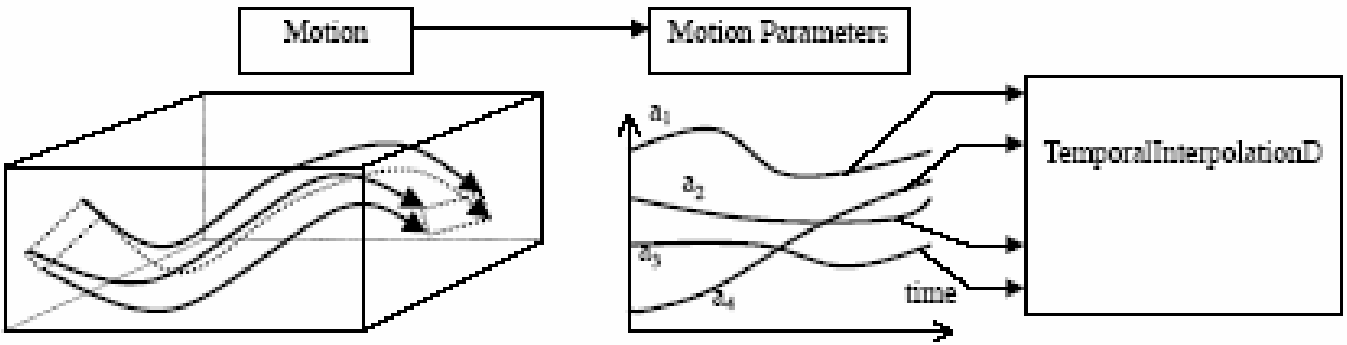


Figure 123 - Motion Descriptions for ParameterTrajectory

MPEG-7 Visual Description tools
Localization – Spatio Temporal Locator (III)

SpatioTemporalLocator {	Number of bits	Mnemonic
CoordFlag	1	bslbf
if(CoordFlag) {		
ref	See ISO 10646	UTF-8
spatialRef	1	bslbf
}		
NumOfRefRegions		vluimsbf5
for(k=0; k<NumOfRefRegions; k++) {		
TypeOfTrajectory	2	bslbf
if(TypeOfTrajectory=="00") {		
FigureTrajectory	See Clause 10.3.5.3	FigureTrajectoryType
} else if(TypeOfTrajectory=="01") {		
ParameterTrajectory	See Clause 10.3.6.3	ParameterTrajectoryType
} else if(TypeOfTrajectory=="10") {		
MediaTime	See annex B	MediaTimeType
}		
}		
}		

MPEG-7 Visual Description tools
Localization – Spatio Temporal Locator (IV)

FigureTrajectory {	Number of bits	Mnemonic
MediaTime	See annex B	MediaTimeType
type	6	uimsbf
for(i=0;i<NumOfVertices;i++) {		
Vertex[i]	see subclause 5.6.3	TemporalInterpolationType
}		
DepthFlag	1	bslbf
if(DepthFlag) {		
Depth	See subclause 5.6.3	TemporalInterpolationType
}		
}		

Table 47 — Semantics of type.

type	Figure	NumOfVertices
0	Forbidden	-
1	Rectangle	3
2	Ellipse	3
3-63	polygon	Value of type

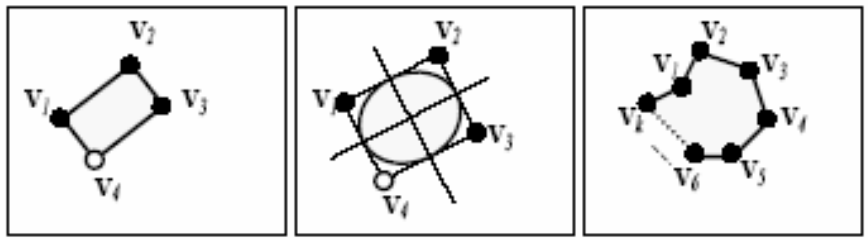


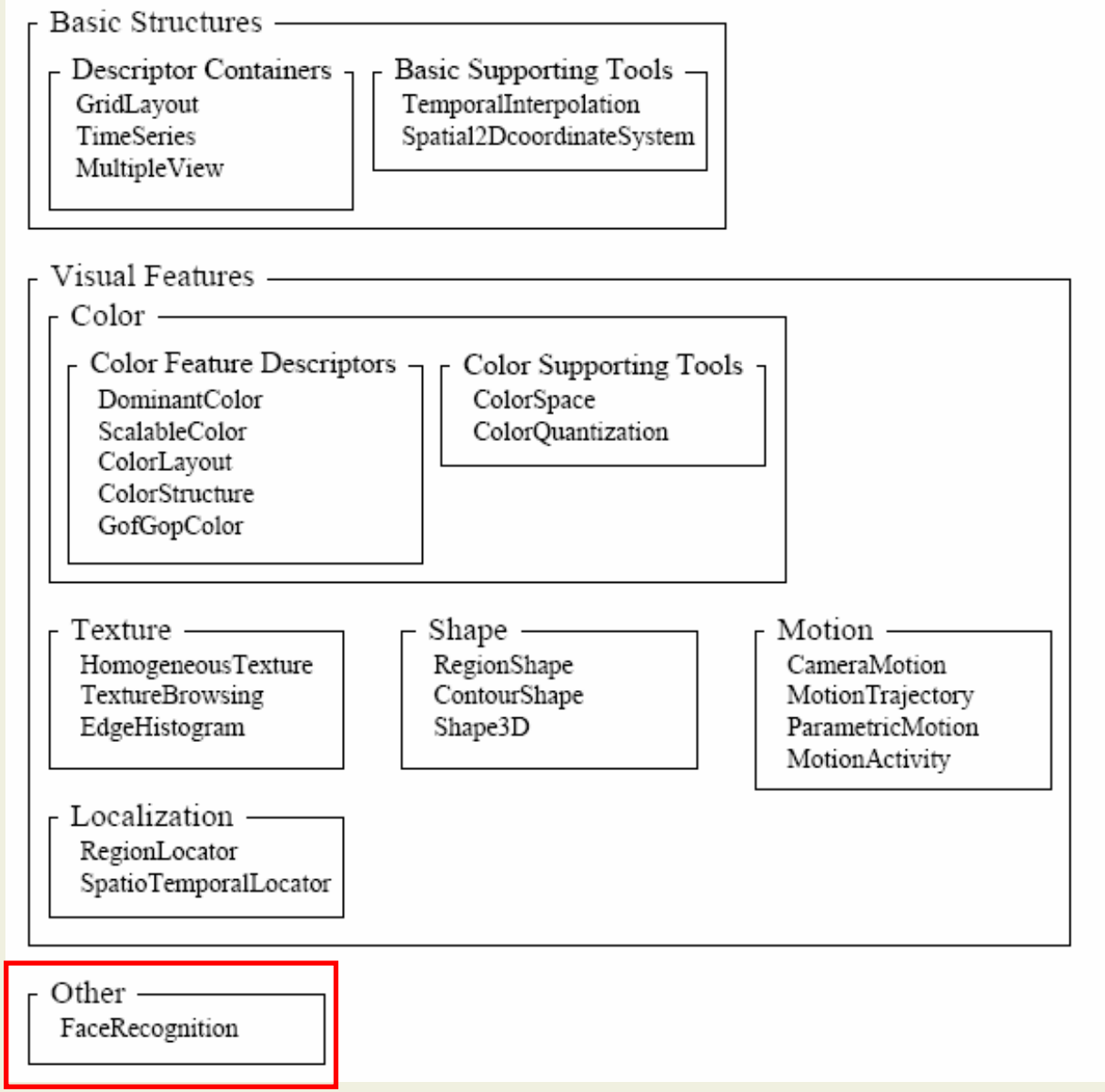
Figure 30 — Representative points of rectangle, ellipse and polygon.

