MAC-REALM: A Video Content Feature Extraction and Modelling Framework

MINAZ PARMAR and MARIOS C. ANGELIDES*
Department of Electronic and Computer Engineering
College of Engineering, Design and Physical Sciences
Brunel University
Uxbridge UB8 3PH
*Corresponding author: (marios.angelides@brunel.ac.uk)

This paper discusses MAC-REALM, a framework for extraction of syntactic and semantic content features and content modelling with either little or no user interaction. The framework integrates a four filter-plane strategy: a pre-processing plane that filters redundant data, a syntactic feature extraction plane that filters syntactic features, a semantic relationships analysis and linkage plane that filters the spatial and temporal relationships of content features, and finally a content modelling plane where the syntactic and semantic content features are integrated into a content model. Each of the four planes is split into three layers: the content layer, where the content to be processed is stored, the application layer, where the content is converted into content descriptions, and the MPEG-7 layer, where content descriptions are serialized. Using MPEG-7 standards to produce the content model will provide wide-ranging interoperability, while facilitating granular multi-content type searches. MAC-REALM aims at 'bridging' the semantic gap, by integrating the syntactic and semantic content features from extraction through to modelling.

1. INTRODUCTION

The explosion of multimedia content on the Internet and in digital archives over the last decade has led to a striking increase in data volume being transferred and stored [Vijayakumar and Nedunchezhian 2012]. The increase in data has lead to the need for better methods for processing and storage of content [Apache 2013; Dropbox 2013; Microsoft 2013; SugarSync 2013]. Data can also be stored in a semantically rich way that allows for better links to be made between information stored in the content [Chiarcos et al. 2012; Mika and Greaves 2012]. The film industry amongst others (i.e. gaming industry) has made extensive use of multimedia content for their businesses [Fromme and Unger 2012; Tryon 2012], including Internet and mobile streaming services [Lawrence et al. 2012; Sarmiento and Lopez 2012]. Multimedia content may convey shots, scenes, people and objects as well as low-level information such as structural and signal level descriptions [Dal Mutto et al. 2012]. One main disadvantage to such abundance of semantic information available within the data is that it is largely ignored. Other metadata apart from the semantics within the content that is largely ignored is, for example, how the content was created and what formats is available in. Typically, methods of searching require a granular description of the content in order to fully utilize semantic meanings within the media. There have been research endeavours to improve the representation and querying of multimedia content [Moens et al. 2012; Weiming et al. 2011]. Google's image search is one such endeavour that can now search for images, using an image as search criteria. However, research pertaining to video searching using similar methods are still not as readily available and is an on-going area of interest [Mezaris et al. 2009].

A content model can facilitate the automatic extraction of content semantics and the intricacies pertaining to multimedia interpretation (Garg and Ramsay, 2011) and it can also allow content producers/consumers to effectively query and retrieve content [Weiming, Nianhua, Li, Xianglin and Maybank 2011]. Automatic content representation is an implicit requirement from the combination of the increase in the amount of content being generated, and the wealth of information stored in multimedia content itself [Lavee et al. 2009; Moens, Poulisse and VRT 2012]. The focus of this paper is the merging of content models with automatic feature extraction into the MAC-REALM cross-functional framework. MAC-REALM uses automatic feature extraction techniques on video content and models them into a hierarchically linked scheme. The automatically extracted features are analysed and semantic relationships derived, to allow the user to query relationships between entities in the multimedia content effectively. The extracted features are also structurally and conceptually linked together to provide a richly descriptive, granular and standardised (MPEG-7) content model, allowing users to view the content from multi-faceted perspectives. The rest of the paper is organized as follows: section 2 offers a related research review, section 3 presents the MAC-REALM framework, section 4 undertakes a performance evaluation and section 5 concludes.

2. LITERATURE REVIEW

In this section we will review those features that are at the core of describing a video stream and need to be modelled in a content model as they cover the majority of features that are generally queried by most video content search applications. The content model provides a content description "proxy" of the content contained within a video stream, and indexes the content to recreate the visually salient points of the content that would be of interest to users to formulate queries with. Content models represent the content of a video stream in a complete and detailed manner. The content features must be described in both structurally syntactic and a semantically meaningful terms, concisely and comprehensively (Moens et al., 2012). The types of descriptors and granularity of the description scheme have direct impact on the usefulness of the content model to different domains and consumers. This leads to the issue of interoperability of the content model across multiple platforms, applications, vendor and propriety independence (Haslhofer and Klas, 2010). To achieve this, the syntactic and semantic content descriptions that make up the model must be integrated so that a symbiosis of structure and concepts within the content become manifest. This is referred to as the semantic gap, i.e. the difference between what a user perceives as the meaning of the content (semantics) to what can be extracted using machine based indexing methods (syntactic) (Küçük and Yazıcı, 2011).

The content features that are modelled must include all levels of the content feature hierarchy. The content features can be categorised into groups depending on their structural and/or conceptual attributes. [Baştan et al. 2010] state that user queries could be categorised into four categories, but also stated this list was not exhaustive. Related works [Angelides 2003; Inigo and Suresh 2012; Lavee, Rivlin and Rudzsky 2009; Mezaris, Papadopoulos, Briassouli, Kompatsiaris and Strintzis 2009; Moens, Poulisse and VRT 2012; Ren et al. 2009; Smeaton et al. 2010; Snoek and Worring 2009; Weiming, Nianhua, Li, Xianglin and Maybank 2011; Zhu and Guo 2012] have categorised all the query types based on what type of content the query addressed. The core content features proposed by [Angelides 2003] contain four content feature classes, spatiotemporal objects, spatial relationships, event segments and temporal relationships. However, these four basic categories only cover the mid-level, high-level and semantic relationships requirement for content based video querying. Another category that is worth consideration is that of low level syntactic features or temporal segments. The difference between a temporal and an event segment is that a temporal segment action may not have a substantive semantic meaning of its own e.g. it may just be a simple action of camera movement such as a pan shot, whereas an event segment has a definitive semantic meaning to it. An event segment could comprise of many temporal segments, who's individual actions add up to an event. Conversely a temporal segment on its own could have a definitive semantic occurrence, e.g. car crashing, but this does not necessarily mean that it is an event as there could have been other actions that complete the event, e.g. tyre blows out.

2.1 Raw Media

Before content feature extraction can take place the raw media, in most cases, needs to be pre-processed to optimise the effectiveness and/or efficiency of the extraction process. Typically, the pre-processing is a filtering step to remove artefacts that could cause errors in the extraction process [Chen et al. 2010; Yongquan et al. 2009], and can be used to reduce the time or complexity of processing the features [Amiri and Fathy 2011; Chan and Wong 2011; Li et al. 2010]. This is usually a case of normalising, converting or filtering the media in the pre-processing step. Flattening is a pre-processing step that can help in making the syntactic feature extraction processes become more accurate. In [Yongquan, Weili and Shaohui 2009] there are many different grey scale levels within a real time scene image. A correlation window is used to compute the grey mean of pixels across the image. To avoid a complex over segmentation of the image, Yongquan et al smooth the regions using a temporal correlation window that samples the different grey scale values and use a 3x3 median filter to normalise values of adjacent pixels, if they are within a certain range. This reduces the region into candidate areas that are likely to be foreground and background regions. In [Chen, Deng, Guo, Wang, Zou and Wang 2010] they transform the colour space profile from the H.264 YUV colour space, into a more humanly perceptible HSV colour space.

One method to reduce the amount of information needing to be processed is by removing redundant data. In [Amri and Fathy 2010] the sampling frame rate of the video sequence is reduced by a several factors before the extraction process. This was shown that it was adequate for video clips with no fast action sequences. The method employed reduces the high computational cost for processing higher frame rates. In [Chan and Wong 2011] they use a sampling rate of one frame per half second, or 2 frames per second (fps) for the pre-processing step. They use this sampling strategy since it assumes that for most domains, shot lengths are longer than 15 frames or half a second. In [Chan and Wong 2011] they go on to use Edge Change Ratio ECR to perform a first-pass on the video for optimal performance

of the algorithm, and generate metrics for the evaluation of the genetic algorithm GA fitness function. Another way of reducing redundancy is to remove the amount of frames by applying a simplified data technique. Before shot extraction can begin, [Li, Ding, Shi and Li 2010] reduce the number of shot candidate frames by using a block colour histogram difference. This method is highly effective as a computational efficiency tool, as the shot boundaries included in film programs typically amount to less than 1% of total frames; thus it is inefficient and extremely time consuming to apply boundary detection processing to detect all the frames. Newer codecs such as H.264 are better suited to video feature extraction as they have advanced features such as motion vector encoding that can be for syntactic feature extraction. In the work by [Fei and Zhu 2010] they segment objects based on motion vectors (MV) directly from H.264 formatted media. They still have to temporally normalise the raw MV to provide a uniform sample, which is ready for the segmentation process. In [Zajić et al. 2011] they convert from DIVX format to uncompressed AVI format which provides more uncompressed frames and is a more suitable feature extraction as there are more frames in every frame sequences to process. This improves the precision of the extraction process.

2.2 Syntactic Content Features

Syntactic feature extraction involves segmentation of a video signal into its constituent parts. These parts represent the different physical aspects of video content that are directly discernible from viewing the video. Each aspect has its own unique physical attributes that describe a certain physical feature of the content that is of interest to a consumer. Such techniques can be grouped into three categories: pixel based, object based and logic based [Lavee, Rivlin and Rudzsky 2009]. Pixel based techniques are generally used for temporal segmentation, and employ the processing of colour, texture, or gradient information in the content. Object based techniques are those that identify features that are the basis for description of semantic items, such as object detection and tracking or face recognition. Object based events aggregate edges, colour and textures into recognisable items. Logic based techniques are the observation that the world is not described by multi-dimensional parameterizations of pixel distributions, or even a set of semantic objects and their properties, but rather by a set of semantic rules and concepts, which act upon units of knowledge. Thus it aims to abstract low-level input into statements of semantic knowledge (i.e. assertions) that can be reasoned on by a rule based event model. Both logic and, to a lesser extent, object based techniques can be described as mid-level features. The major challenge for content based retrieval is to bridge the gap between the low level syntactic features and high level semantic features [Huang and Tung 2010].

There are two categories of content based features that can be analysed in syntactic feature extraction: global features extracted from a whole image and the local or regional features describing the chosen patches of a given image [Harikrishna et al. 2011]. Each region is then processed to extract a set of features characterizing the visual properties including the colour, texture, motion and structure of the region. The shot-based features and the object-based features are the two approaches used to access the video sources in the database. We will begin examining the syntactical structure by looking at the basic foundation of temporal segmentation, and look at the role it plays in deriving semantic features. This will be followed by a review of spatial segmentation techniques and their importance in starting a semantic narrative of the content. Finally, we will look at how the semantic gap, in terms of human perspective, has an influence on temporal segmentation when segmenting into hierarchical components.

2.2.1 Syntactic temporal segmentation

Video can be thought of as a hierarchical syntactical structure comprised of scenes. The scenes are logical story units that describe a singular event. The scenes can be split into shots. Shots are units of action and consist of a continuous set of frames. The transition from one shot to the next may be of various types: broadly categorized as abrupt change shots and gradual change shots. Abrupt change shots, also known as cut shots, denote an instantaneous transition from one shot to another. This occurs due to simplest physical concatenation of two successive shots. On the other hand, a gradual transition shot is obtained by incorporating photographic effects, usually through editing. It can be further classified as fade-out, fade-in, dissolve, and wipe shot. Fade-out is a gradual transition of a scene by diminishing overall brightness and contrast to a constant image (usually a black frame). Fade-in is a reverse transition of fade-out. Dissolve is a gradual super-imposition of two consecutive shots. In general, abrupt transitions are much more common than gradual transitions, accounting for over 99% of all transitions found in video[Krulikovska et al. 2010]. It is well known that, in case of abrupt transition, the last frame of a shot and the first frame of the following shot are uncorrelated [Mohanta et al. 2012]. Different techniques have been proposed in the literature to address the temporal segmentation of video sequences [Haller et al. 2009; Li and Ngan 2011]. Many

research works have focused on the uncompressed domain [Amri and Fathy 2010; Grana and Cucchiara 2007; Hameed 2009]. The simplest technique employed is one based on pixel-wise difference between consecutive frames [Grana and Cucchiara 2007] but it is very sensitive to the motion of objects. To address the variation in pixel difference and mutual information due to object motion and small camera pan, zoom, and tilt, features like motion vectors [Krulikovska, Pavlovic, Polec and Cernekova 2010] are incorporated to measure continuity. In [Mohanta, Saha and Chanda 2012] motion vectors are used as localised feature statistics.

Greyscale or colour histogram-based features are relatively stable though they lack spatial information. While most systems use intensity [Mohanta, Saha and Chanda 2012] or RGB colour histogram[Ma et al. 2012], some use other colour triplets, for example, YUV [Hameed 2009] or HSV palette [Chen, Deng, Guo, Wang, Zou and Wang 2010; Tapu and Zaharia 2011; Xu and Xu 2010]. When using colour histogram features, it is necessary to decode the compressed video streams firstly [Ma, Yu and Huang 2012]. Measuring the colour histogram difference is a good indicator of abrupt shot change and has provided high rate detection results [Chen, Deng, Guo, Wang, Zou and Wang 2010]. This can be affected by fast global motion, such as action scenes and quick pan and zoom and special effects. It is argued that different colour spaces are better for shot boundary detection [Hameed 2009]. In [Krulikovska, Pavlovic, Polec and Cernekova 2010] they used both RGB and YUV colour spaces and found that RGB format gave a marginally higher detection rate. There are rare colour triplets in use for SBD such as L*a*b* colour space which is used by [Küçüktunç et al. 2010] for their SBD implementation for a content based copy detection application. Edge and texture information is another content feature description that is useful for detecting shot boundaries [Chan and Wong 2011]. In [Mohanta, Saha and Chanda 2012] they use an edge strength scatter matrix to distinguish between fade in/fade out, dissolve, wipe and cut shots by mapping a scatter matrix of the pre-normalized gradient magnitude of corresponding edge pixels of successive frames which reveals the type of frame transition. Many works have used a hybrid technique in an effort to negate the disadvantages of one technique by using the strength of another. In [Grana and Cucchiara 2007] they use a pixel based approach and histogram based approach in a unified linear transition decomposition. In [Chen, Deng, Guo, Wang, Zou and Wang 2010] they use two algorithms for shot detection, as each one negates the disadvantages of the other.

2.2.2 Semantic Temporal Segmentation

An important step in the process of video structure parsing is that of segmenting the video into individual scenes or "logical units" [Mezaris et al. 2010; Sidiropoulos et al. 2011]. Scenes are defined as "composed of one or more shots which present different views of the same event, related in time or space" [Hunter and Iannella 2009]. Shots describe actions or self-contained events that do not have much focus until they are put together to describe a larger story unit that are commonly called scenes. Shots have a physical boundary that is accurately detectable by computer vision processing methods, whereas scenes are demarcated by semantic boundaries that are harder to detect by automatic methods. Video segmentation to shots and scenes are two different problems that are characterized by considerably different degrees of difficulty [Smeaton, Over and Doherty 2010]. The close relation between video scenes and the real-life events depicted in the video make scene detection a key-enabling technology for advanced applications such as event-based video indexing [Ballan et al. 2011]. This has been used in movie video summarisation [Sang and Xu 2010], artistic video archives [Mitrović et al. 2010], news story classification [Aly et al. 2010; Choroś and Pawlaczyk 2010; Dumont and Quénot 2012; Heejun and Jaesoo 2011], sports video classification [Choroś and Pawlaczyk 2010; del Fabro and Boszormenyi 2010; Huang and Tung 2010; Tjondronegoro and Chen 2010], scene genre identification [Ellouze et al. 2010; Zhu and Liang 2011]. Much work has been done on scene segmentation in the last decade. This can be roughly classified into three categories:

- **Shot clustering based approach:** In [Choroś and Pawlaczyk 2010] they cluster shots based on content features of TV sports news broadcasts. In [del Fabro and Boszormenyi 2010] they cluster shots into scene sequences by employing a distance similarity measure between shot clusters that compare motion information.
- **Boundary detection based approach:** In [Baber et al. 2011] they detect fade and abrupt shot boundaries by frame entropy analysis and frame difference. [Dumont and Quénot 2012] propose a fusion of content feature vectors that, when analysed, will show story segment boundaries where the multimodal vector shows a clear demarcation for most features.
- **Model based approach:** In [Chao et al. 2011] they use a Hidden Semi-Markov Model (HSMM) to model the relationship between the script video alignment and video shot clusters to the hidden scene partition sequence.