MPEG-7 Visual Description tools
Localization – Spatio Temporal Locator (V)

ParameterTrajectory {	Number of bits	Mnemonic
motionModel	3	uimsbf
ellipseFlag	1	bslbf
MediaTime	See annex B	MediaTimeType
InitialRegion	See subclause 10.2.3	RegionLocatorType
Params	see subclause 5.6.3	TemporalInterpolationType
DepthFlag	1	bslbf
if(DepthFlag) {		
Depth	see subclause 5.6.3	TemporalInterpolationType
}		
}		

MotionModel	Parametric motion model	Number of parameters
0	still	0
1	translation	2
2	rotationAndScaling	4
3	affine	6
4	perspective	8
5	parabolic	12
6-7	reserved	n/a

MPEG-7 Visual Description tools



MPEG-7 Visual Description tools
Others – Face Recognition (I)

Face Recognition: specifies the projection of a face vector into 48 basis vectors.

- Face vector: raster scan of a normalized face image
  - 46 rows x 56 lines
  - Center of eyes in points (16,24) and (31,24)
- The descriptor features are 48 projections of the face vector onto a space defined by matrix U (Appendixes of ISO/IEC 15938-3)
  - The features normalized and clipped, and represented with 5 bits

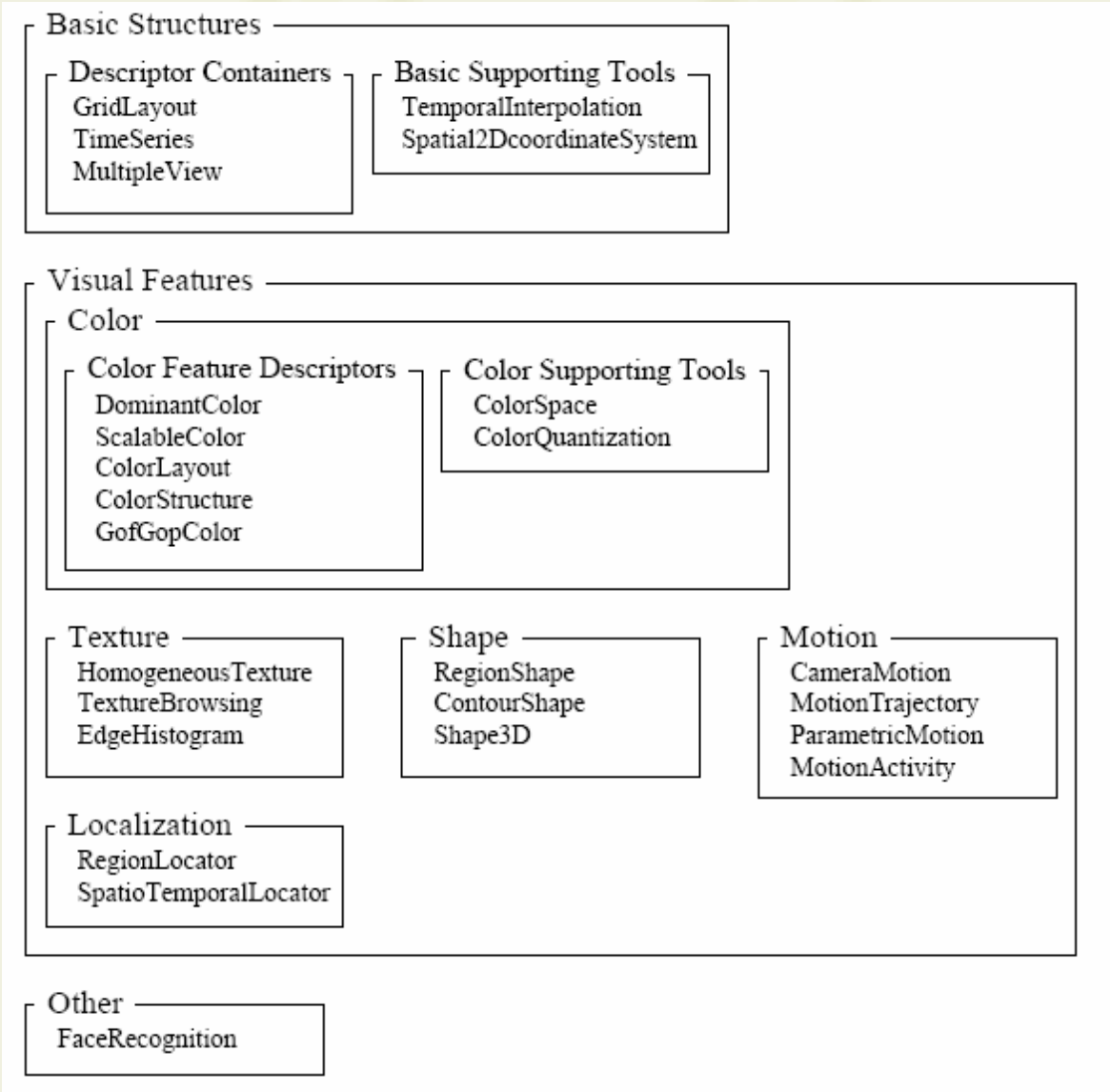
FaceRecognition {	Number of bits	Mnemonic
for(k=0; k<48; k++)		
Feature[k]	5	uimsbf
}		

MPEG-7 Visual Description tools
Others – Face Recognition (II)

Work proposal 10: Face recognition (clause 11.2 ISO/IEC FDIS 15938-3)

- Feature extraction:
  - Matlab program that extracts the Face recognition D from a set of face images in a folder
  - Couple of slides describing the algorithms and the program
  - clause 7.1.1.1 (ISO/IEC TR 15938-8)
- Similarity matching:
  - Matlab program that selects a face image from a folder, extracts the Face Recognition D and provides a ranked list of most similar face images in the folder
  - Couple of slides describing the algorithms and the program
  - clause 7.1.1.2 (ISO/IEC TR 15938-8)

MPEG-7 Visual Description tools



MPEG-7 Audio Description tools

Basic Audio description tools

Spoken Content

Timbre

Sound Effects

Melody

MPEG-7 Audio Description tools
Work proposals

Work proposal 11: Audio Framework (clause 5 ISO/IEC FDIS 15938-4)

Work proposal 12: Timbre (clause 6.3 ISO/IEC FDIS 15938-4)

Work proposal 13: General Sound Recognition and Indexing (clause 6.4 ISO/IEC FDIS 15938-4)

Work proposal 14: Spoken Content (clause 6.5 ISO/IEC FDIS 15938-4)

Work proposal 15: Melody (clause 6.6 ISO/IEC FDIS 15938-4)

- Description (some slides) of the correspoding clauses
- Optionally
  - Feature extraction:
    - Matlab program that extracts the corresponding Ds
    - Couple of slides describing the algorithms and the program
  - Similarity matching:
    - Matlab program that selects an audio file from a folder, extracts the corresponding D and provides a ranked list of most similar audio files in the folder
    - Couple of slides describing the algorithms and the program

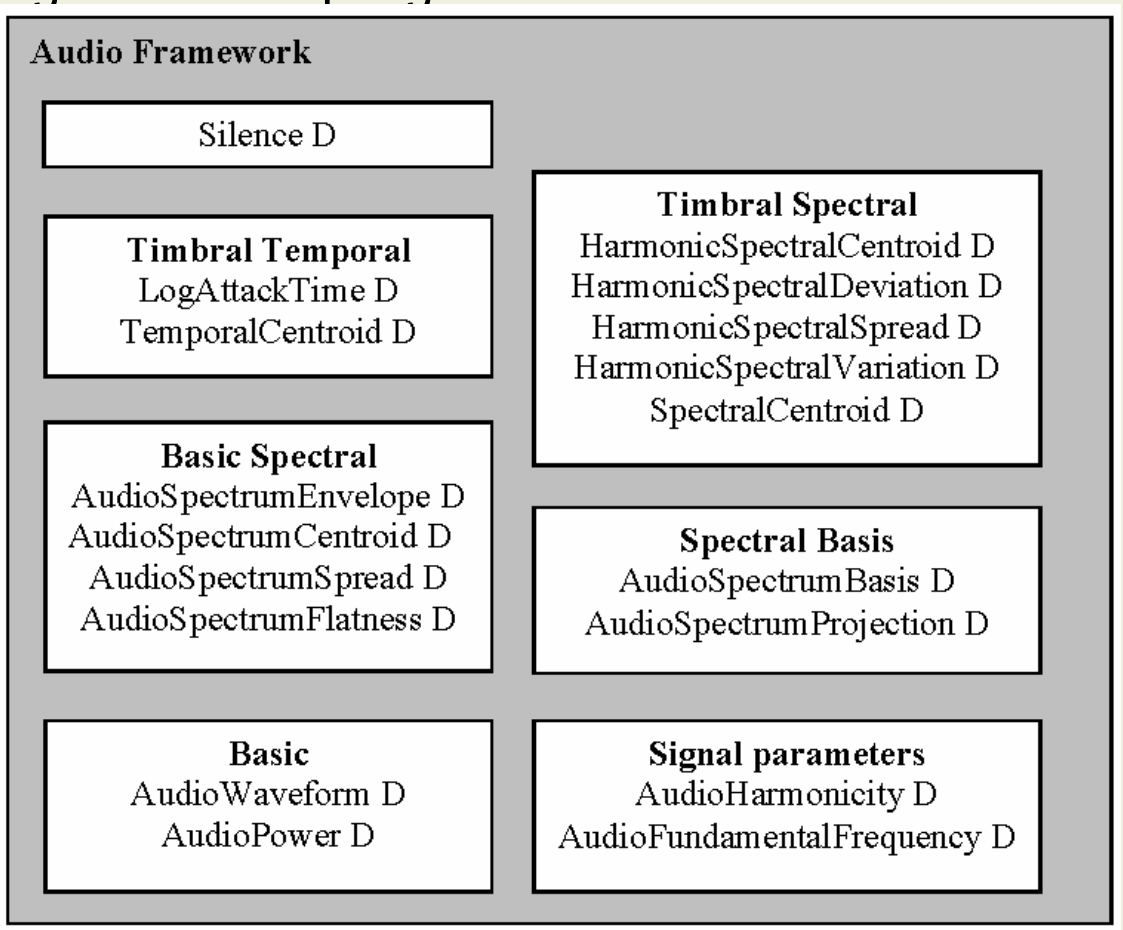
MPEG-7 Audio Description Tools
Basic Audio Description Tools

Scalable Series.

- An efficient representation for series of feature values (scalars and vectors) supporting down-sampling.

Audio Framework.