Many methods have been developed to partition video scenes. Generally speaking, automatic scene boundary detection techniques can be categorized into following classes, i.e. graph based [Ayadi et al. 2012; del Fabro and Boszormenyi 2010; Mezaris, Sidiropoulos, Dimou and Kompatsiaris 2010; Sakarya and Telatar 2010; Sakarya et al. 2012; Seeling 2010; Sidiropoulos, Mezaris, Kompatsiaris, Meinedo, Bugalho and Trancoso 2011; Su et al. 2012; Tapu and Zaharia 2011], film editing technique based [Choroś and Pawlaczyk 2010; Zhu and Liang 2011], statistics learning based [Baber, Afzulpurkar and Bakhtyar 2011; Chao, Changsheng, Jian and Hanqing 2011; Ellouze, Boujemaa and Alimi 2010; Huang and Zhang 2010; Mohanta et al. 2010; Sang and Xu 2010; Seung-Bo et al. 2010; Tjondronegoro and Chen 2010; Wilson et al. 2010; Zeng et al. 2010], and multi-features based [Dumont and Quénot 2012; Ercolessi et al. 2011; Heejun and Jaesoo 2011; Huang and Tung 2010; Hui and Cuihua 2010; Li et al. 2010; Mitrović, Hartlieb, Zeppelzauer and Zaharieva 2010; Poulisse et al. 2012].

Graph based techniques for shot detection have been very successful when employed in semantic scene segmentation. In [Sidiropoulos, Mezaris, Kompatsiaris, Meinedo, Bugalho and Trancoso 2011] they propose a technique, whereby the low-level and high-level features extracted from the visual and the aural channel have been used jointly. In [Tapu and Zaharia 2011] they use a computationally efficient shot extraction method which adopts a normalized graph partition approach. [Sang and Xu 2010] propose an effective method for video scene segmentation based Ncut to decompose the scene similarity graph into sub-graphs (scene clusters). [Ercolessi, Bredin, Sénac and Joly 2011] use speaker diarisation to segment TV series into scenes. Scenes are created first by screen writers who produce a script of the screenplay. The script information itself can be combined with the footage to identify scenes. Both [Li, Wang and Wang 2010; Seung-Bo, Heung-Nam, Hyunsik and Geun-Sik 2010] use the movie script, that has the scene information, and match it to the subtitle information of the footage.

The solution to the problems of relying on one set of features is to use a multi-feature based approach. For example [Chao, Changsheng, Jian and Hanqing 2011] combine script names with faces in the video to negate the problems mentioned before, along with the discrepancies between the script and subtitles and the scarcity of subtitles in non-English speaking languages. In [Poulisse, Patsis and Moens 2012] they use a similar technique for live sports action. In [Dumont and Quénot 2012] they use numerous visual and audio features and fuse them together after applying a local temporal context window to them.

2.2.3 Spatiotemporal Segmentation

Spatial segmentation aims at grouping image pixels together based on attributes that define a pixel region into a semantic object. Spatiotemporal segmentation takes this one step further by adding a temporal element to the segmentation by tracking the pixels over time and defining the object in both appearance and motion [Fei and Zhu 2010; Grundmann et al. 2010; Sharir and Tuytelaars 2012; Tian et al. 2011; Vazquez-Reina et al. 2010]. Spatial segmentation differs from spatiotemporal segmentation in that temporal coherence of the object boundary maybe compromised when segmenting a series of contiguous frames as they are treated in isolation and redefine the object boundary for every frame [Grundmann, Kwatra, Mei and Essa 2010]. Objects can be defined at several levels, from general geometric boundaries, such as bounding boxes [Babenko et al. 2011] to regional granularity [Grundmann, Kwatra, Mei and Essa 2010]. The best balance is achieved when objects are segmented into regions that can be easily recognised by humans [Grundmann, Kwatra, Mei and Essa 2010; Ladický et al. 2010; Ochs and Brox 2011]. These usually follow a hierarchical structure based on perception. For example, a person can be segmented into arms, torso, arms and legs [Shao et al. 2012]. The most popular spatiotemporal approach employed is that of Optical flow, a time-domain motion analysis algorithm [Ghuffar et al. 2012; Lezama et al. 2011; Lin et al. 2011; Ochs and Brox 2011; Sharir and Tuytelaars 2012; Tian, Xue, Lan, Li and Zheng 2011; Van den Bergh and Van Gool 2012]. Other types of Motion Analysis techniques apart from Optical flow have been suggested [Christodoulou et al. 2011; Fei and Zhu 2010; Porikli et al. 2010] but are very similar in their machinations. Conditional Random Fields (CRF) and Markov Random Fields are techniques that have recently been gaining popularity for spatiotemporal segmentation [Vazquez-Reina, Avidan, Pfister and Miller 2010].

Numerous works have looked at modelling the background first and then detecting the pixels of foreground objects by differencing the current frame with the background. This approach is only effective if the camera is stationary or has a background that is unchanging. These techniques are more suited to the surveillance domain of CCTV [Appiah et al. 2010; Bai et al. 2010; Ladický, Sturgess, Alahari, Russell and Torr 2010; Ma and Chen 2010]. The most basic way to do this is by using a frame difference techniques such as in [Christodoulou, Kasparis and Marques 2011].

This looks at the temporal difference in pixels across frames that identify moving object pixels across a non-moving background of pixels. The most popular method for background modelling is a unimodal approach that uses a Gaussian Mixture Model [Zhu et al. 2012]. It constructs a gray-scale distribution model of each pixel based on the distribution information of each pixel in time domain and builds a background model of the pixels [Bai, Wang and Sapiro 2010; Subudhi et al. 2011]. The technique gives poor results when used in modelling non-stationary background scenarios like waving trees, rain and snow. In [Appiah, Hunter, Dickinson and Meng 2010] they use a multimodal approach, modelling the values of each pixel as a Mixture of Gaussian. The background is modelled with the most persistent grey scale intensity values.

With the advent of stereoscopic cameras and the emergence of 3D video, techniques have been developed that take advantage of the depth field to provide spatiotemporal segmentation. In [Ma and Chen 2010] they have used a stereoscopic camera to integrate depth information into the object segmentation process. They produce a 3D depth density image from the disparity map and then apply a region growing method to segment foreground objects. In [Ghuffar, Brosch, Pfeifer and Gelautz 2012] they use motion estimation and segmentation of independently moving objects in video sequences from a time of flight range camera that can record depth. They present a motion estimation algorithm which is based on fusion of range flow and optical flow constraint equations. The flow fields are used to derive long-term point trajectories. A segmentation technique groups the trajectories according to their motion and depth similarity into spatiotemporal objects. In [Van den Bergh and Van Gool 2012] they use a real-time super-pixel segmentation algorithm, which employs real-time stereo and real time optical flow. To reduce computational expense a few works have tried to segment video without decoding the signal from its compressed state [Fei and Zhu 2010; Khatoonabadi and Bajic 2013; Porikli, Bashir and Huifang 2010; Tsao 2011]. In [Appiah, Hunter, Dickinson and Meng 2010] they process a multimodal background differencing algorithm on a single Field Programmable Gate Array chip and four blocks of RAM.

Temporal continuity of the spatiotemporal segmentation regions can only be achieved by tracking the object boundaries over the duration of a shot. In [Vazquez-Reina, Avidan, Pfister and Miller 2010] they extract multiple segmentation hypotheses of super-pixel flows in each frame, and then search for a segmentation consistent over multiple frames. Robust unsupervised video segmentation must take into account spatial and temporal long-range relationships between pixels that can be several frames apart. Segmentation methods that track objects by propagating solutions frame-to-frame [Yongquan, Weili and Shaohui 2009] are prone to overlook pixel relationships that span several frames. According to [Ayvaci and Soatto 2012] local image measurements often provide only a weak cue for the presence of object boundaries. At the same time, object appearance may significantly change over the frames of the video due to, for example, changes in the camera viewpoint, scene illumination or object orientation [Lezama, Alahari, Sivic and Laptev 2011].

Two or more syntactic features are often used to segment objects. This hybridisation is applied in two ways; by combining techniques that use different features symbiotically to segment the object or use different features to independently segment the object and use the results from one to reinforce the other. Examples of a symbiosis technique is given in [Bai, Wang and Sapiro 2010] where they use motion estimation as a probability framework of object localisation and then adapt the selection of colour model from global to localised for different parts of the object so successive frames can be easily segmented. The work from [Hu and Hsu 2011] is an example of the second type that uses different syntactic features to extract and then reinforce object segmentation. They combine all three feature classes: colour, motion and edge information to extract foreground objects. Sometimes the same feature can be used by two different techniques to reinforce each other. An example of this, is the use of pixel intensity values in [Mahesh and Kuppusamy 2012] with both frame difference and intersection of frame algorithm to extract objects.

2.3 Semantic Content Features

There has been much work recently on semantic concept detection [Weiming, Nianhua, Li, Xianglin and Maybank 2011]. The two semantic features that need to be modelled are spatial and temporal relationships. Spatial relationships exist only between spatiotemporal regions and can evolve over time. Temporal relationships can be modelled between all features, syntactic or semantic, as all video features have a temporal component. The representation of spatial relationships in video has been extensively discussed [Weiming, Nianhua, Li, Xianglin and Maybank 2011]. One of the most important abilities of a semantic content model should be to be able to query the position of objects in relation to other objects or their relative positioning within the shot, not just as a reference to their absolute positioning stated as coordinates [Agius and Angelides 2005]. Unlike spatial relationship in images,

spatial relationships in video have a temporal dimension. Temporally consecutive frames have explicit spatial constraints with object inheritance, spatial relationships and motion information from their previous frames. Temporal trajectories of spatial relations among objects are as important as temporal object trajectories to represent object activities and reveal semantic evolution of spatial properties over time [Baştan, Çam, Güdükbay and Ulusoy 2010; Kannan et al. 2010; Vrochidis et al. 2010]. In such systems, indexing techniques work on modelling video by treating video shots/scenes as collections of still images, extracting relevant key-frames, and comparing their low-level features. Spatial relationships are formalised using Allen's temporal logic as a basis [Güsgen 1989]. Spatial relationships between objects describe the relative location of objects in relation to other objects, rather than their absolute screen coordinates, within the segment. Sometimes when it is difficult to derive screen coordinates, a spatial relationship is the only way to model an object's presence. The spatial relationships between two objects may differ over time within the same segment [Manjunath et al. 2002]. Spatial relationships are a highly active field in other domains such as content based information retrieval [Singhai and Shandilya 2010], human activity classification [Ryoo and Aggarwal 2009], robotics [Rosman and Ramamoorthy 2011] and surveillance [Ryoo et al. 2010].

Semantically queries can be structured to investigate the temporal relationships not only between syntactic features, but also semantic features. Temporal relationships between these features allow the content model to express dynamism at the higher level [Agius and Angelides 2005]. Temporal relationships were first defined meaningfully by J.F. Allen [Allen 1983]. Often, the exact relationship between two times is not known, but some constraints on how they could be related are known. For example, when modelling knowledge of history, one may only need to consider time in terms of days, or even years. Finally, the model should support persistence. It should facilitate default reasoning of the type, "If I parked my car in lot A this morning, it should still be there now," even though proof is not possible as the car may have been towed away or got stolen. Allen's scheme for temporal relationships was expanded on by the MPEG-7 group [Manjunath, Salembier and Sikora 2002].

2.4 Problems with extracting and modelling video content features

The importance of pre-processing raw media is, firstly, to make feature extraction more effective by either filtering or converting the media so the salient points of the features of interest are easier to extract and, secondly, to reduce processing time to within acceptable levels by reducing computational complexity. The reviewed literature shows this to be a benefit in having a defined media preparation stage as the resultant media conversion improves feature extraction by many factors. Most feature extraction systems have not employed an active strategy in defining a raw media pre-processing methodology. They have seen it as an implicit factor of extracting only a certain feature and do not apply a more broad philosophy to the system as a whole. Such a method could be more beneficial as the pre-processing could be applied more methodically in order to improve feature extraction for more features.

The choice of low-level syntactic primitives used for syntactic feature extraction is an important factor in the success and effectiveness of extracting the desired content features. To extract a certain syntactic feature the chosen primitive feature type is integral to the feature extraction process whose choice is influenced by its physical properties and attributes. For instance, when wishing to extract shots by identifying shot boundaries there are a number of techniques available. If using a colour histogram difference technique the choice to use colour histograms is implicit, but the choice of colour space to be used is not. The choice of colour space has a direct bearing on the detection rate. Similarly the choice of low-level syntactic primitive should be made with a view to reusability and polymorphism of use in a multi-content feature extracting environment. Most systems assign one primitive to one process, increasing computational expense and not fully leveraging the benefits that a more multi-content centric approach could yield.

Temporal video segmentation is a fundamental building block of all syntactic and semantic feature extraction systems. The ability to segment video into a temporal hierarchy is imperative to the construction of a video content model. Central to temporal video segmentation is shot segmentation. Many techniques have been proposed and all address the same problem, i.e. identify shot and type of boundary between shots, e.g. abrupt or transition. Very few techniques have equal success at identifying both types, and if they do they do not have the precision and recall of techniques that identify one or the other. The type of shot boundary can be semantically significant, as it can indicate the start of a semantic event. For example the presence of a fade in is usually an indication that a new scene has started. The relative entropy of the shot can also indicate genre, for instance action scenes usually have fast moving panning shots or a lot of camera shake. Identifying such features and attributes of shots is important in linking semantic meaning to the underlying syntactic features.

Whereas shot segmentation is a purely syntactic derivative, scene segmentation is dependent on the semantic relationship between shots. A scene describes a collection of shots that are temporally related to describing a semantic event as a narrative unit. Due to the semantic nature of scenes there are no effective machine readable techniques that can be used to directly identify scenes generically. There are syntactic feature techniques that are genre specific that identify possible scene boundaries by certain syntactic "landmarks" but these rely too heavily on format and content within the content stream being standardised with little change. More generic techniques have used either film grammar or machine learning techniques to either cluster shots, detect boundaries or model shots into scenes. These techniques though do not enjoy a high level of precision and recall such as shot segmentation. Scenes are an implicit structure required for content modelling, as they are a bridge between the physical content of the media and the meaning of the content. Scenes can be described for this reason as a mid-level syntactic feature by the way they temporally group shots into semantic events. Scene segmentation needs techniques that are genre independent and is more semantically correct in boundary definition. The second point needs the technique to have an understanding of the semantics of the content. This requires knowledge of the events going on within each shot and how they relate to other shots that could be semantically grouped with them.

Another important step in syntactic feature is spatiotemporal segmentation. This is integral in defining events by establishing the interaction between objects. Similar to scene segmentation, spatiotemporal segmentation defines the boundaries of semantically meaningful objects. Spatiotemporal segmentation has similar problems to scene segmentation in the fact that delimiting the borders of an object is a subjective process based in semantics. Due to its semantic nature the spatiotemporal objects can be classified as mid-level syntactic features. Also due to its temporal nature the segmentation evolves over time, this is what differentiates it from image segmentation. Techniques for spatiotemporal segmentation are centred on grouping pixels based on changes in colour, texture or motion. Although there has been relative success with unsupervised techniques these are limited by certain conditions that must be met for it to be successful. Most techniques employ a learning phase or training data as a base line for segmentation.

The relationship between features is as important as the features themselves, as these allow content to be queried in a meaningful way that is natural to consumers. Spatial and temporal relationships answer two of the four major categories of querying, i.e. "where" and "when", "who" and "what" [Agius and Angelides 2005]. The spatial relationship between objects is important the querying of events as it allows users to query a particular arrangement of objects, or change in arrangement, that might indicate a particular event or action. Temporal relationships are the basis of querying the occurrence of events in relation to other events. The ability to query temporally is a powerful tool as it not only queries syntactic or semantic features homogenously but is also able to find the relationships between heterogeneous features. This allows the content model to be queried in a multi-faceted manner for all the features contained within. To have both of these relationships to be stated explicitly modelled means that the content can be uniformly queried from any system with the results being the same regardless of method.

Many different video indexing and retrieval systems use various content feature sets that are usually sculpted to fit the purpose of their querying methodology. No syntactic and semantic content feature integration exists that could help in reducing the semantic gap [Wang et al. 2010]. By directly mapping semantic features to supporting syntactic features, the semantic gap can greatly be reduced. If video content indexing systems applied this classification to the content features they extract they could create content models that would be universally compatible with all similar video content retrieval systems and the results from a query on one system would have identical results on another system if the same query were used.

3. MAC-REALM FRAMEWORK

The only way to achieve the goal of extracting and modelling content features from the video stream is to produce a framework were the flow of control follows a path of processing the raw media into a content model, whilst transforming the content features into content descriptions. As the content passes through the framework, it will be refined into more complex and meaningful content descriptions. The strength of the framework is that each stage of its process is designed to provide a complete set of content features and descriptions that can be reused or extended to capture even more content features and descriptions. The framework as a whole will provide a content model that will have a syntactic content description base that is semantically linked spatially and temporally, reducing the semantic gap between those sets of syntactic and semantic features.