- A collection of low-le



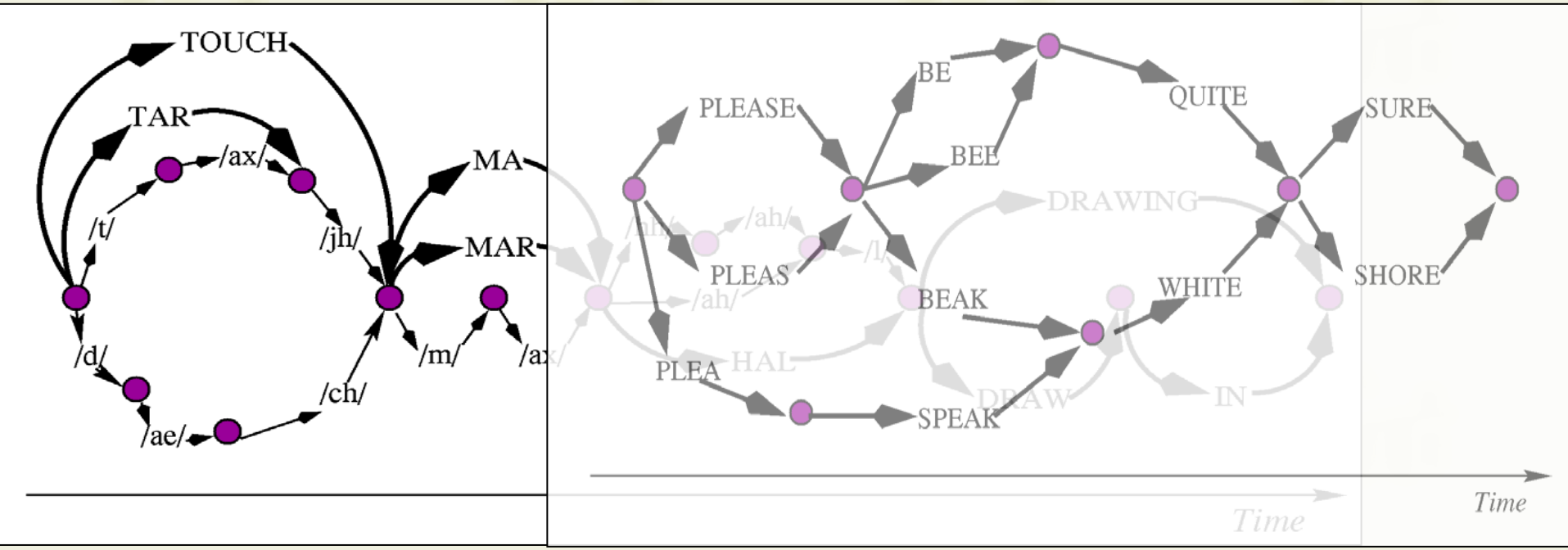
MPEG-7 Audio Description Tools
Spoken Content

Spoken Content Header

- Speaker and ASR Information
- Word Lexicon, Phone Lexicon

Spoken Content Lattice

- "Timed" blocks of nodes
- Nodes and links
  - words and phones



MPEG-7 Audio Description Tools
Timbre

Timbre

- perceptual features that make two sounds having the same pitch and loudness sound different
  - Harmonic Spectral Centroid (hsc)
  - Harmonic Spectral Deviation (hsd)
  - Harmonic Spectral Spread (hss)
  - Harmonic Spectral Variation (hsv)
  - Spectral Centroid (sc)
  - Temporal Centroid (tc)
- Instrument Timbre
  - Harmonic and Percussive

MPEG-7 Audio Description Tools
Sound Effects

Sound Effects Category

- taxonomy of sound effects

Sound Effects Features

- RMSEnergyEnvelope
- SpectrumBasisProjectionEnvelope
- Additional AudioDescriptors

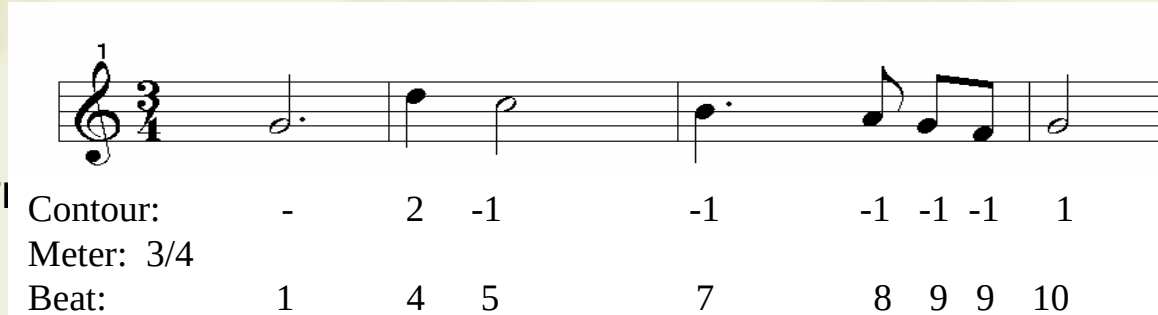
Sound Effect Classifier

- Sound Effect Model
  - Sound Effect Category Reference
  - Extraction Information
  - Probability Model
  - Spectrum Basis

MPEG-7 Audio Description Tools
Melody

Melody Contour

- Contour
  - 5-level contour infor
- Meter
  - time signature (rhythmic information)
- Beat
  - nearest whole-beat of each note of a melody



Melody

- Meter
- Scale
- Key Mode (major, minor)
- Melody Sequences

Audiovisual Features for Indexing

- Spatio-temporal structure features
- Low-level features
- Mid-level features
- High-level features

Mid-level features

Mid-level features can be defined as “semantic” generic features that can be “inferred” from low-level features

- Objects detection (e.g., homogeneous -moving- region)
- Persons detection (e.g., skin color)
- Location detection (e.g., outdoors-indoors, day-night)
- Event detection (e.g., running people)

Some recognition could also be considered mid-level

- Objects recognition (e.g., ball, vegetation)
- Persons recognition (e.g., male-female, young-old)
- Location recognition (e.g., sea, sky)
- Event recognition (e.g., explosion)

Audiovisual Features for Indexing

- Spatio-temporal structure features
- Low-level features
- Mid-level features
- High-level features

High-level features

High-level features can be defined as the “semantic” understanding of the reality depicted in the analysed content