MAC-REALM is such a framework that extracts syntactic and semantic content features from a video stream and then models them into an MPEG-7 content model that integrates the features so that the semantic gap is reduced for multi-content type queries. The way the content features are extracted is directly related to the way the content descriptions are modelled. The segmentation of the content features is designed to extract features the features in a hierarchical extraction process. This hierarchical extraction process is then mimicked in the modelling of the content model so that the syntactic and semantic content features are closely coupled. The resulting richly and granularly detailed content model is structured to facilitate multi-content type content type search from compliant content based video search applications. MAC-REALM's content model can be used by many video search applications that are MPEG-7 compliant. It models the features so that the same query in each application should retrieve the same results. It achieves this by removing ambiguity caused by the differing ways that applications interpret relationships between content features. It also allows the content to be queried both syntactically and semantically in a manner that is familiar to the way video is structured and perceived by consumers.

The framework extracts three types of feature: low and mid-level syntactic and high level semantic relationships. Five feature components represent these three types of feature. The first feature to be modelled is the low-level feature of shots. Other content features use the shots as the reference features. The mid-level features are objects and scenes. Unlike the low level features, they are semantically derived syntactic features. They cannot be extracted by purely machine driven processes, as they require a level of semantic "recognition", as well as comprehension, to their syntactic boundaries. The objects require a twofold approach; first they are segmented from the background, similar to image segmentation, and then they need to be tracked for the duration of the shot. Scenes do not have a generic syntactic marker that can be used to segment them. They are usually demarcated with specific film grammar techniques or domain specific graphic or effect transition. Each video stream has its own formulation of syntactic features that identify where the scene boundaries are. The high level semantic relationships consist of two components: spatial and temporal relationships. The spatial relationships are modelled in two ways, absolutely and relatively. This allows the position of objects to be queried or analysed with respect to their global position and their position to each other. Unlike the spatial relationships that are modelled for only one feature, the temporal relationships are modelled between all content features. The modelling of temporal relationships between all features, both syntactic and semantic, makes the querying and analysis of the content multi-dimensional and allows syntactic and semantic content descriptions to be queried temporally by direct comparison. This not only allows polymorphic querying of the content, but can also be used by temporal concept learning methods to model concepts to features that are not exclusively syntactic or semantic, such as scenes.

To model these five feature components we must extract them from the video stream. The raw signal must be pre-processed before extraction can take place. This is to improve the efficiency and effectiveness of the extraction process. In the extraction process, the syntactic features that form the foundation of the content model are segmented. The features are then analysed together both spatially and temporally and linked to form semantic relationships. The syntactic and semantic features are then modelled into an MPEG-7 compliant content model that is made available to all MPEG-7 video search engines.

3.1 MAC-REALM Architecture

The MAC-REALM Framework comprises four planes and three layers: the raw media plane, the extraction plane, the analysis and linkage plane and the modelling plane. The three layers are the MPEG-7 layer, the application layer and the content layer. In **Error! Reference source not found.** we show the MAC-REALM architecture. It shows the flow of the content processing through the planes and the content transformation through the layers. Where MAC-REALM intersects between layers and planes we have stages of processes of content or processed content. Each stage is responsible for the content conversion process at that intersection. The flow of content media between stages goes left to right and down then back up in the next plane.

The raw media is the video stream that will be extracted into content features and then into content descriptions, and finally represented by an MPEG-7 content model. The pre-processing stage processes the raw video streams into syntactic media that is optimised for feature extraction. The pre-processing stage removes redundant data by eliminating chunks of data that is only incrementally different to each other by a small margin, as to be insignificant in change. The media is then filtered to emphasis the content feature properties that are used for feature extraction.

The syntactic media stage stores the filtered frames and histograms from the pre-processing stage, ready for the syntactic feature extraction stage. The syntactic feature extraction stage processes the syntactic media into syntactic content features. Three processes are part of the syntactic feature extraction stage, the shot, object and scene extraction processes. The shot processes extracts cut and transition shots. The object sub-process segments the objects and then tracks them. The scene process detects and segments scene boundaries. The segmented syntactic content features are then sent to two places the first is the semantic media stage for storage and the second is to the syntactic modelling stage. The syntactic modelling stage is where content features, once converted, are stored as MPEG-7 syntactic feature descriptions. The semantic media stage is where the temporal and spatiotemporal syntactic features are stored ready for processing by the spatial-temporal mapping stage. The spatial-temporal mapping stage consists of two processes, the spatial and temporal relationships process. The spatial relationship process analyses the spatiotemporal objects and maps the spatial relationships between them. The temporal relationship process then analyses all content features created and maps all the temporal relationships between them. Once the semantic content features are converted to MPEG-7 content descriptions they are stored in the semantic modelling stage.

All the MPEG-7 syntactic and semantic content descriptions are stored in the syntactic and semantic descriptions stage. The syntactic and semantic descriptions are analysed and then integrated into a MPEG-7 content model. The content model is then serialised and stored in the modelled media stage.

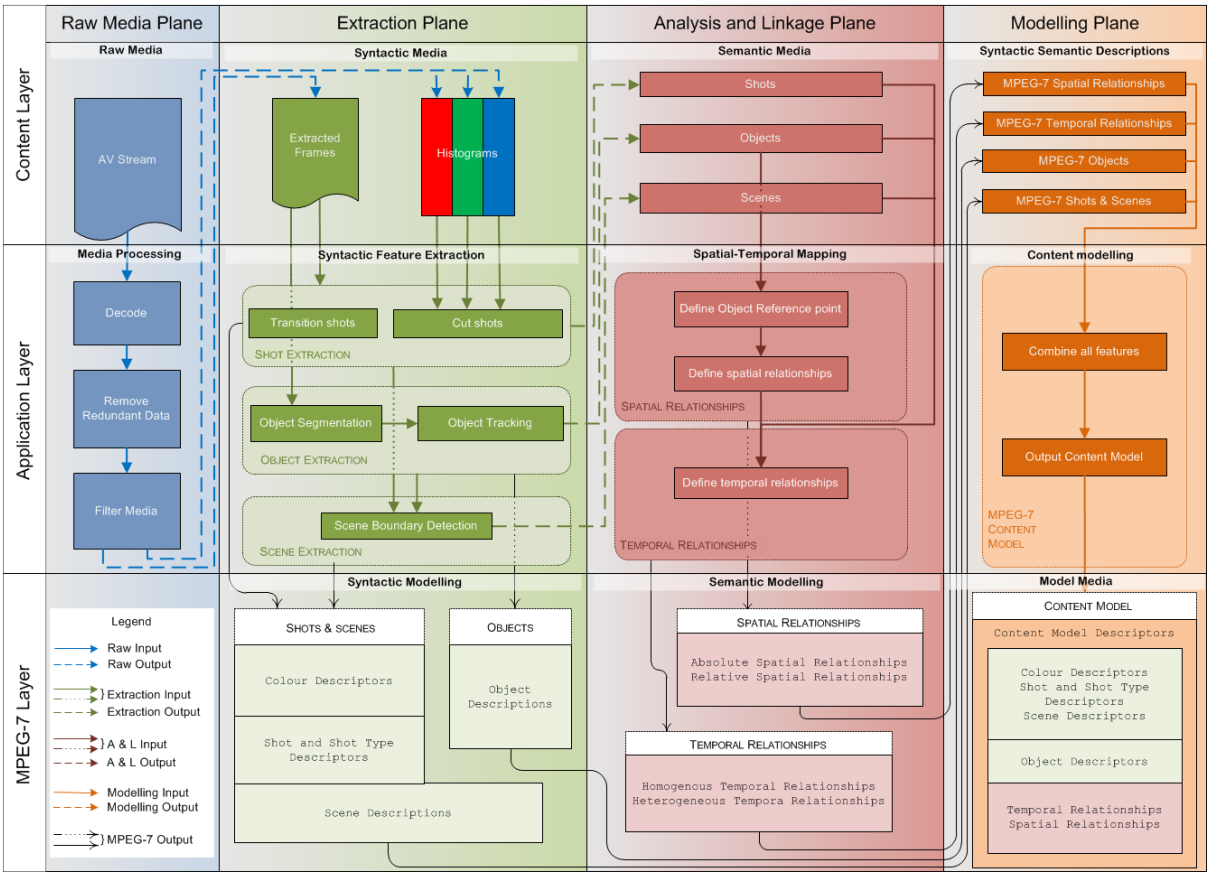


FIGURE 1: MAC-REALM ARCHITECTURE

3.2 MAC Layers

The three layers of MAC-REALM relate to the processing of the content that each plane goes through to convert its content media into modelled MPEG-7 content descriptions. The three MAC layers are: the **M**PEG-7 layer, the **A**pplication layer and the **C**ontent layer. The content layer stores the media to be processed. The application layer processes the media and outputs syntactic or semantic descriptions of that media. In the MPEG-7 layer the syntactic or semantic description is modelled into MPEG-7 content descriptions. Whereas the planes describe the

transformation of the video stream into a content model, the layers describe the process of the content being transformed and translated from media into content descriptions of the media.

The content layer contains the media for MAC-REALM. The type of media contained changes as you move down the planes. As the media moves from raw to modelled state the content and content features become more advanced and the content descriptions are of a higher type. In each plane, the content media can be used as supplementary media for the MPEG-7 content model. This could prove useful for adapting the media to a user's usage environment as discussed in [Sofokleous and Angelides 2008]. The application layer is the processing layer for MAC-REALM and processes the content media into MPEG-7 descriptions. The processing of the content media becomes more complex content feature-wise as MAC-REALM goes across the planes. The application layer has two tasks: to process all content description into either syntactic or semantic content features and to convert these content features into MPEG-7 descriptions. The MPEG-7 layer stores the MPEG-7 content descriptions as they are created for each plane. In the syntactic and semantic content feature extraction planes, the MPEG-7 content descriptions for those planes are stored. In the modelling plane they are combined to finally create a content model of the syntactic features and semantic relationships. The MPEG-7 descriptions for each plane are complete and can be extracted and used to build customised content models for specific uses and domains if necessary.

3.3 REALM Planes

0 shows the basic flow chart diagram for content extraction and modelling. Each stage of processing where the content is converted into another content type is represented by a plane within the MAC-REALM framework. This represents the REALM processing model and in the framework is represented as four planes: **R**aw media, **E**xtraction (of syntactic features), **A**nalysis and **L**inkage (of semantic relationships) and **M**odelling of content features. The diagram shows how the video is processed through each stage beginning with pre-processing of the raw media. The syntactic features are extracted from the filtered media and the semantic features are derived and linked to those features. Finally, both syntactic and semantic content features are modelled into a standard content description that can be read by any compliant video search and retrieval system.

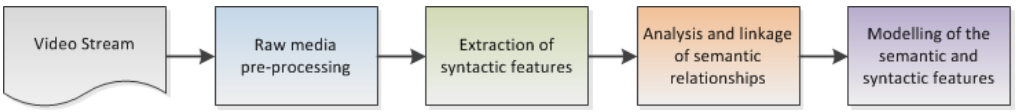


FIGURE 2: REALM VIDEO EXTRACTION AND MODELLING PROCESS

3.3.1 Raw Media Plane: Pre-processing

The video stream has to be pre-processed to filter the media so that it makes extracting the features more effective and efficient. The video stream also has to have feature redundancy techniques applied to remove the non-salient content data that adds no value to the extraction process and increases processing time. Before filtering, we must initially decode any compressed video into an uncompressed state, where each frame becomes available. During the initial decoding, we perform a redundancy operation whilst we decode all the frames. There is usually between 24 to 30 frames per second for any video footage. Experiments have shown that only two frames per second is adequate for shot segmentation [Chan and Wong 2011]. Two key frames are picked per second for use in the extraction process. The key frames that are chosen are at the beginning and middle frames of every second:

$$Computational\,reduction = \frac{1_{start}}{fps} + \frac{1_{finish}}{fps}$$

To decide on what filtering techniques we need we must first look at the features that are to be extracted and what it is required to extract them. The syntactic content features we need to extract from the raw media directly are the key frames, shots and objects. Each feature needs different filtering techniques applied to improve its particular segmentation process. Shots need to have the lighting source in the target video clip to be even and without any abrupt changes e.g. flashing light sequences such as lightning. Objects, depending on the technique used for extraction, also need the light source to be even throughout the shot for the segmentation to be effective as the outline of the objects becomes obscured in dimly-lit scenes. Both also need the removal of distortion or "noise" that can affect the segmentation process. The way to negate the effect of such lighting changes is to use a colour space that is tolerant of such changes and can reduce their impact on shot segmentation and spatiotemporal extraction. To

reduce the effect of lightning changes the video needs to be converted into the RGB colour model, if not in RGB already. RGB is shown to reduce the effect of lightning changes and improve invariance to shadows [Kristensen et al. 2006]. Noise is removed by performing a flattening function over each of the extracted frames. Noise comes in the form of pixel "particulates" that are usually formed as artefacts left over from the decoding process as information was lost during the compression of the original video stream. The technique from [Yongquan, Weili and Shaohui 2009] is adopted. A median filter is used over each key frame to reduce the noise of each pixel by smoothing the pixel using the adjacent pixels. The noise reduction removes pixel-fine artefacts from the frame that could cause erroneous segmentation boundaries for both object and shot boundary detection. The brightness and contrast are then adjusted to compensate for bad lighting levels. Once the adjustment is performed, the prominent features of the video stream become much more visible and thereby make the extraction processes more reliable. Figure 3 depicts the processing that takes place at Raw Media Plane

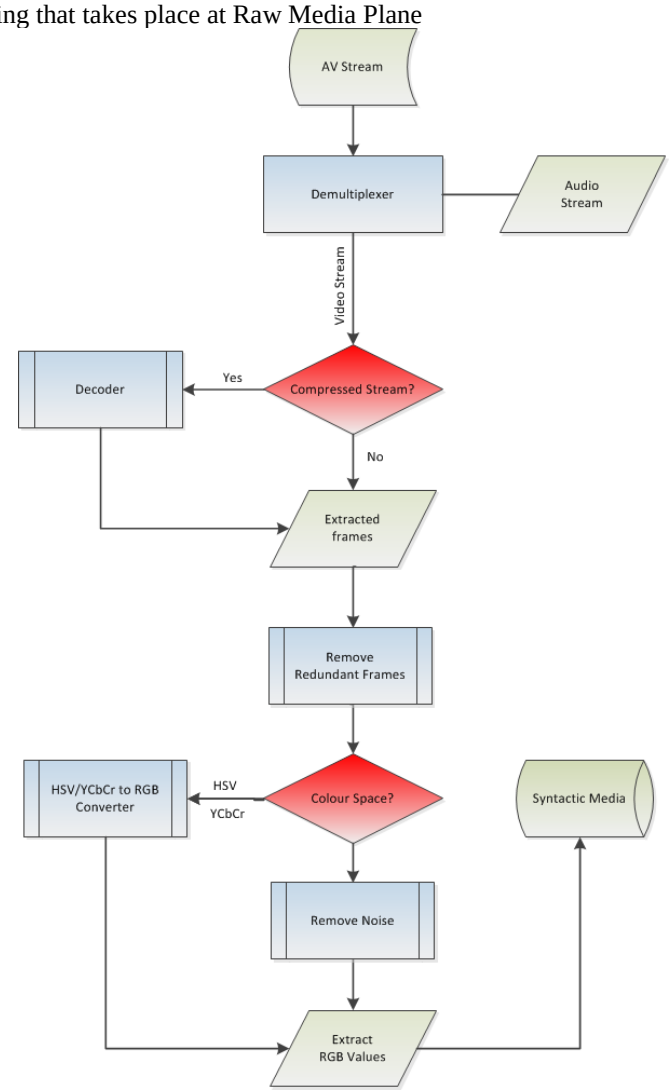


FIGURE 3: RAW MEDIA PLANE PRE-PROCESSING

3.3.2 Extraction Plane: Syntactic Feature Extraction

Temporal and spatiotemporal segmentation are key syntactic features that serve as the foundation of the content model. The temporal segmentation content features consist of one low level syntactic feature and two mid-level syntactic features. The low level syntactic feature is the shot, and the mid-level syntactic features are the scenes and objects. The mid-level syntactic features have a conceptual structure and are harder to extract directly from the video stream. To facilitate this better the low level syntactic features are extracted first and then used as the basis for extraction of the mid-level syntactic features. The shots are the basic building blocks for the content model. Each

shot is represented by a key frame extracted during the pre-processing stage. The spatiotemporal segmentation of the video stream begins after the shot extraction process. After the spatiotemporal segmentation has taken place the scenes are extracted. Once all the features are extracted, they are described by MPEG-7 syntactic content description schemes. The syntactic feature extraction process for this plane is shown in Figure 4.

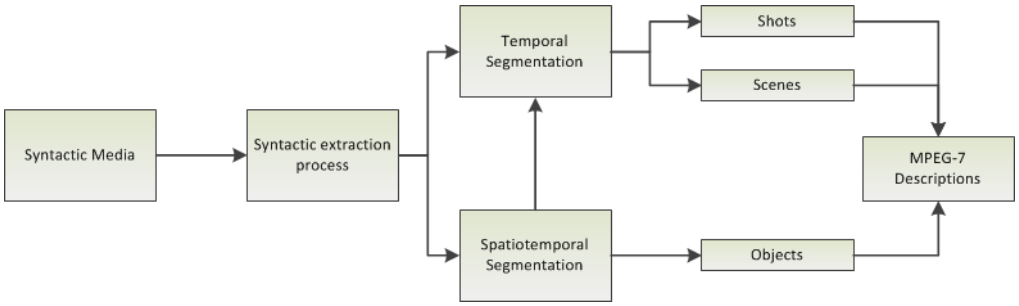


FIGURE 4: EXTRACTION PLANE SYNTACTIC FEATURE EXTRACTION

Shot segmentation requires two attributes of the shot need to be detected, the first is the boundary between shots and the second is the type of transition between the shots. type of boundary usually indicates a semantic event change. Normal abrupt cut transition shots are normally associated with non-semantic changes, whilst gradual transition type shots are usually an indicator of a new semantic narrative within the video stream. These visual cues are important for establishing semantic event boundaries and are therefore important to any content extraction and modelling. MAC-REALM uses Colour Histogram Difference for abrupt transition detection and Edge Change Ratio (ECR) for FOI/Dissolve transitions. This improves the performance of the overall extraction process for both types of shot. Each is well suited to its particular type of transition and each achieves good precision and recall rates. We reduce the complexity of the calculation using one step processes for both abrupt and gradual transition shots. However, reduced complexity does not result in reduced performance. Results are comparable in precision and recall to similar techniques. Figure 5 depicts the shot extraction algorithm which consists of three sub-processes: ECR, Colour Histogram Difference and shot fusion. Shot extraction implementation is explained in detail in proceeding sections.

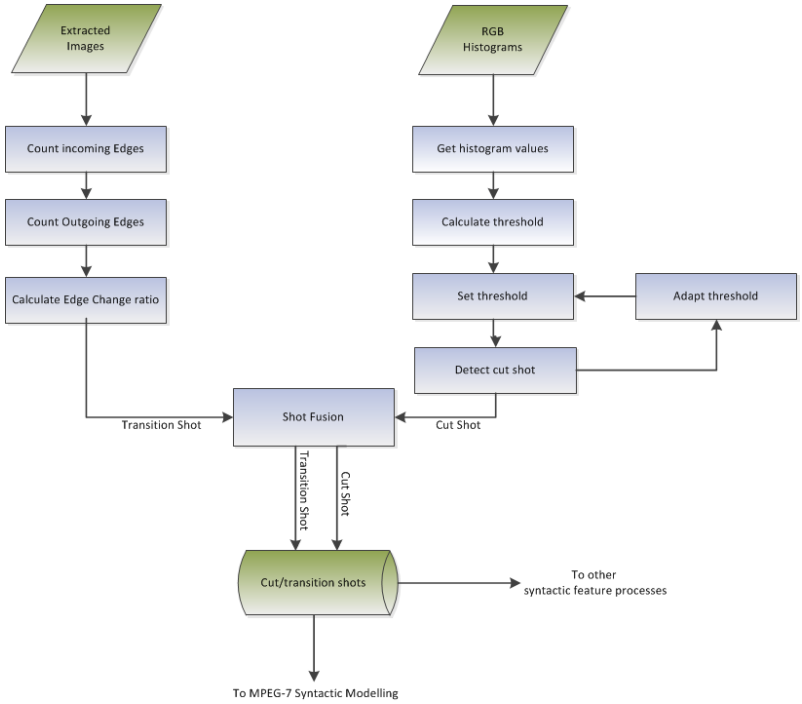


FIGURE 5: SHOT EXTRACTION

CHD work by detecting colour discontinuities between frames over a certain threshold. This indicates that a shot has been detected. This measure is denoted by CHD_i, where i is a frame of the shot, and is related to the difference or

discontinuity between frame i and $i + k$ where $k > 1$. The absolute difference between frames is used to compute the value of CHD_i [Lienhart 2001]:

$$CHD_i = \frac{1}{N} \cdot \sum_{r=0}^{2^n-1} \sum_{g=0}^{2^n-1} \sum_{b=0}^{2^n-1} |p_i(r,g,b) - p_{i+k}(r,g,b)|$$

where $p_i(r,g,b)$ is the colour histogram of a frame i with 2^{n-1} bins per histogram being considered. The RGB values for each frame are retrieved from the syntactic media component. Once the histograms are retrieved the colour histogram difference between each frame is calculated. For each frame stored in the syntactic media component, the colour value for each colour band is stored (RGB_n). To compute the histogram difference the formula by [Jacobs et al. 2004] has been adopted. The CHD between each frame is calculated, giving an initial threshold value using:

$$\Delta_{RGB} = 2 \times (RGB_{n+k} - RGB_n)$$

The threshold is adaptive as it is constantly revaluated since it works through the series of frames. When it detects a shot it resets the threshold and calculates a new threshold based on the first successive frame in the new shot. This method works to make the threshold maxima sensitive to the localised colour differences within the shot. The threshold technique is more suitable for MAC-REALM as it requires less processing time. This is because the square difference method uses a calculation over five contiguous frames for the additional task of finding transition shots.

To stop false positives arising from flashing lights we simply omit frames whereby $R_n, G_n, B_n \geq 250$ with the next frame $n + 1$ whereby $R_n, G_n, B_n < 250$, is used instead to determine if there was a cut shot. Figure 6 depicts the pseudo code for the CHD process.

1. Get histograms for frame $n = 1$ and $n + k$, RGB_n and RGB_{n+k}
2. Calculate initial $\Delta_{RGB} = 2 \times (RGB_{n+k} - RGB_n)$
3. For frames n to $\forall n$
    a. if $RGB_n > 250$ then skip RGB_n
    b. if $RGB_{n+k} > RGB_n + \Delta_{RGB}$
        i. then mark n as start of shot
        ii. Set $n = n + 1$
        iii. calculate new $\Delta_{RGB} = 2 \times (RGB_{n+k} - RGB_n)$
4. Mark last frame as end of shot

FIGURE 6: CHD PSEUDO CODE

The CHD technique is effective for abrupt cut shots were there is a sharp colour difference between two shots due to a sudden change of all colour pixel values. If there is a transition shot in which the colour pixel values between the two shots change gradually and smoothly, CHD will not pick up the change and will miss the shot change. To counteract this disadvantage, MAC-REALM extends the work by producing edge transition graphs over a 10 frame sliding window. Within the 10 frames we can identify the types of gradual transition from the shape of the transition graphs. The equation for this is given by:

$$ECR_n = \max\left(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-k}^{out}}{\sigma_{n-k}}\right)$$

In fade shots the amount of hard edges of objects increases from zero or decreases to zero over time. Fade ins, having increasing visible edges, lead to a positive slopped graph. Fade outs, having decreasing visible edges as the shot gradually fades to black, create a negative slopped graph. Dissolve shots on the other hand produce a concave hyperbolic graph as the pre-dissolve edges dissolve and the post-dissolve edges form. The algorithm charts the edge change ratio over the 3 frame sliding window. The sliding window allows the computational complexity of the algorithm to be greater than that of the original method at the beginning of the process. This is due to the edge count for the first ten frames is first being calculated and then after that only the 3rd frame is processed for a new edge count. The ECR algorithm for the sliding window is given below in Figure 7.

1. For Frames $n = 1$ to $\forall n$
2. Perform Canny edge detection[Canny 1986]
3. Then for every frame $n + k$, where $1 \leq k \leq 3$
   a. Count the number of P_n^{in} and P_{n+k}^{out} pixels.
   b. Dilate the edges and invert the images.
      i. Store dilated & inverted image n in DI_{n-k}^{out}
      ii. Store dilated & inverted image $n + k$ in DI_n^{in}
   c. Perform bitwise AND operation
      i. For every pixel (i,j)
         1. n && di_{n+k}
         2. $n + k$ && di_n
   d. Count the number of entering and exiting edge pixels in the images to obtain X_n^{in} and X_{n-k}^{out}.
   e. calculate the $ECR_n = \max\left(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-k}^{out}}{\sigma_{n-k}}\right)$
4. compare edge transition plot to FOI and dissolve patterns and see if a gradual transition is present
5. Increment n by one and repeat step 3

FIGURE 7: ECR PSEUDO CODE

Both techniques are used in shot fusion. The ECR technique picks up both types of shot, which we will call ECR_n^{cut} and ECR_n^{trans}, which represent cut and transition shots respectively. The ECR_n^{cut} is used as a confirmation on the CHD cut shots, CHD_n. If it is confirmed, then the probability of cut shot is considered high. If not, then the cut shot is considered a medium probability of being correct. If there is an ECR_{cut} but no corresponding CHD_n cut the probability of a shot is considered low. For transition shots, ECR_{trans} confirmation of a shot change is given by the first and last frame of a transition, ECR_n^{trans} and ECR_{n+k}^{trans} and comparing them against the corresponding frames from the CHD process, CHD_n and CHD_{n+k} to check to see if a shot change has occurred. If the histograms of CHD_n and CHD_{n+k} are compared sequentially and a cut shot is found we can deduce that ECR_n^{trans} and ECR_{n+k}^{trans} are the start and finish of a transition shot. The algorithm is presented in Figure 8.

1. For $\forall CHD_n^{cut}$
   a. If $CHD_n^{cut} = ECR_n^{cut}$ then CUT_{prob} is high
   b. If CHD_n^{cut} != ECR_n^{cut} then CUT_{prob} is average
2. If ECR_n^{cut} != CHD_n^{cut} then CUT_{prob} is low
3. For $\forall(ECR_n^{trans}, ECR_{n+k}^{trans})$
   a. Get CHD_n^{cut} and CHD_{n+k}^{cut}
   b. If $(CHD_n^{cut}, CHD_{n+k}^{cut})$ is cut then
      i. Mark ECR_n^{trans} as start of transition shot
      ii. Mark ECR_{n+k}^{trans} as end of transition shot

FIGURE 8: SHOT FUSION

Shot fusion improves significantly precision and recall of both abrupt and gradual transition shots. The modification of the ECR algorithm makes it effective at identifying gradual transition shots and does not assume additional computational expense than the original that compared two frames only. The CHD is reduced in computational efficiency by reducing the amount of frames processed for the threshold value. The reductions in computational expense lower processing time, which provides a more feasible overall time span for processing in MAC-REALM.

Object segmentation is a two-task process of segmentation and tracking. The first task is to segment the foreground objects from the background. The segmentation must also be able to differentiate and group multiple objects correctly, even when they are overlapping. The second task is to track the object(s) over time as they move. The tracking must be consistent and maintain the integrity of the object boundary from the initial segmentation. MAC-REALM approaches the two-step problem with a unified two-phase algorithm. In the first phase it uses graph cut theory to segment the initial frame, which can segment multiple objects [Noma et al. 2012]. The second phase tracks the objects segmented from the first phase, maintaining the integrity of the object silhouette, even if tracking multiple objects. Object extraction, or image segmentation divides an image into a number of disjoint regions such that the features of each region are consistent with each other. Since images generally contain many objects that are further surrounded by clutter, it is often not possible to define a unique segmentation. As a shot moves on from the original frame objects shape and their position may change. The objects shape changes for non-rigid bodies as they move, even rigid bodies can change their shape through the effect of perspective. Their position may change over time slowly, such as a sit-in interview, or rapidly, such as action sequences. The fast change sequences poise a problem, as it is hard to find continuity with consecutive frames as the position of the object could have drastically altered.

Object extraction is performed using a semi-automated procedure that segments based on structural pattern recognition to extract objects from their background. The object extraction process begins by creating two attributed relational graphs (ARG's). ARG's are very useful at not only model initialisation but also providing information on image structure. The first graph is an over segmented image using a watershed algorithm. The second image is a user defined input image that has different coloured stroke marks for different objects and the background. The first graph, the input graph, is processed against the second user defined graph, the model graph. The model graph is used to prime segmentation by providing an approximation of the objects core. From the initial stroke marks the regions are expanded, by merging the interconnected regions based on colour similarities and structural consistency. The background strokes are used to grow the background regions in the same manner. Once all regions have been assigned to either objects or background the segmentation stops. This method is very fast, and deals with the problem of image clutter by using user feedback to determine the initial ROI.

Once we have identified the region we have to track it across several frames so we can track the objects movements and spatial orientation for the duration that they appear. The algorithm described for segmentation was conceived for the use with still images. Using the same algorithm to segment the rest of the frames in the shot would lead to two problems. The first is the continuity of the object outline. The silhouette would become unstable, as the algorithm would segment each frame of the shot individually. This would cause the outline to fluctuate as the segmentation information from the prior frame is ignored. The second problem would be that the stroke marks used in the initial frame could become inaccurate as the shot progresses through the frames. What is required is a second algorithm that takes the ROI and tracks the pixels, using the information from the previous frame as the starting point for tracking. In order to solve the problem of tracking in the second phase we use the region covariance technique implemented by [Tuzel et al. 2006]. The tracking is initialised by extracting feature vectors from the ROI's of the key frame. From the vectors a covariance matrix is built of the feature vectors for the ROI's of each frame. The covariance matrix is a measure of how much two variables vary against each other. This is used to track the adjacent pixels next to each other in the ROI. The covariance becomes more positive for each pair of values that differ from their mean in the same direction, and becomes more negative with each pair of values that differ from their mean in opposite directions. The covariance descriptor method can use any set of features, intensity, colour, gradients, filter response.

For MAC-REALM, colour and intensity has been chosen for the covariance descriptors. These are selected as they are convenient features to extract as the colour histogram extraction algorithm can be used to track objects, and with a small modification can also be used to extract image intensity as an alpha value. The tracking algorithm used in MAC-REALM has many advantages over other techniques. It is robust against lighting changes and moving camera motion. It can track non-rigid bodies as they change. It can track fast moving object even if there is a large gap in position since the last consecutive frame. The algorithm is very fast at computing covariance as it uses integral images which are intermediate image representations for fast calculation of region sums. Using a two phase approach to segmenting and tracking the objects makes the overall result more precise, robust and fast than just using a single

technique. The image segmentation phase segments the initial key frame into ROIs quickly and precisely. The user defined strokes eliminate the confusion of image clutter and provide a template for the region growing algorithm. Once the key frame is segmented into ROIs the tracking algorithm then tracks them through covariance matrices of extracted feature vectors of the ROI's. The result is that objects can be reliably segmented and then tracked with minimal input from the user. It can handle multiple objects and objects that are similar in size and colour. Figure 9 shows the algorithm used for object segmentation and tracking.

1. Define $\alpha, \beta_V^j, for\ j = 1, \dots, |L_V|$, and $\beta_E^k, for\ k = 1, \dots, |L_E|$
2. $G_d \leftarrow G_m$
3. **for** each vertex $v_i \epsilon V_i$ **do**
    a. $f_{min} \leftarrow \infty$
    b. $minlbl \leftarrow -1$
    c. **for** each vertex $v_m \epsilon V_m$ **do**
        i. let $v_d \epsilon V_d$ be the vertex corresponding to v_m
        ii. let $E_d(v_d)$ be the vertex corresponding to v_d
        iii. save the original attribute vectors $\mu_d(v_d)$ and $v_d(e_d), \forall e_d \epsilon E_d(v_d)$
        iv. **for** each attribute $j = 1, \dots, |L_v|$ of vertex $v_d \epsilon V_d$ **do**
            1. $\mu'_d(v_d)[j] \leftarrow \sigma_\mu(\mu_d(v_d)[j], \mu_i(v_i)[j])$
        v. **end for**
        vi. **for** each vertex $e_d \epsilon E_d(v_d)$ **do**
            1. **for** each attribute $k = 1, \dots, |L_E|$ *of* e_d **do**
                a. $v'_d(e_d)[k] \leftarrow \sigma_v^k(v_d(e_d)[k])$
            2. **end for**
        vii. **end for**
        viii. compute the value of the cost function $f(G_d(v_i, v_m), G_m)$
        ix. **if** $f < f_{min}$ **then**
            1. $f_{min} \leftarrow f$
            2. $minlbl \leftarrow v_m$
        x. **end if**
        xi. restore the original attribute vectors of $\mu_d(v_d)$ and $v_d(e_d), \forall e_d \epsilon E_d(v_d)$
    d. **end for**
    e. $label\ (v_i) \leftarrow minlbl$
4. **end for**
5. Take $label\ (v_i)$ as input for C_1
6. Initialise $\hat{C} = C_1$
7. **repeat**
    a. **For** t=1 **to** T
        i. compute $c_t = log(\hat{C}^{-1} C_t)$
    b. **end for**
    c. compute $\Delta \hat{C} = exp\left(\frac{1}{T}\sum_{t=1}^{T} c_t\right)$
    d. assign $\hat{C} = \hat{C}\Delta\hat{C}$
8. until $\|log(\ \ )\Delta\hat{C}\| < \epsilon$

FIGURE 9: OBJECT EXTRACTION AND TRACKING

Scene segmentation clusters shots together into semantically themed scenes. Syntactic queues that identify the scene boundaries are hard to identify as different cinematography is applied to different genres and even between different film makers. Within genres and specific footage, rules can be produced for identifying scene boundaries with a high level of precision and recall. However, these rules are limited to their own genre and cannot be applied universally. What is required is an algorithm that can formulate rules for any video clip that is supplied. MAC-REALM uses a Genetic Programming (GP) approach based on work from [Angelides and Lo 2005] that evolves rules from a set of pre-defined features that are good indicators of scene boundaries in general film production. MAC-REALM improves on this by selecting different features which are better indicators and that also reduce the computational

expense of processing the footage to formulate the rules. This approach gives higher precision in identifying scene boundaries, whilst reducing overall processing time.