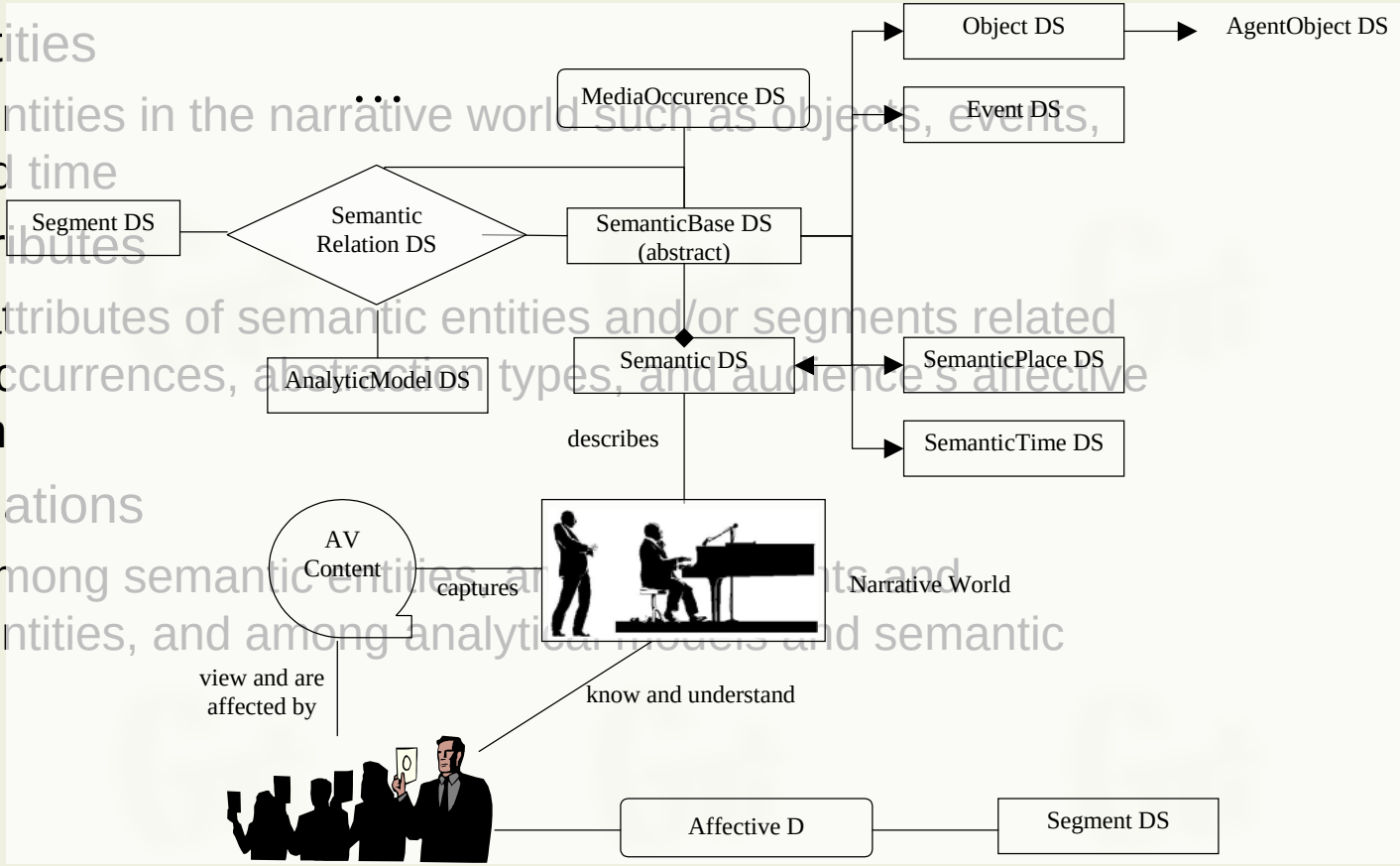
- Objects recognition (e.g., tree)
- Persons recognition (e.g., identity or role)
- Location recognition (specific)
- Event recognition (e.g., goal)

For extraction/generation the context is a main help

Semantic Description: the MPEG-7 proposal

Semantic of Content

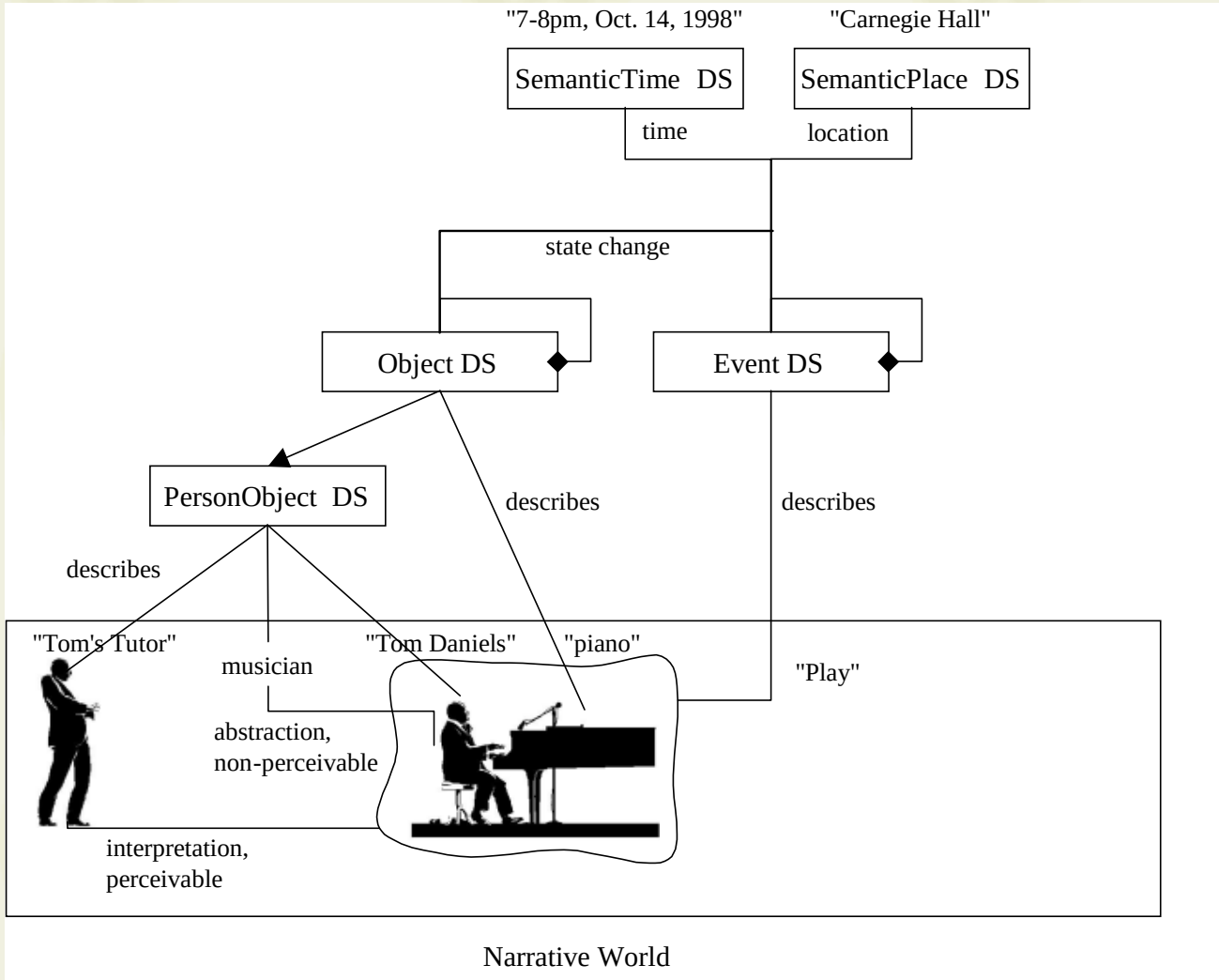
- Semantic Entities
  - o semantic entities in the narrative world such as objects, events, places, and time
- Semantic Attributes
  - o semantic attributes of semantic entities and/or segments related to media occurrences, abstraction types, and audience's affective information
- Semantic Relations
  - o relations among semantic entities, and among events and semantic entities, and among analytical models and semantic entities



Semantic Description: the MPEG-7 proposal

Semantic Base

- UsageLabel
- Label
- Definition
- Property
- MediaOccurrence
- Relation



AV features for indexing: Conclusions

Low-level features are “easy” to obtain but insufficient for “useful” retrieval (from a user point of view)

- Computational load for extraction
- Compactness of description (efficient storage)
- Efficiency in retrieval (precision and recall) both for searching and filtering
- Scalability
- Granularity of indexing (objects/segments versus item)

Mid-level and High-level features describe semantic concepts but are difficult to obtain from low-level features

- Restricted context

Bridging the semantic gap!!

Indexing of AV Content

Contents

Bridging the semantic gap

"Bridging the semantic gap" refers to the inference of higher level descriptions (metadata) from low-level audiovisual descriptions

- Most research is based on MPEG-7 low-level descriptors …
  - Dominant Color, Structured Color
  - Homogeneous texture
  - Shape
  - Motion
- … calculated after spatio-temporal segmentation yielding to "homogeneous" regions/segments.
- Adding context constraints some inference is possible
  - Additionally domain ontologies are good for "normalizing" the restrictions and results

Towards high-level descriptions

The first step is spatio-temporal segmentation

- First temporal segmentation in homogeneous segments (i.e., shots)
- Afterwards spatial segmentation and object tracking
  - Object tracking (including structure discovery) may help to refine the segmentation

The second step is obtaining the low-level audiovisual features descriptions

The third step is to reconsider the segmentation obtained in the first step, merging or splitting regions based on the analysis (coherence, distance, smoothing, …) of the regions annotated with the low-level descriptions

- Fine Classification: neural networks, HMM, genetic algorithms, SVM, distance based tree simplifications, …

Towards learning semantics

Once the low-level and syntactic (structure) descriptions have been refined it is time to take into consideration context

- Since some time, all research has been focused to context dependent inference

Observing the context some (all) relevant concepts are selected

- Ontologies (thesauri, vocabularies, …) are here of “great” help

And for each concept the “canonical” low-level descriptions and structural relationships are defined

Mapping concepts to low-level descriptions

Concepts are analyzed in order to obtain a “canonical” mapping (context dependent) with low-level descriptions

- Concept
  - Grass
- Low-level descriptions
  - Color: green –maybe yellow-
  - Texture: generally statistically (low resolution of images), but may be “vertical”
  - Movement: generally static (may depend on wind ;)
  - Shape: does not apply (textural region)
- Syntactic/Structural (spatial –and maybe temporal-) relationship
  - Below sky
  - Adjacent to sand and rocks
  - People may be included in grass region

(trying to) Inference

Once all concepts are mapped the system needs to compute all the possibilities and labels the different regions

- Regions may be labelled unknown
- It is usually iterative

Inference …

- RDF/OWL inference engines

… can be done

- at the “audiovisual mapped concept” and spatial (temporal) relationship level
- at the concept and relationship level, and if incoherencies revisit the “audiovisual mapped concepts”

After knowledge inference (semantic annotation) some regions may be merged (exceptionally splitted)

Example

Voisine et al. (WIAMIS'05)