The scene boundary detection is a semi-automatic process that detects boundaries by using a trained GP algorithm that identifies low level feature combinations that identifies scene boundaries. Due to scene boundaries having a semantic definition, the boundary must be perceived semantically. This means a user must train the GP with a small video clip of pre-identified scene boundaries. The GP then formulates rules that identify certain feature sets, i.e. histogram difference, object number, shot transition type and shot duration. It uses a fitness function based on how well the rule correctly identifies scene boundaries from the training clip. Input for scene extraction is sourced from both shot and object extraction, as well as the content layer. The shot duration and transition type is sourced from shot extraction, whilst the number of objects present in a shot is sourced from the object extraction engine. The histogram values are sourced from the content layer via the shot extraction engine.

MAC-REALM uses instead four video features only, and no audio features. This has been done for two reasons. The first is that although audio features are a good indicator of scene change, they are only as good as the analysis of the features extracted. Voice recognition has to be used, as well as other audio recognition algorithms, as a scene change is usually indicated by a change in actors or environments. In the original work they use speech and audio breaks to formulate rules. Although these are adequate, they do not provide accuracy to the start of a scene. The second is that it reduces computational complexity and therefore allows processing time to be kept to a minimum. All the features used by MAC-REALM to create the rules have already been extracted during the previous processes. The video features that are used in replacement of the audio features are the number of objects in the shot and the global histogram difference of the shot both of which are good indicators of a scene change. The number of objects in a shot is a good indicator to the start of a scene as they usually have a low or fixed number for the establishing shot of the scene. The histogram difference can provide a good measurement for a scene change as the colour distribution for shots belonging to the same scene are more similar to each other than the colour distribution from a shot from another scene. So the feature set to be used as the main parameters of the GP algorithm includes: Shot duration (the shot length in seconds till the start of the next shot), histogram difference (the change in the mean histogram values of the RGB values of the first frame of a shot to a specified preceding/subsequent shot), transition Effect (what transition effect there is between the shots), and number of Objects (how many identified objects are in the shot).

The goal of the GP algorithm is to discover rules that determine scene boundaries. The GP algorithm takes as input a series of shots S1, S2, … SN, and their corresponding features. The choice of features affects directly the result. If not enough features are selected, an optimal rule may never evolve (rules evolve by reproduction, crossover and mutation). On the other hand, if there are too many features, the search space will become inoperably large and seriously affect the processing time of the system. We attribute the following five features with each shot: transitional effect, number of objects, shot duration and histogram difference. Shot duration is measured using the W3C time code from the ISO 8601 Standard [Wolf and Wicksteed 1998]. With regards to the histogram difference, two key frames are extracted from each shot. The first one is extracted from the beginning of the shot and the second one from the end. The first key frame's colour histogram difference with the second key frame from the previous shot is computed using the same formula as the one used in shot boundary detection. The function set of the

algorithm can be defined as $\mathrm{F} = \{SD, HD, TE, NO, AND, OR\}$, where *SD*, *HD*, *TE*, *NO* are the four video features: shot duration, histogram difference, transition effect, number of objects. The terminal sets comprise of

$\mathrm{T} = \{\mathrm{sp}, \mathrm{bv}, \mathrm{pi}, \mathrm{op1}, \mathrm{op2}\}$, where $sp = \{A, B, C, D, E\}$ and is the position of the shot to be compared against the

current shot (C), $\mathrm{bv} = \{\mathrm{true}, \mathrm{false}\}$, pi is a positive integer in {1 – 126789}, $\mathrm{op1}$ in $\{=, \neq\}$ and $\mathrm{op2}$ in $\{<, \geq\}$.

The Scene boundary rules use the grammar provided by reverse polish notation [Visser 2011]. This is convenient because as a last-in-first-out (LIFO) stack is used implementing the stackbuffer method in java. It also makes calculations much more efficient by reducing the complexity of the calculations as all brackets and parentheses are eliminated. After finalising the syntax for the rules, the initial population has to be created to evolve the rules from. The initial population of rules are grown using three different GP strategies. These increase the diversity of the rules and helps evolve more varied and healthier children that are more resistant to convergence of the population. There are three popular generative methods in classic GP: full, grow and ramped half-and-half [Torres et al. 2009]. The full generative method creates a population with full trees whereas the grow method generates the initial population with

trees that are variably shaped. The ramped half-and-half generative method is a combination of the full method and the grow method. The ramped half and half method has a depth limit of five to achieve a reasonable level of diversity. Half of the trees are generated by the full method and half of the tress are created by the grow method. After the rules have been generated, they need to be assessed to see which are better at identifying scene boundaries. A fitness function is used that identifies the rules that are more proficient at finding scene boundaries. The fitness function is given by [Angelides and Kevin Lo 2005]:

$$f = \frac{Nc}{Nt}$$

where Nc is the number of correctly identified scene boundaries, and Nt is the total number of shots. The fitness function gives a score between 0 and 1, with 1 representing the optimal solution. The fitness function evaluates the rule quality, i.e. a rule's performance in determining scene boundaries. Once the fitness of all the rules are assessed a new generation of rules is created that are better adapted to identifying scene boundaries. These rules must carry over the best traits from the existing rules for producing better ones in the next evolution. MAC-REALM uses a method of cloning, mutation, crossover and introducing new rules to facilitate this; the top 10% are copied over to the next generation, whilst the bottom 70% is discarded. The top 30% are mutated to provide new rules, and used in a crossover operation to provide another set of new rules. The last 30% of rules are generated using the same methods as the initial population. This technique of creating new generations allows the properties of the best rules to be favoured in the next cycle of evolution whilst making sure that the population stays diverse enough to stop convergence. Ensuring that suitable divergence is assured is paramount, or the algorithm could converge too early on a less than optimal solution. The algorithm is iterative and will stop either when an optimal rule is obtained, i.e. the fitness value of the rule matches the target fitness value, or the maximum pre-determined number of generations is reached. The optimal rules fitness value limit is set to a minimum of 95%. The maximum number of generations generated is set to 300. Figure 10 shows the key steps of the GP scene boundary detection algorithm. The GP scene boundary detection algorithm of Figure 10 formulates the rules as feature vectors based around video features that are good indicators of scene boundaries regardless of the domain or content. As the rules are judged on a fitness function that uses the training data as ground truth, a rule can be generated that takes into account the abstract semantic nature of the scene boundary. This makes the scene boundaries identified very close to the semantic perspective of users.

STEP	INSTRUCTION
1	GENERATION = 0
2	CREATE INITIAL POPULATION WITH SIZE P
3	APPLY *FITNESS FUNCTION* TO EVALUATE THE FITNESS VALUE OF EACH RULE
4	SORT THE RULES ACCORDING TO THEIR FITNESS VALUE IN DESCENDING ORDER
5	IF TERMINATION CRITERION MET (BEST FITNESS VALUE > 0.95 OR GENERATION > MAX GENERATION K), OUTPUT THE BEST RULE AND EXIT. ELSE GO TO STEP 6
6	THE WORST FIT RULES (THE WORST 70%) ARE DISCARDED
7	GENERATION = GENERATION + 1
8	PERFORM *REPRODUCTION* OPERATION (TOP 10%)
9	PERFORM *CROSSOVER* OPERATION (TOP 30%)
10	PERFORM *MUTATION* OPERATION (TOP 30%)
11	CREATE NEW RULES (30%)
12	GO TO STEP 3 UNLESS A) GENERATIONS = 300 B) A RULE HAS 95% FITNESS SCORE
13	END

FIGURE 10: GP SCENE BOUNDARY DETECTION

3.3.3 Analysis and Linkage Plane: Semantic Relationships

Although semantic relationships have been critical in semantic search and retrieval they have also been noted as playing an important part in concept detection methods [Weiming, Nianhua, Li, Xianglin and Maybank 2011]. The spatial and temporal relationships between content features play an important part in determining the relationships between concepts. From these relationships, knowledge of actions and events can be learnt, and concepts that share similar themes can be grouped and new concepts inferred for the content. This makes accurate modelling of spatial and temporal relationships very important in discovering and learning concepts and ontologies. The MAC-REALM analysis and linkage plane is responsible for modelling the relationships between low and mid-level features extracted in the extraction plane. Existing video indexing and modelling systems model treat spatial and temporal relationships between content features as a post process to modelling that is done in an ad-hoc manner. This may

often lead to ambiguity as different methods for processing spatial and temporal relationships can lead to them being interpreted with different meanings. For uniformity between query results, and for improving concept detection through spatial and temporal concept modelling, having explicitly modelled spatial and temporal relationships would allow the formation of consistent results and concept detection using semantic ontologies over all applications that use a content model. Figure 11 shows the processes of the analysis and linkage plane. Semantic media from the content layer is processed to produce spatial and temporal relationships. The temporal relationships between spatial relationships and other features are also modelled and converted into MPEG-7 content descriptions.

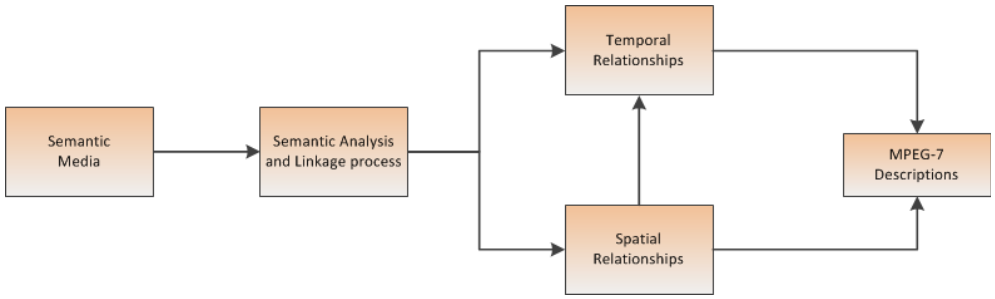


FIGURE 11: SEMANTIC RELATIONSHIPS ANALYSIS AND LINKAGE PROCESS

Defining spatial relationships arbitrarily may lead to ambiguity, as the method of calculating the position of objects varies between systems. This often leads to different applications modelling different spatial relationships for the same content, which in turn may lead to inaccurate query results. Spatial relationships usually need to fulfil two criteria; calculating the reference point for the centre of an object in a consistent and natural manner that makes the resulting measurements logical and intuitive in relation to queries and designating relationships for both global positions of individual objects and also for relative positions of objects to each other. MAC-REALM uses a centroid algorithm that can work out the centre of an irregular shaped lamina, and then calculates the absolute and relative positions of the object using standard techniques. Hence, modelling of spatial relationships in MAC-REALM begins with finding the centroid of object(s) which is used as the reference point to measure their position. In order to find the centroid of an object the boundary of the object must be defined. Object silhouettes are retrieved from the semantic media module and are used as object boundaries. They are analysed frame-by-frame with each object's edge boundary Cartesian coordinates used as the input for the centroid algorithm.

Spatial relationships are of two classification types: absolute and relative. Absolute spatial relationships are modelled using the points of the compass and are precise about location in terms of description. This is done for each object independently and gives a spatial orientation that is dependent on its global orientation within a frame. Relative spatial relationships are modelled in terms of an object's position in relation to another object. Examples of this are object 1 is ***above*** object 2 and object 1 is on the ***right*** of the screen. In order to calculate absolute spatial relationships we split the screen according to the visual composition rule known as the "rule of thirds" [Liu et al. 2010]. The screen is split into 9 different sections and forms a 3x3 matrix. Each position in the matrix has an absolute spatial relationship attached to it depending on its absolute spatial orientation within the matrix P, such that:

$$P_{(i,j)} = \begin{Bmatrix} NW & N & NE \\ W & C & E \\ SW & S & SE \end{Bmatrix}$$

The absolute position of an object is given by the placement of the centroid within the matrix's boundary. If the object is in the middle of the matrix its position is designated as "centre". In order to calculate the relative spatial relationships of objects two methods are employed: generalised position and relative to another object. With the first method two matrices are used, one for vertical positions and the other for horizontal positions. This is done to model the spatial relationship of an object in both planes. An object is considered as having a single position, i.e. in one plane only, when in a central or neutral position in the other. If V and H are 3x3 matrices and have members:

$$V_{(i,j)} = \begin{Bmatrix} T & T & T \\ 0 & 0 & 0 \\ B & B & B \end{Bmatrix}, H_{(i,j)} = \begin{Bmatrix} L & 0 & R \\ L & 0 & R \\ L & 0 & R \end{Bmatrix}$$

where $T = top, B = bottom, L = left$ and $R = right$. The combined relative position in both planes can be calculated by adding the two matrices together:

$$(V+H)_{(i,j)} = \begin{Bmatrix} TL & T & TR \\ L & 0 & R \\ BL & B & BR \end{Bmatrix}$$

There is no "centred" position with regards to relative spatial positions as there is no relative centre as the both objects positions are arbitrary. For the relative positioning between two objects, A and B, the cardinal point positions of one object in relation to another are given in degrees. North is taken to be $\theta = (0°, 360°)$. For any cardinal point, any $22.5°$ angled section from that point, both clockwise and anti-clockwise, can be considered as having the same bearing as that cardinal point. Each cardinal point is given the same $45°$segment around its point. The half cardinal points are given to the boundaries of each main segment. This arrangement gives the best mapping cognitively to what users perceive when they think of direction. Therefore, the cardinal point sections are represented by:

$$B = \begin{pmatrix} 315° < 45° & = & N \\ 45° & = & NE \\ 45° \leq 135° & = & E \\ 135° & = & SE \\ 135° \leq 225° & = & S \\ 225° & = & SW \\ 225° \leq 315° & = & W \\ 315° & = & SW \end{pmatrix}$$

The inverse bearing is given by:

$$B^{inverse} = \begin{pmatrix} if\ \ B \leq 180° \ \therefore B + 180° \\ if\ \ B > 180° \ \therefore B - 180° \end{pmatrix}$$

Temporal relationships provide a linking mechanism between syntactic and semantic features which allows making connections between the physical structure of the video and the meaning of the content. MAC-REALM's explicit media structure enables temporal relationships between the syntactic features of scene, shot and object to be determined through a partial temporal ordering of these entities. A partial ordering < can be defined on a set of features as follows:

$$[i_1, j_1] < [i_2, j_2] \ \textbf{\textit{if}}\ i_1 \leq i_2 \ \therefore j_1 \leq j_2$$

where $i_1 =$ start of syntactic features and $i_2 =$ end of syntactic features thus syntactic features can be ordered according to each associated feature's i value such that the feature denoted by $[i_1, j_1]$ precedes the feature denoted by $[i_2, j_2]$.Partial ordering enables MAC-REALM to determine which content features occur before or after other content features and which intersect or occur simultaneously, defined as $[i_1, j_1] \subseteq [i_2, j_2]$). For example, once partially ordered, to determine if a syntactic feature, A, occurs before or after a given group of syntactic features, B, the i value of A is compared to the i value of the first syntactic feature within B. If it is smaller, then A occurs before B. However, if the j value of A is greater than the j value of the last syntactic feature within B, then A occurs afterwards. Similarly, two syntactic features, $S_1 = [i_1, j_1]$ and $S_2 = [i_2, j_2]$, can be compared to determine if they intersect, which will be in one of five ways; (1) $i_1 = i_2$ and $j_1 < j_2$, (2) $i_1 < i_2$ and $j_1 = j_2$,(3) $i_1 < i_2$ and $j_1 < j_2$,(4) $i_1 = i_2$ and $j_1 = j_2$ and (5) $i_1 < i_2$ and $j_1 > j_2$. With these temporal ordering rules, it is

possible during querying, for example, to determine the next group of syntactic features within a given set of syntactic features that occur simultaneously. This takes place as follows. Once all time stamps for the start and finish of each syntactic feature are collected, those constituent syntactic features that occur within a specific parent syntactic feature are partially ordered. The next group of syntactic features is determined by taking the first syntactic feature and then adding to the group those syntactic features whose j values are not greater than the j value of the first syntactic feature. These syntactic features are thus those that occur simultaneously with the first syntactic feature.

The semantic relationships produced during the content analysis and linkage phase need to be referenced in order to show not just the relationships between the spatial and temporal relationships of the low level features (i.e. scenes, shots and objects), but also the temporal relationships of the spatial relationships between themselves and the low level features. Using the SemanticDescriptionType DS a referencing system can be constructed using the Graph DS that allows for the flexibility and grammar needed to achieve such a referencing system, and also use MPEG-7 classification schemes to define the relationships between entities. The Graph DS describes language-independent terms for use in multimedia descriptions and schemes for classifying a domain using a set of such terms. Using the node graph structure allows both syntactic and semantic features to be named in a manner that is independent of their abstract type and attributes, and thus makes stating the relationships between features of heterogeneous origin uniform and standard. The ClassificationScheme DS describes a vocabulary for classifying a subject area as a set of terms organized into a hierarchy. A term defined in a classification scheme is used in a description with the TermUse or ControlledTermUse datatypes. In the instance of referencing all low and high level features with a homogenous referencing system, the Graph DS allows us to create nodes that identify each feature set using the Node D tool. This tool allows us to assign a unique id tag to all low level and high level features that is related to the temporal instance of that feature. The low level features are referenced using their id tag from their original feature description. In the case of the spatial relationships the id tag from their original feature of the first object is used because spatial relationships only need a time reference point to be identified with, since a spatial relationship is compared in terms of its temporal relationship to other features. An example of scenes modelled into nodes is given in Figure 12.

```
<Description xsi:type = "SemanticDescriptionType">
     <Semantics>
           <Labels>
               <Name>Nodes for Temporal/Spatial Relationships</Name>
           </Labels>
           <Graph>
               <Node id = "SC1" href="AVP-SCENE-1"/>
               <Node id = "SC2" href="AVP-SCENE-2"/>
               <Node id = "SC3" href="AVP-SCENE-3"/>
               <Node id = "SH1" href="AVP-SCENE-0-SHOT-0"/>
               <Node id = "SH2" href="AVP-SCENE-0-SHOT-1"/>
                ..............
               <Node id = "SH28" href="AVP-SCENE-0-SHOT-27"/>
               <Node id = "SH29" href="AVP-SCENE-1-SHOT-28"/>
               <Node id = "SH30" href="AVP-SCENE-1-SHOT-29"/>
               <Node id = "SH31" href="AVP-SCENE-1-SHOT-30"/>
                ......................
               <Node id = "OB1" href="AVP-SCENE-0-SHOT-0-OBJECT-1-13"/>
               <Node id = "OB2" href="AVP-SCENE-0-SHOT-1-OBJECT-1-41"/>
               <Node id = "OB3" href="AVP-SCENE-0-SHOT-2-OBJECT-1-51"/>
               <Node id = "OB4" href="AVP-SCENE-0-SHOT-3-OBJECT-1-73"/>
                .........
               <Node id = "SR1" href="AVP-SCENE-0-SHOT-10-OBJECT-1-160"/>
               <Node id = "SR2" href="AVP-SCENE-0-SHOT-12-OBJECT-1-182"/>
               <Node id = "SR3" href="AVP-SCENE-0-SHOT-13-OBJECT-1-213"/>
               <Node id = "SR4" href="AVP-SCENE-0-SHOT-14-OBJECT-1-270"/>
               <Node id = "SR5" href="AVP-SCENE-0-SHOT-17-OBJECT-1-341"/>
            </Graph>
       </Semantics>
```

FIGURE 12: LOW AND HIGH LEVEL FEATURES REFERENCED IN MPEG-7

Spatial relations are modelled using the SpatialRelation CS, which defines all spatial relationships that are describable in MPEG-7. Typecasting the Description DS to “SemanticDescriptionType” allows for the description of

the spatial relationships between objects. Using the Semantics DS, objects are stated and the spatial relationships described between them. The spatial relations graph is labelled using the Label DS within this element. The Graph DS is then used to describe the spatial relationships between those objects. The Relation DS is used to describe the spatial relationship between two objects. The spatial relationship is stated using the SpatialRelation CS, which defines the relationship in terms of a source node applied to a target node. The node structuring allows for a flexible and clearer way of describing relationships than if stating them directly. An example of MPEG-7 SpatialRelationship CS is given in Figure 13 that shows the relative spatial relationships between objects.

Temporal relationships are modelled in much the same manner as spatial. Once again we typecast the Description DS to "SemanticDescriptionType" to indicate the following graph is describing high level features. The graph is labelled using the Label DS to identify it as a temporal relationship graph. In a manner similar the Relation D within the Graph DS is used to describe the relationships. The difference is that now the MPEG-7 TemporalRelation CS is used to typecast the graph as containing temporal relationships. Using the aforementioned referencing system the temporal relationships are described between both homogeneous and heterogeneous content type feature sets. In Figure 15 an example of the variety of different types of temporal relationship is shown between homogeneous content type feature sets. The nodes in Figure 15 represent scenes and shots content descriptions. In Figure 14 an example is given of temporal relationships modelled between heterogeneous content feature types, whereby the nodes represent shot and spatial relationships. From the two examples of homogeneous and heterogeneous content feature types it can be seen that the nodes provide a proxy representation of the features. This abstract representation of the feature sets allows temporal comparison and facilitating the requirement of multi-content type search.

```
<Description xsi:type = "SemanticDescriptionType">
  <Semantics>
    <Labels>
      <Name>Spatial Relationships</Name>
    </Labels>
    <Graph>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB11"
target = "OB12"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB13"
target = "OB14"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:south" source ="OB15"
target = "OB16"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB17"
target = "OB18"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB21"
target = "OB22"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:east" source ="OB23"
target = "OB24"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB25"
target = "OB26"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB28"
target = "OB29"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB41"
target = "OB42"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB46"
target = "OB47"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB55"
target = "OB56"/>
         <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB60"
target = "OB61"/>
          <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB64"
target = "OB65"/>
         <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB64"
target = "OB65"/>
      <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB69"
target = "OB70"/>
    </Graph>
  </Semantics>
</Description>
```

FIGURE 13: SPATIAL RELATIONSHIP REFERENCED IN MPEG-7

```
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SR13" target =
"SH68"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SR13" target =
"SH69"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:coOccurs" source ="SR13" target =
"SH70"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" source ="SR13" target =
"SH71"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metBy" source ="SH71" target =
"SR13"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SR13" target =
"SH72"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SR13" target =
"SH73"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SR13" target =
"SH74"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SR13" target =
"SH75"/>
```

FIGURE 14: TEMPORAL RELATIOSHIPS BETWEEN HETEROGENEOUS CONTENT TYPE FEATURE SETS

```
<Description xsi:type = "SemanticDescriptionType">
  <Semantics>
    <Labels>
      <Name>Temporal Relationships</Name>
    </Labels>
    <Graph>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" source ="SC1" target =
"SC2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metBy" source ="SC2" target =
"SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SC1"
target = "SC3"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC2" target
= "SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" source ="SC2" target =
"SC3"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metBy" source ="SC3" target =
"SC2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC3" target
= "SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC3" target
= "SC2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC1" target
= "SH1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC1" target
= "SH2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains" source ="SC1"
target = "SH29"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:starts" source ="SC1" target
= "SH29"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:startedBy" source ="SH29"
target = "SC1"/>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains" source ="SC1" target
= "SH30"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:strictDuring" source ="SC1"
target = "SH30"/>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:strictContains" source ="SH30"
target = "SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains" source ="SC1"
target = "SH31"/>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:strictDuring" source ="SC1"
target = "SH31"/>
```

FIGURE 15: MPEG-7 TEMPORAL RELATIONSHIPS WITHIN A HOMOGENEOUS FEATURE SET

3.3.4 Modelling plane: Content Modelling

The primary goal of MAC-REALM, which is the innovation behind its development, is to merge all content descriptions into a single content model. Only then is the full potential of the content descriptions achieved, as the content model gives all features a context and relationship to the structure and meaning of content. The final content model links all features together through a hierarchical structure and should provide mechanisms for all the features to be interlinked into a flat structure to optimise search capabilities and content discovery. The MAC-REALM modelling algorithm achieves this by layering the content features to the root node so that the top-level container for each feature is only one link away from any other top-level container. This makes the content more easily searchable and content discovery multi-faceted as the features are not rigidly structured in a nested tall structure. MAC-REALM's content model facilitates generic use through the structure and interlinking of its content descriptions.

The modelling plane has three distinct sections: Syntactic and Semantic Descriptions, Content Modelling, and Model media. The first section represents the MPEG-7 output from the extraction and analysis and linkage planes. These descriptions are modelled in the MPEG-7 layer. The second section parses all MPEG-7 descriptions and then combines them into one DOM. The DOM is then serialised to the final stage, which is the model media, the final process of MAC-REALM, that outputs an MPEG-7 model that can be used by any search/filtering application that is MPEG-7 compliant.

The syntactic and semantic content description schemes are linked by two methods that allow the multimedia content to be integrated so that the semantic gap can be bridged. The first method allows the syntactic and semantic content features to be linked on a semantic level. The semantic linking mechanism is provided by modelling all the features into nodes that represent all content features abstractly. The nodes are then used as proxy representations for the features and the temporal relationships between all these features are modelled allowing direct comparison between all content features, regardless of content type. The second method for facilitating multi-content type search is to model the syntactic and semantic content descriptions together into a content model with all the features interlinked through their logical dependencies. The features were extracted using a hierarchical input/output extraction process where each feature extracted was used as input for the next feature. Due to the extraction process all features are intrinsically and implicitly linked together as the features have many attributes in common.

The scenes and shots use the media time to link them together and are naturally nested within each other, as scenes consist of shots. The objects are created from the shots and are linked to them through their id reference attributes. All the syntactic features are then modelled into nodes. The object nodes are used to model the spatial relationships, which implicitly link the spatial relationships to the shots and scenes through inheritance of attributes. The nodes of all features are then modelled into temporal relationships, providing semantic linking of all features. The arrangement of the linking mechanisms throughout the content makes joint syntactic and logical content based video search more effective as one search parameter can be applied to any amount of content features simultaneously.

MAC-REALM unifies the syntactic and semantic content using these linking mechanisms. The framework is set by defining the top level elements that state this MPEG-7 document relates to content description of the structural and conceptual content of video. Within this modelling structure both types of content can be defined and linked together, specifically using MPEG-7 part 5 MDS descriptions. The first structural elements to be defined are the top level elements, as these are the skeleton of the content model and establish MPEG-7 compliance. The Multimedia DS, which is typecast to video, is the anchor element for both the syntactic and semantic content description schemes. There are two description schemes that relate directly to the Multimedia DS, as they describe two global values associated with the video: MediaLocator DS and MediaTime DS. The MediaLocator DS contains the MediaURI D which describes the physical location of the media. The MediaTime DS uses the MediaTimePoint DS and MediaDuration DS to describe the global start time of the media and its duration respectively.

Within the structural description schemes there are three main description schemes anchored to the Multimedia DS: AnalyticEditingTemporalDecomposition DS (container element for Shots DS), VideoSegmentTemporalDecomposition DS (scenes) and MovingRegion DS (objects). The VideoSegmentTemporalDecomposition DS defines scenes through VideoSegment DS child nodes. Within the VideoSegment DS, which represents the scenes directly, there is anchored the

AnalyticEditingTemporalDecomposition DS that contains the shots for a particular scene. The AnalyticEditingTemporalDecomposition DS can also be a direct node from the Multimedia DS that can represent shots that do not belong to a scene. The AnalyticEditingTemporal-Decomposition DS can only be instantiated once as a top level structural type, unlike the VideoSegmentTemporalDecomposition DS, but can appear many times under the VideoSegment DS, on a one-on-one basis per instantiation of the VideoSegment DS. The MovingRegion DS, which represents objects, is rooted to the Multimedia DS and treated as a structural top level type. Each object is represented by its own instantiation of a MovingRegion DS. Each object can appear as an instantiation of the MovingRegion DS as a top level node as many times as is necessary. Alternatively if there aren't any objects, then there will be no MovingRegion DS instantiations. All semantic relationships of the content model are anchored to the Multimedia DS through the SemanticDescriptionType DS, which is cast through the abstract Description DS. Whereas the structural components had used time as the basis of their hierarchical structure, within the SemanticDescriptionType DS the graph structure of semantic relations is used. The Graph DS is the only child node of the Semantic DS and is instantiated for both spatial and temporal relationships. The Shot DS, VideoSegmentTemporalDecomposition DS and the MovingRegion DS are all model into Node D's. Node D allows making relationships not between different content type features and allows the content model to detail the intricacies of relationships between syntactic and semantic content features. Figure 16 shows an entity diagram of the unified content model with relationships between the top level document and syntactic and semantic nodes within. Figure 17 depicts an example of a simplified MAC-REALM content model.

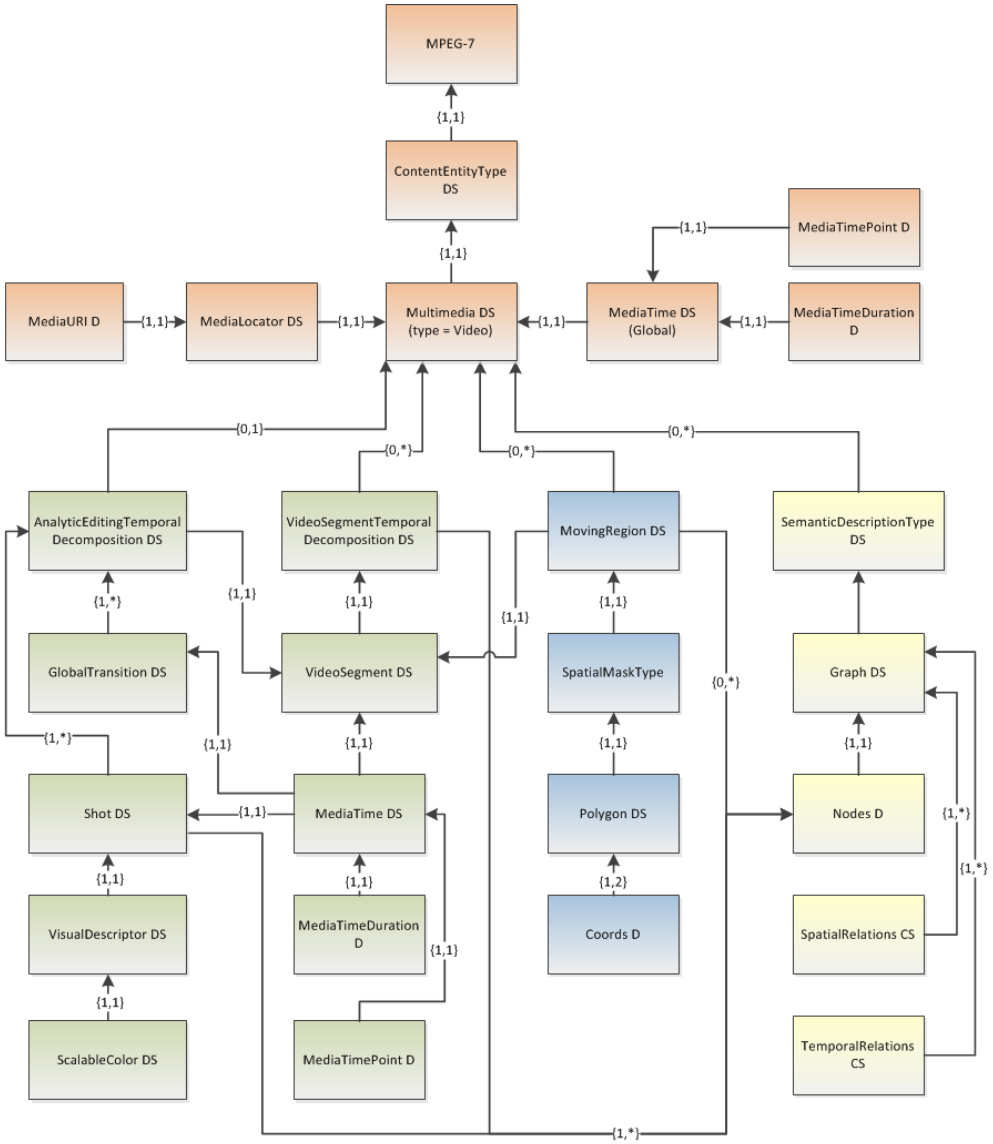


FIGURE 16: UNIFIED MPEG-7 CONTENT MODEL ENTITY DIAGRAM

```
<Mpeg7xmlns="urn:mpeg:mpeg7:schema:2001"xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"xmlns:schemaLocation="urn:mpeg:mpeg7:schema:2001:Mpeg7-
2001.xsd">
     <Description xsi:type="ContentEntityType">
       <Multimedia xsi:type="VideoType">
         <Video>
            <MediaLocator>
              <MediaURI>F:/AVP/Video/AVP_test.mpg</MediaURI>
            </MediaLocator>
            <MediaTime>
              <MediaTimePoint> PT1H5M3S0N25F </MediaTimePoint>
              <MediaDuration>PT25M35S20N25F</MediaDuration>
            </MediaTime>
            <AnalyticEditingTemporalDecomposition>
             ……..
               <Shot id = "AVP-SCENE-0-SHOT-1">
                ……..
            </AnalyticEditingTemporalDecomposition>
            <VideoSegmentTemporalDecomposition>
               <VideoSegment id = "AVP-SCENE-1">
               <AnalyticEditingTemporalDecomposition>
                  ……..
                 <Shot id = "AVP-SCENE-1-SHOT-1">
                  ……..
               </AnalyticEditingTemporalDecomposition>
            </VideoSegmentTemporalDecomposition>
            <VideoSegmentTemporalDecomposition>
              <VideoSegment id = "AVP-SCENE-2">
            </VideoSegmentTemporalDecomposition>……..
            <MovingRegion id = "AVP-SCENE-0-SHOT-1-OBJECT-1-41">
             ……..
            </MovingRegion>
             ……..
            <MovingRegion id = "AVP-SCENE-3-SHOT-13-OBJECT-2-786">
             ……..
           </MovingRegion>
           <Description xsi:type = "SemanticDescriptionType">
             <Semantics>
               <Graph>
                 <Node id="SC1" href="AVP-SCENE-1"/>
                  ……..
                 <Node id="SH203" href="AVP-SCENE-3-SHOT-1"/>
                  ……..
                 <Node id = "OB50" href="AVP-SCENE-2-SHOT-55-OBJECT-1-3463"/>
                  ……..
                 <Node id = "SR13" href="AVP-SCENE-2-SHOT-69-OBJECT-1-5908"/>
               </Graph>
             </Semantics>
             <Semantics>
               <Graph>
Relation type = "urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB17" target = "OB18"/>
               </Graph>
             </Semantics>
             <Semantics……..
               <Graph>
Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows"source ="SH180" target = "SR15"/>
               </Graph>
             </Semantics>
           </Video>
       </Multimedia>
     </Description>
  </Mpeg7>
```