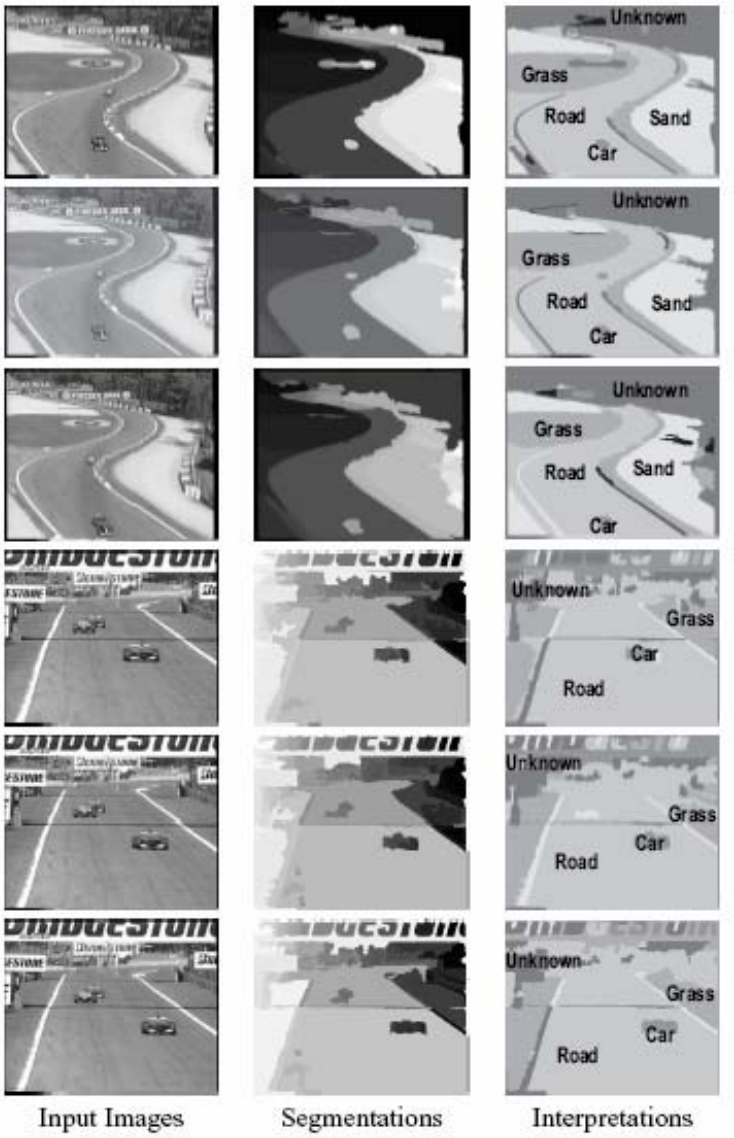


Fig. 1. Formula One domain results

Concept	Visual models	Spatial relations
Road	$DC^1_{road} \vee DC^2_{road} \vee DC^3_{road}$	Road ADJ Grass,Sand
Car	$MOV^1_{car} \wedge CPS^1_{car}$	Car INC Road
Sand	$DC^1_{sand} \vee DC^2_{sand}$	Sand ADJ Grass, Road
Grass	$DC^1_{grass} \vee DC^2_{grass} \vee DC^3_{grass}$	Grass ADJ Road,Sand
Field	$DC^1_{field} \vee DC^2_{field} \vee DC^3_{field}$	Field ADJ Wall
Player	MOV^1_{player}	Player INC Field
Line	$DC^1_{line} \wedge CPS^1_{line}$	Line INC Field
Ball	$DC^1_{ball} \wedge CPS^1_{ball}$	Ball INC Field
Wall	$DC^1_{wall} \vee DC^2_{wall} \vee DC^2_{wall}$	Wall ADJ Field

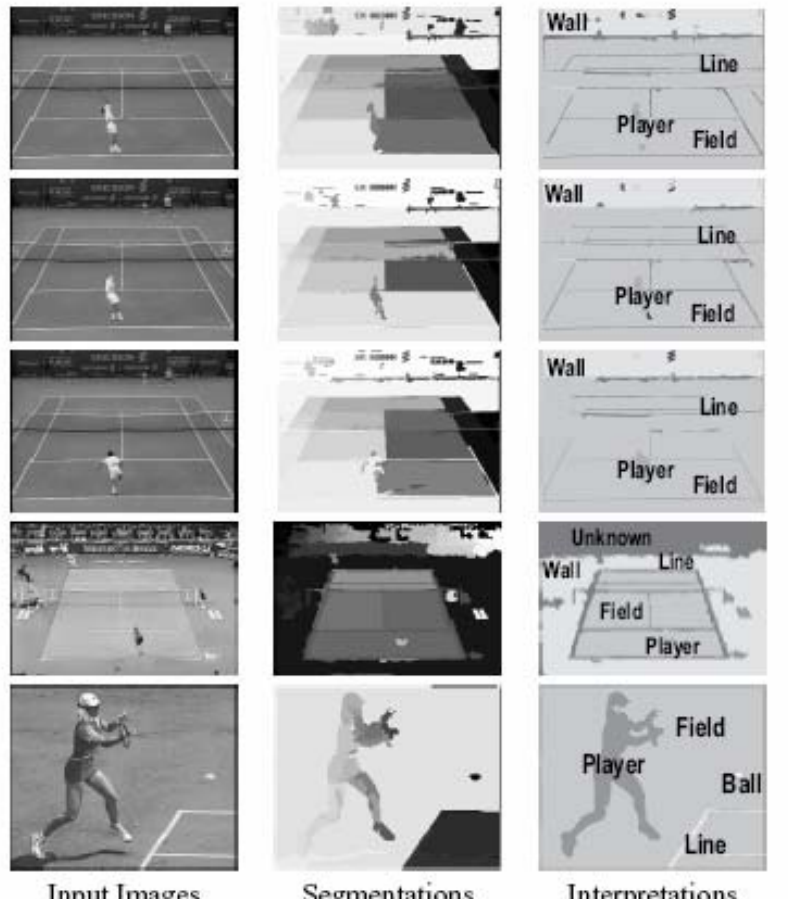


Fig. 2. Tennis domain results

Indexing of AV Content

Contents

Collections of AV content

Besides the description of the contents sometimes it makes sense to group content into “collections”

- Albums: grouping different content into “user driven” groups
  - Mainly Media Classification (location, date, genre, …) and Semantic driven (events, objects, persons, …)
- Clusters: grouping different content into “analysis driven” groups
  - Searching first at cluster level and afterwards at content level
  - Description not only of content but of the collection