FIGURE 17: UNIFIED MPEG-7 CONTENT MODEL SKELETON

The content model begins with the top level description schemes that provide the anchor for any MPEG-7 standard content model. Using the attribute declaration "type" within the Multimedia DS, the element is typecast as "VideoType". The child element of the Multimedia DS is the Video DS that encompasses all the syntactic and semantic content description schemes and relates them to the description of video. The Video DS contains the MediaLocator DS, and the MediaTime DS, stating the physical location and global time properties of the media respectively. After the content model container is initialised the first content features to be added are the shots that do not belong to a scene. These orphan shots are modelled within the top level AnalyticEditingTemporalDecomposition DS element. Within this element the Shot DS's are modelled with an id reference that consists of the name of the movie, a scene number of '0' and the shot number, all delimited by a hyphen e.g. "AVP-SCENE-0-SHOT-1". Next the scenes are modelled, which are explicitly stated within VideoSegmentTemporalDecomposition DS via a single child VideoSegment DS. The VideoSegment DS id attribute is given the scene number of the clip in the same manner as the orphan shots but with the shot information omitted, e.g. "AVP-SCENE-1". Within the VideoSegment DS the shots for the scene are represented via a child AnalyticEditingTemporalDecomposition DS. The Shot DS's are referenced as before but have a scene number that references the scene number from the corresponding parent VideoSegment DS. This way the scenes, shots and orphan shots can all be searched using the same search parameters. The id reference for scenes and shots is extended for objects and used in the MovingRegion DS to include the object number and frame the object first appears in, e.g. "AVP-SCENE-0-SHOT-1-OBJECT-1-41". This provides the link between scenes/shots and objects.

This naming convention for scenes, shots and objects is then used as the input for the Semantics DS relations structure, via the Graph DS. The referencing mechanism employed allows for the relationships between heterogeneous and homogenous content features to be explored and stated without the attributes associated with content feature or content type. This abstract approach to syntactic and semantic content type linkage opens up the opportunity to explore relationships between syntactic and semantic information that are more informative. They provide a more cognitive approach to the content model that is more holistic and true to the content and the myriad ways a user perceives and searches content.

4. PERFORMANCE EVALUATION

We use four 20min video clips from Alien vs. Predator (AVP) [W.S. Anderson 2004] to test how accurately MAC-REALM extracts syntactic features from content. AVP is used because it includes objects on screen that are invisible to the human eye. The presence of invisible objects provides the hardest footage to test MAC-REALM with. All video clips are digitised in MPEG-1 format at a frame rate of 25 fps (total of ~720,000 frames) and a resolution of 352*288 pixels (commonly known as the CIF standard [Richardson 2010]). This was carried out using Adobe Premiere Pro. The tests were run three times and the median average is taken for all three runs.

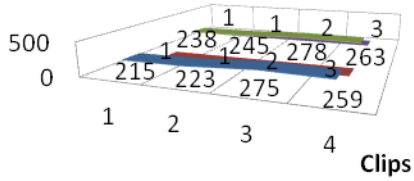
4.1 Shot Boundary Detection

Before beginning the experiments, the segmentation algorithm is tuned on a number of small (< 10 minute) video segments extracted from the test set. These training runs enable fine-tuning of the adaptive threshold levels for each clip. The experiment is conducted with the four sample clips, and the results are depicted in Figure 18, alongside the number of groundtruth shots. Detection rates are provided separately for both cut and gradual transition shots. All the gradual transition shots are detected with a 100% recall. The cut shots have variable rates of detection: clip 1 = 90.34%, clip 2 = 91.34%, clip 3 = 98.92% and clip 4 = 98.48%. Clips 1 and 2 are poorly-lit scenes and, therefore, the colours in them are not as vivid as in clips 3 and 4. If they have then they would have similar detection rates. The gradual transitions rely on edge information, which although it is diminished, it is still able to accurately detect the gradual transitions. In Figure 19 we can see how many shots detected are correctly identified and how many are incorrectly labelled as shot boundaries. From these results, $TP = 941, FP = 42, FN = 52$ and $P = 1031$. From this we calculate that the recall, precision and $f1$ score of the shot detection algorithm as:

$$Precsion = \frac{941}{941+42} = 95.73\%, \text{Recall} = \frac{941}{1031} = 91.27\%, f1 = \frac{2 \times 95.73 \times 91.27}{95.73 + 91.27} = 93.44$$

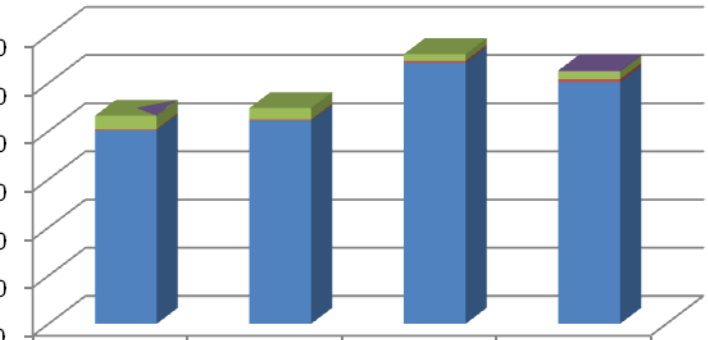
In order to provide a comparison, MAC-REALM's shot detection rate is compared to a similar automatic feature extraction and content modelling system called SHIATSU (Semantic Hierarchical Automatic Tagging of videos by

Segmentation Using cuts) [Bartolini et al. 2011], which also extracts shots automatically using a double dynamic threshold system that implements a hybrid HSV based CHD and ECR. Both techniques are used for detecting cut shots but only ECR is used for detecting transition shots. The recall rate for SHIATSU is 85.8%, with a precision of 69.22%. This demonstrates that the adaptive thresholding technique used in MAC-REALM yields better results than the double dynamic thresholding technique used in SHIATSU.



	1	2	3	4
MAC-REALM Cuts	215	223	275	259
MAC-REALM Transitions	1	1	2	3
Baseline Cuts	238	245	278	263
Baseline Transitions	1	1	2	3

FIGURE 18: Shots discovered per clip



	1	2	3	4
False Transitions	0	0	0	1
False Cuts	14	12	7	8
Correct Transitions	1	1	2	3
Correct Cuts	201	211	271	251

FIGURE19: Number of shots correctly identified

4.2 Object Detection

To detect an object, the object detection method requires an object to be segmented accurately so that the contour of the object is the boundary of the object, and that the object boundary is tracked accurately once segmented. The methodology employed to measure the object segmentation, uses 4 randomly selected objects. These objects, the intersection between them and the groundtruth samples are recorded. The accuracy calculated for all four objects is presented in Figure 20. The results for sample selections of the object extraction are shown in Figure 21, with each row showing the results for a key frame of a shot. The original colour images are shown on the far left of each row.

The user-defined label traces overlaid on the image are shown in the left middle column of each row. Each colour represents a different label, object 1 is red and the background is yellow. Green is used as the colour for second objects. The object segmentation is shown in the third from left of each row. On the far right of each row we have a colour map of the objects, clearly showing the boundaries of each object and the backgrounds through colour.

The image regions may present similar grey-levels due to dark scenes and belong to different model classes defined by the user labels. Also, there are some image regions with substantial grey-level variation because of belonging to non-homogeneous textured regions, which are traditionally very difficult to segment. The structural information leads to a robust segmentation performance even in such cases. For brighter regions with well contrasted boundaries the segmentation has accuracies of between 0.97 – 0.998. We can see that an object has bled into the "letterbox" lines of the image in frames B, D and E. These lines are never traced as background and so are not eliminated. If done so they would have been removed too. The rough tracing of objects has led to some objects edges not being defined, as in C and E. The same four objects are tracked. These do not handle scenarios of occlusion and partial occlusion. FIGURE 22 shows the OTE calculated for the four objects. Tracking is good for 1 and 4 as the object contours are well-defined and their motion smooth. Object 2 has problems with tracking as it is in a fast moving scene and there is some motion blur that affects the integrity of the object boundary. For object 3 the problem is that the object has not been segmented well and therefore the tracking is erroneous.

OBJECT	ACCURACY
1	0.87
2	0.80
3	0.73
4	0.83

FIGURE 20: Segmented Object Accuracy

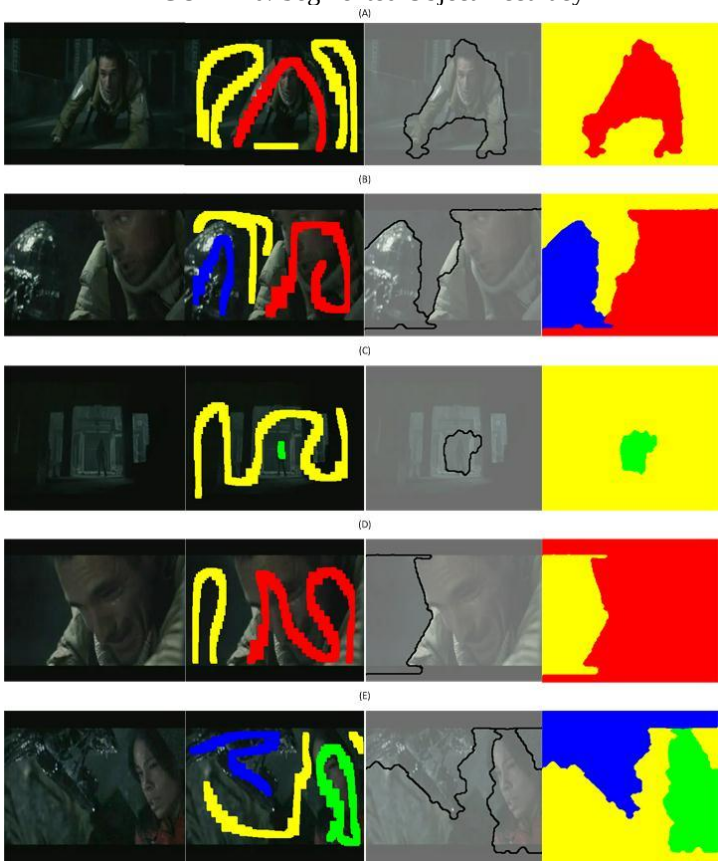


FIGURE 21: Examples of objects extracted from images

OBJECT	OTE
1	0.98
2	0.75
3	0.55
4	0.93

FIGURE 22: OTE for segmented objects

Object detection has been compared to the techniques used by similar systems that also extract and model objects. In [Dyana et al. 2009] they use Gaussian Mixture Models (GMM) to model the background and foreground objects. Similar rates of object detection are achieved consistently by GMM for video (80%) that has consistent background (e.g. CCTV). MAC-REALM outperforms GMM on video that includes moving shots or background that changes radically over time, with the object detection rate falling to (45%). This shows that MAC-REALM yields better results in detecting objects in generic video where the background often changes randomly. Tracking of objects shows the OTEs for both systems are comparable with OTEs on average at around 75%. BilVideo-7 [Baştan, Çam, Güdükbay and Ulusoy 2010] uses a flood fill algorithm to detect objects. Anthropos-7 [Tsingalis et al. 2012] and DanVideo [Kannan, Andres and Guetl 2010] both use manual input to segment the objects, so although extremely accurate, it is very time consuming and is prone to human error.

4.3 Scene Detection

Scene detection is tested by using the first clip of AVP as the training data for the GP algorithm. The resulting rule is then used on the remaining three clips to ascertain how well the clips are segmented into scenes. The four features used in the GP algorithm, i.e. shot duration, number of objects, colour histogram and shot transition, is the result of shot and object detection on the four AVP clips presented as java data structures serialised to data text files. Each test is run with parameters p (population size) = 500, k (maximum generation) = 300 and f (maximum fitness) set at 98%. The experiment is run three times over the same dataset. An optimal rule is found with 98% fitness around 118 – 120 generations. The best machine-generated rule from each run is shown in Figure 23 in Reversed Polish Notation. The rules are applied to testing data for measuring its accuracy. We use the same performance measures used for shot detection, precision and recall, to evaluate the accuracy of the rule. The methods have been used extensively to compare the performance of shot boundary detection techniques. Since the nature of scene boundary detection is similar to shot boundary detection, it is plausible to use the method as well without any modification. There are 786 shots in total in the three AVP clips chosen to be segmented. Among them, there are 23 scene boundaries (manually counted) and the rule has discovered 21. But only 13 of them are correct, so there are 8 false alarms. Of the 21 scenes found only 13 have correct boundaries, which means it has missed 8. Recall and precision are computed as:

$$Recall = \frac{\frac{Number\ of\ scenes\ correctly\ identfied\ by\ MAC-REALM}{Total\ number\ of\ baseline\ scenes}}{} = \frac{13}{23} = 56.5\%, \quad Precision = \frac{\frac{Number\ of\ scenes\ correctly\ identfied\ by\ MAC-REALM}{Total\ number\ of\ scenes\ identfied\ by\ MAC-REALM}}{} = \frac{13}{21} = 61.9\%$$

From this we can calculate the $f1$ score for the scene boundary detection algorithm:

$$f1 = \frac{2 \times 61.9 \times 56.5}{61.9 + 56.5} = 59.08$$

The performance for the GP algorithm gets better using the four video features compared to two video and two audio features for the sample clip. When tested using the audio/video feature combination 23 clips are identified and only 10 were correct with a rule with 96% fitness. The results for this test are 47.6% for precision and 43.5% for recall, giving an f1 = 45.45. One of the possible reasons for this is because AVP does not have much dialogue and long pauses of silence for suspense, making audio breaks rare. This is extraction and modelling technique is unique to MAC-REALM. Bilvideo-7 [Baştan, Çam, Güdükbay and Ulusoy 2010] does model scenes but this is done manually using an annotation tool.