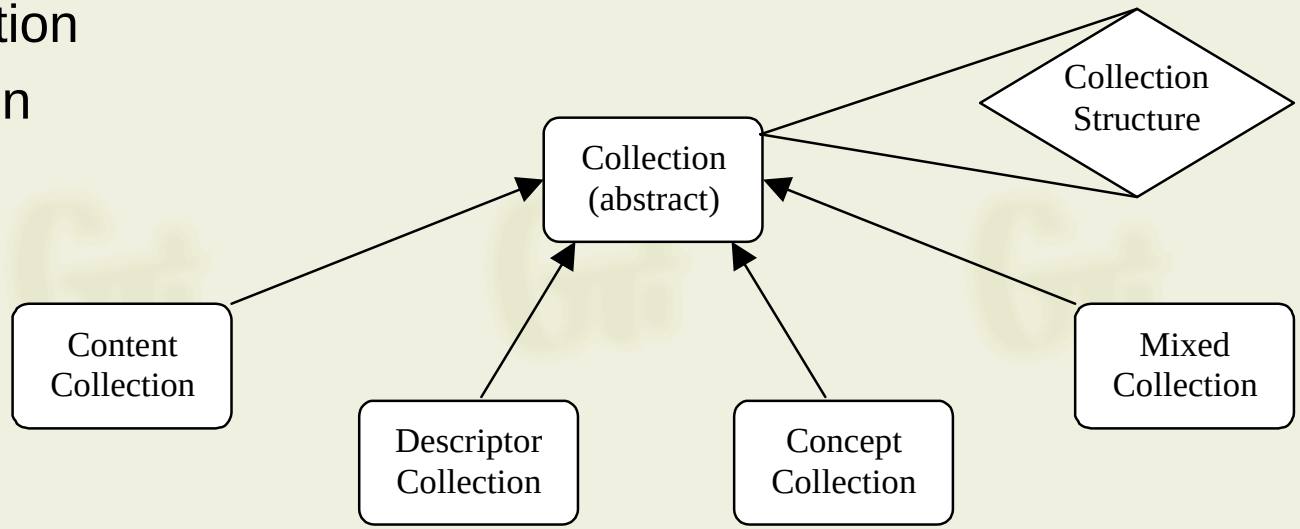
    Model of the description of the cluster

Content may be part of different collections

Collections: the MPEG-7 proposal

Collection

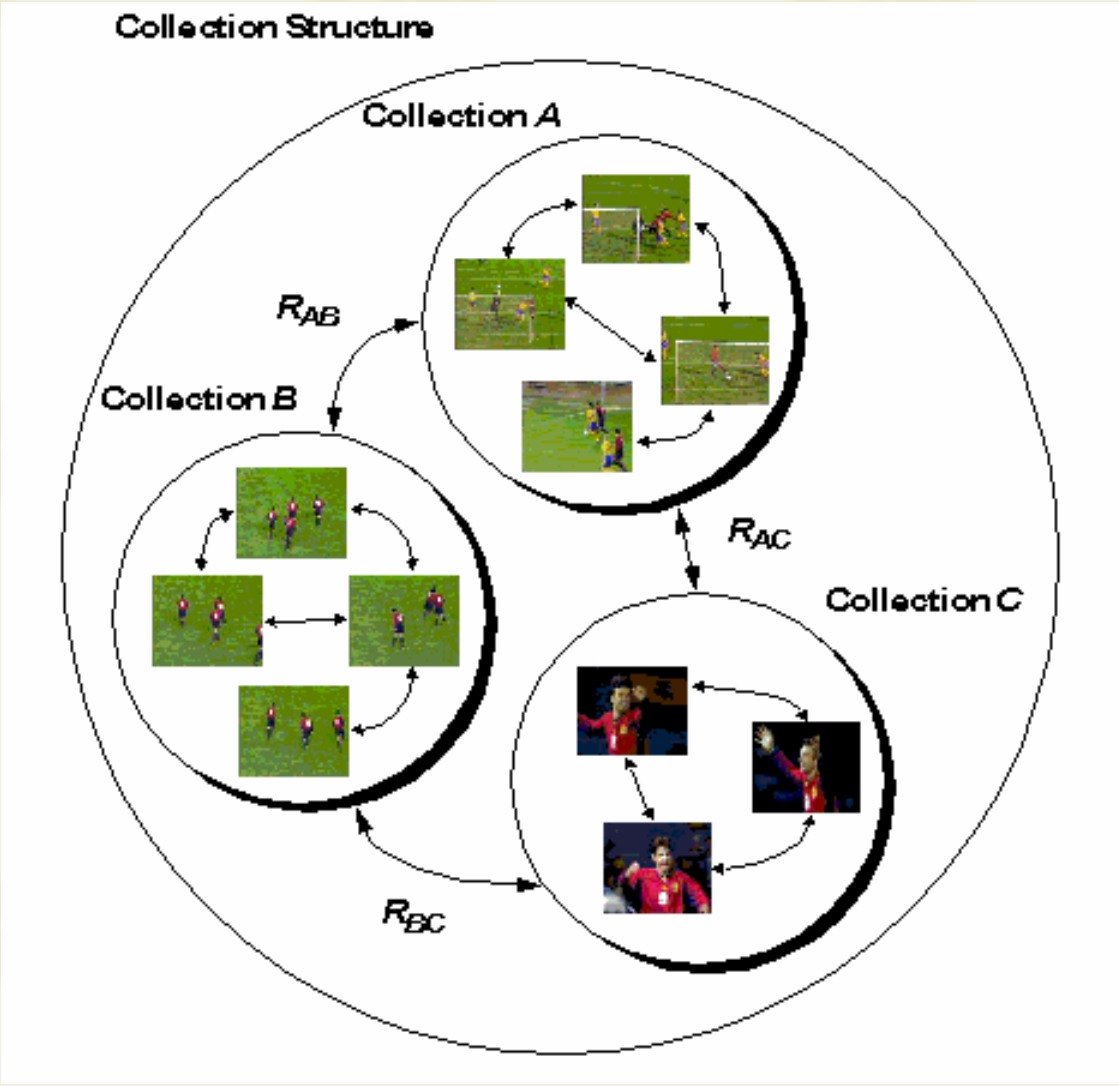
- CreationInformation
- UsageInformation
- TextAnnotation
- Summarization
- Collection
- CollectionRef
- name



Collections: the MPEG-7 proposal

Collection Structure

- Collection
- CollectionRef
- CollectionModel
- CollectionModelRef
- ClusterModel
- ClusterModelRef
- Graph (Relationships)



Indexing of AV Content

Contents

Metrics between descriptors and relevance feedback

Similarity Retrieval (versus Retrieval by Matching)

- More than one result
- Different parts of an image (video, audio, …) may be of interest
  - And it is not possible to index everything at “infinite” granularity
- User confidence in its own specification (query)
  - “Find images looking like this”
  - Use results as further “query by example” (using the media and the associated metadata)
- Results ranked by similarity
  - N-dimensional feature space and distance (Euclidean, Mahalanobis, …)
  - Weighting (personalized) of different features for the overall criterion

Indexing of AV Content

Contents

Conclusions

You have the floor!!!