Best Rule 1	(((dC03<cA75<dA39<&)((cC04>cA85<cA09<^)(dE03<cA78<cA39<&)&)&)((dE01<cA79<cA33<&)((cC04>cA42<cA09<^)((dC03<cA71<cA39<&)(((cE03<cA78<cA39<&)((((cC04>cA82<cA09<^)(cD04<cA72<cC09<&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)^)((bC04>cA52<cA09<&)((cC04>cA42<cA09<^)((cC04>cA42<cA09<^)((dC03<cA75<cA39<&)((dE01<cA79<cA39<&)(((dC03<cA78<cA39<&)(((cC04>cA72<cA08<&)(((cC04>dA72<cA04<&)(dC03<cA78<cA39<&)^)(dC03<cA78<cA39<&)&)&)((dB03<dA78<cA39<&)(dB01<cA79<cA39<&)^)&)&)((((cC04>dA72<cA04<&)((cC04>cA72<cA08<&)(dC03<cA78<cA39<&)&)^)(cC04>cA72<cA08<&)&)((cC04>cA72<cC09<&)(dC03<cA78<cA39<&)&)&)&)&)&)&)&)&)&)&)	118
Best Rule 2	(((dC03<cA75<dA39<&)((cC04>cA85<cA09<^)(dE03<cA78<cA39<&)&)&)((dE01<cA79<cA33<&)((cC04>cA42<cA09<^)((dC03<cA71<cA39<&)(((cE03<cA78<cA39<&)((((cC04>cA82<cA09<^)(cD04<cA72<cC09<&)&)((cC06>cA72<cC09<&)(cD04<cA72<cC09<&)&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)^)((bC04>cA52<cA09<&)((cC04>cA42<cA09<^)((cC04>cA42<cA09<^)((dC03<cA75<cA39<&)((dE01<cA79<cA39<&)(((dC03<cA78<cA39<&)(((cC04>cA72<cA08<&)(((cC04>dA72<cA04<&)(dC03<cA78<cA39<&)^)(dC03<cA78<cA39<&)&)&)((dB03<dA78<cA39<&)(dE01<cA79<cA39<&)^)&)&)((((cC04>dA72<cA04<&)(cC04>cA72<cA08<&)^)(cC04>cA72<cA08<&)&)(cC04>cA42<cA09<^)&)&)&)&)&)&)&)&)&)&)&)&)&)&)	120
Best Rule 3	(((dC03<cA75<dA39<&)(dC03<cA78<dA19<&)&)((dE01<cA79<cA33<&)((cC04>cA42<cA09<^)((dC03<cA71<cA39<&)(((cE03<cA78<cA39<&)((((cC04>cA82<cA09<^)(cD04<cA72<cC09<&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)^)((bC04>cA52<cA09<&)((cC04>cA42<cA09<^)((cC04>cA42<cA09<^)((dC03<cA75<cA39<&)((dE01<cA79<cA39<&)(((dC03<cA78<cA39<&)(((cC04>cA72<cA08<&)(((cC04>dA72<cA04<&)(dC03<cA78<cA39<&)^)(dC03<cA78<cA39<&)&)&)((dB03<dA78<cA39<&)(dE01<cA79<cA39<&)^)&)&)((((cC04>dA72<cA04<&)(cC04>cA72<cA08<&)^)(cC04>cA72<cA08<&)&)(cC04>cA42<cA09<^)&)&)&)&)&)&)&)&)&)&)&)&)&)	119

FIGURE 23: Best Scene boundary change - Rules generated by GP for detecting scene changes in AVP

4.4 Spatial relationships

The results of the object extraction process are used as the input dataset for the spatial relationship analysis. To be able to accurately calculate the spatial relationships the centroid of the objects is calculated in order to provide a reference point of the objects. A groundtruth of spatial relationships is produced for all objects by visually examining all shots and manually determining where the centre of the main body of an object lies and stating its absolute position. If two objects are within the shot frame then a relative position is calculated for them. Figure 24 shows that all absolute and relative spatial relationships are correctly identified and all spatial relationships are correct according to the groundtruth observations for spatial relationships. Figure 25 depicts a list of absolute spatial relationships derived by the spatial relationship analyser. Using the centroid function as the point of calculation for spatial relationships it holds true for the dataset used. When absolute spatial relationships are analysed many of the objects are only marginally present within the boundary of the particular position they are allocated. Many of the objects' central mass, not their central point, are in the centre of the screen. A more accurate derivation for a number of spatial relationships would be more accurately described as "central". Seeing that "central" or "centre" are not in the MPEG-7 spatial relationship CS, this could not be used. In Figure 26 we have a grid of relative spatial relationships. The relative spatial relationships have a fuzzier categorisation as they can have both a horizontal and vertical element for their relative positioning. Of the 270 relative spatial relationship's 60% are single positions whilst 40% had two positions both horizontally and vertically. With 40% of spatial relationships having 2 positions, compared to those with just 1, it is correct for MAC-REALM to report two positions as this gives clarity to positioning of objects.

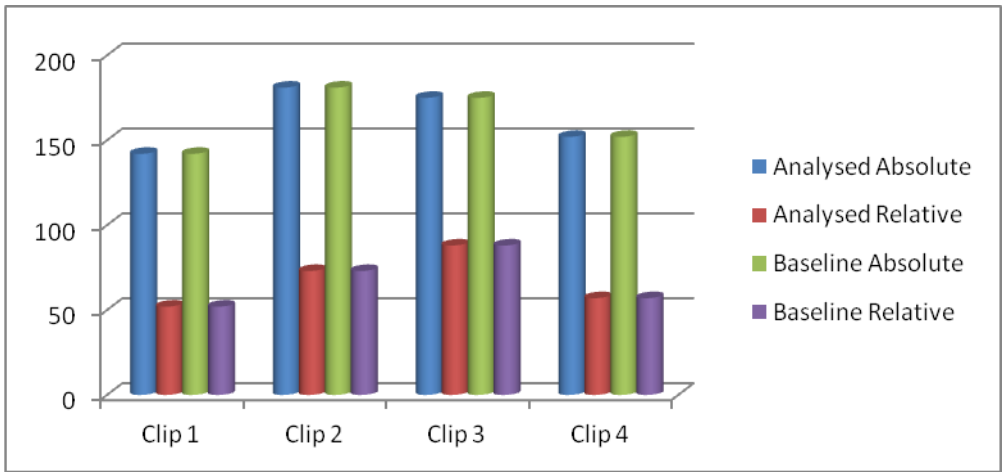


FIGURE 24: Derived spatial relationships vs. total amount of spatial relations

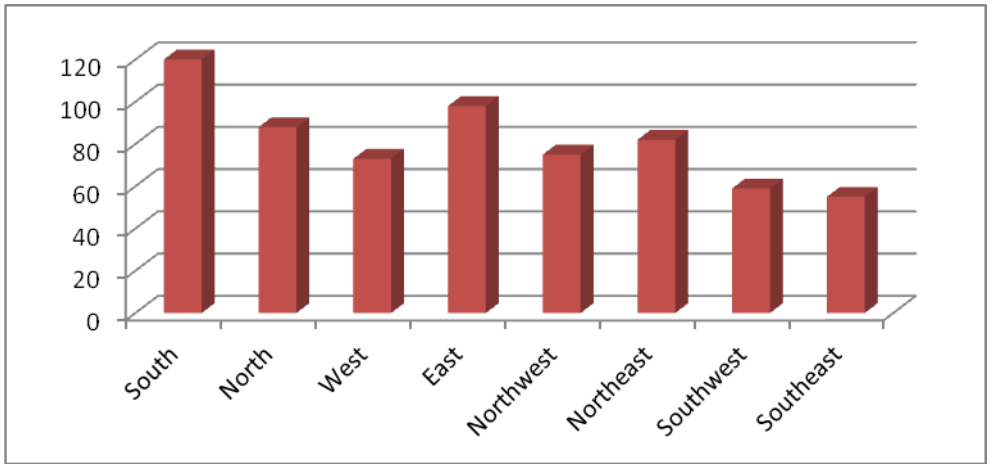


FIGURE 25: Number of absolute spatial relationships found for each position

	Above	Below	Left	Right
Above	40		25	22
Below		43	24	28
Left	25	24	40	
Right	22	28		38

FIGURE 26: Number of relative spatial relationships found for combination of positions

Bilvideo-7 [Baştan, Çam, Güdükbay and Ulusoy 2010] defines spatial relationships between objects but uses a Minimum Bounding Rectangle (MBR) to define the objects. The centre of the MBR is taken as the reference point to calculate the position of spatial relationships. This method is not as accurate as MAC-REALM which uses the centre of mass of an object as the MBR assumes the shape of the object is uniform within the MBR. This gives MAC-REALM an advantage with relative spatial relationships as the spatial relationship is defined closer to the semantic meaning of spatial relationships. DanVideo [Kannan, Andres and Guetl 2010] uses manual indexing of the dance moves which is highly accurate, but as with object detection it is time consuming and prone to human error.

4.5 Temporal relationships

Temporal relationships form the basis of semantic querying by allowing the user to investigate both semantic and syntactic features through their chronological relationship to each other and the meaning of those relationships. All features have a temporal component and can therefore have a temporal relationship with any other feature. This

intra/inter temporal relationship dependency allows for more intuitive search queries from the user as it enables them to link abstract concepts to physical elements. For example, a query can be formulated as "When does object A and object B reverse positions". This query involves all content features that have been extracted and sets the context for a user query. In Figure 27 we have types of content feature along both axes. The intersection where they meet shows the amount of temporal relationships between them. The figure shows both binary and inverse binary relationships. These are shown together to eliminate duplication. One may see that the amount of temporal relationships increases by multiple factors depending on the number of the instances of the content feature within the video stream. As there are only a few scenes the amount of temporal relationships is small. However, as there are a large amount of spatial relationships there are potentially thousands more temporal relationships. The table shows temporal relationships add descriptive meaning exponentially depending on the increased presence of a feature, thus, giving more querying advantages. The graph on Figure 28 shows that the majority of temporal relationships are "precede" or "follows".

This is usual norm when one considers the generalised case of a feature's time point $[i_x, j_x]$. If it is but one of many features $[i_N, j_N]$ then $\forall[i_{N-x}, j_{N-x}] < [i_x, j_x] < \forall[i_{N+x}, j_{N+x}]$ which therefore means that for every feature there may be an exponential increase for every other feature that it is compared too. We may also see that temporal relationships only require one time point for a feature to meet the time point of another feature with "meets", overlaps, starts, finishes, co-occurs and their inverses being the most popular. Finally, those that need both time points of both features to be satisfied i.e. contains, during, strict contains and strict during were the least used.

Feature vs. Feature	Shots	Scenes	Objects	Spatial Relationships
Shots	685584	12420	117576	761760
Scenes	12420	225	2130	13800
Objects	117576	2130	20164	130640
Spatial Relationships	761760	13800	130640	846400

FIGURE 27: Temporal relationships found for combination of features

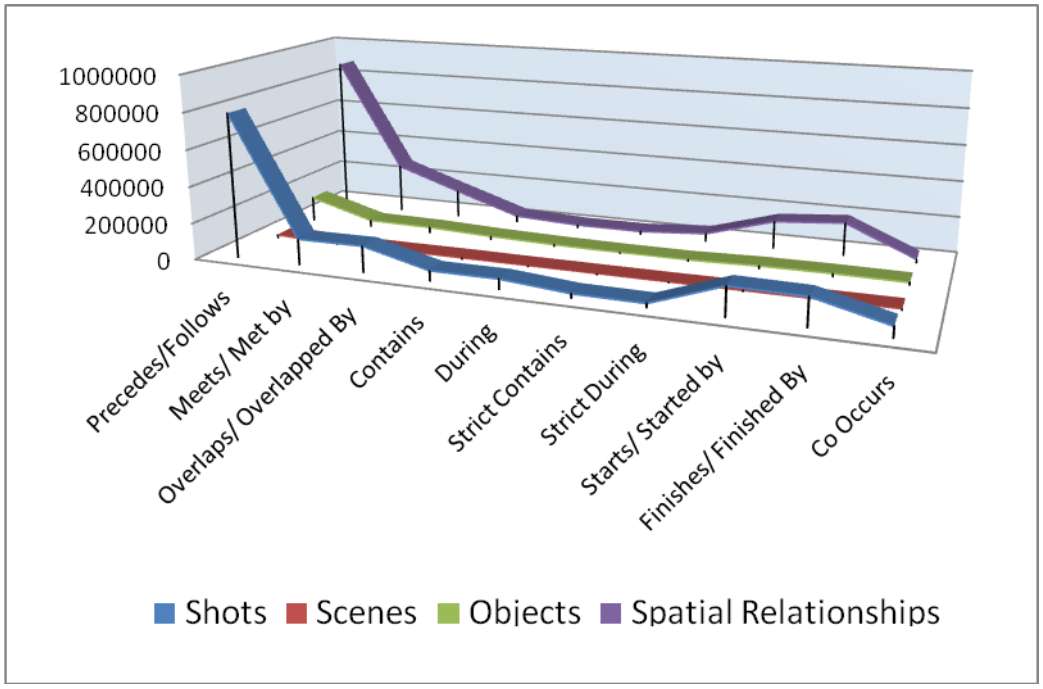


FIGURE 28: Temporal relationships found for each Feature

Bilvideo-7 [Baştan, Çam, Güdükbay and Ulusoy 2010] extracts and models temporal relationships but is unclear if it models the full set of MPEG-7 temporal relationships. DanVideo [Kannan, Andres and Guetl 2010] manually annotates temporal relationships, which is yet again time consuming, but its implementation does not report how complex this modelling approach is, and if the full set of temporal relationships is used, as set out in the MPEG-7 temporal relationship classification [Manjunath, Salembier and Sikora 2002].

4.6 Content Modelling

The MAC-REALM content model was validated against different versions, profiles and constraints of MPEG-7 standard versions 1 of 1999 and 2 of 2004, with three different profiles, namely DAVP, TRECVID and AVDP, along with temporal validation. The MAC-REALM content model successfully completed all possible permutations. The MPEG-7 valid MAC-REALM content model means that the model is accessible to all MPEG-7 compliant content based video search applications. To fully test the content model for its multi-content type ability and granular search capabilities, the MAC-REALM content model needs to be tested against a number of MPEG-7 compliant content based video search applications none of which are available. The MAC-REALM content model validates for all known profiles of MPEG-7, and is compliant to all parts of the standard. From the compliance results it can also be extrapolated that all other MPEG-7 compliant content based video search applications will also be interoperable.

Other related works used approach this using XQuery [Baştan, Çam, Güdükbay and Ulusoy 2010; Döller et al. 2011; Kannan, Andres and Guetl 2010]. Often extensions are added to XQuery for multimedia [Xue et al. 2009] and SQL/MM, MMDOC-QL [Kang et al. 2003]. Such XQuery search tools are not guaranteed to be MPEG-7 compliant, as combining these varied query approaches with alternate metadata description formats and retrieval interfaces prevents effective interoperability between MPEG-7 multimedia retrieval systems. The non-standardised process to designing MPEG-7 search functionality means, although many content based video search systems claim to be "MPEG-7 compliant", they were in practice limited in their compatibility of the MPEG-7 standard, especially with regards to the semantic querying of multimedia content.

MPEG-7 query format (MPQF) is created to solve the problem of search interoperability and was ratified officially into the MPEG-7 standard [MPEG 2012]. MPQF provides a query syntax that makes access to distributed multimedia resources unified. The standardisation of MPQF into MPEG-7 leads to two main benefits: interoperability between parties in a distributed environments and platform independence. The key feature of MPQF is that it addresses the weaknesses of XQuery such as fuzzy request handling and formal semantics for syntax and processing multimedia objects. MPQF allows for queries specifically targeted by the MAC-REALM content model such as query-by-example media, query-by-example-description, query-by-keywords, query-by-feature-range, query-by-spatial-relationships and query-by-temporal-relationships. This allows querying to be performed in a "coarse to fine" manner, with capability for searching only relevant content features within the MAC-REALM content model. Applications that are fully MPEG-7 compliant would provide a better test platform for the MAC-REALM content model, but there are currently no applications available for testing. As MAC-REALM is validated against a broad range of specifications for the MPEG-7 standard, it can be concluded that interoperability between MAC-REALM and MPQF may facilitate multi-content type and granular searches.

5. CONCLUDING DISCUSSION

MAC-REALM conceptualises the entire content feature extraction and modelling process. It uses a mixture of existing algorithms that have been extended or adapted in order to enable extract content features into content description from raw media, then amalgamates and structures the content descriptions into a content model. The abstract design of MAC-REALM provides enough flexibility in order to customise both its architecture and functionality and therefore, its implementation. In turn, this yields several advantages. Firstly, separation of its functionality into four distinct planes with three layers, allows customisation at each layer rather than the entire framework and helps to achieve low level of coupling, for example modularity. Secondly, use of a novel pre-processing technique improves the feature extraction from the media and reduces unnecessary processing by removing redundant data. Thirdly, the syntactic feature extraction plane uses a hierarchical architecture, which extracts three syntactic features using a mixture of unsupervised and semi-supervised algorithms, reducing the need for user interaction. Where human interaction is required it is to provide input that defines the semantic characteristics of the syntactic features. Fourthly, the semantic relationships of the content features are derived from analysis, along with the subsequent linking of the syntactic features, provides semantic links between all features. Spatial relationships provide a spatial context to the content model, facilitating spatial search parameters on the content. The temporal relationships allow queries of the syntactic and semantic features in the same temporal context. The semantic relationships provide a semantic foundation to the content model that enables the addition of high level concepts and facilitates event based querying.

The choice of MPEG-7 as the content model feature description language was made to ensure that the MAC-REALM content model was acceptable to the widest range of applications and that any descriptions are backward compatible. This makes possible "coarse to grain" search by of any feature or combination of features. Through the temporal relationships, a semantic temporal search is possible on all of the features. The interlinking of the syntactic and semantic features through the syntactic features elements, physical attributes and the semantic modelling nodes, provides tight coupling between the syntactic and semantic features. The tight integration of these features on a structural level also allows the content to be queried using logic based queries. Modelling all the temporal relationships between all content features provides an abstract temporal relationship, which allows the content to be queried using event based queries. These two types of querying provide multi-content type search, through the integration of logic and semantic search capabilities. MAC-REALM is a functional prototype and proves that MAC-REALM can convert "raw media into a content model through a process of content feature extraction and modelling". The framework is a novel approach to content conversion, where there are three layers to the content extraction and four modelling planes. Each layer provides modularity, by defining the function of each component at the intersection between planes. These components can be updated to provide better extraction of existing features, or extended to extract new features that better fulfil the desired functionality. The sequential arrangement of custom modules at layers, allows for custom video processing pipelines to be created. Finally, it can be argued that MAC-REALM is a significant attempt at bridging the 'semantic gap', by combining automated syntactic and semantic content extraction into an MPEG-7 searchable content model, while providing an extensible and modular development framework, which allows for custom pipelines to be created.

REFERENCES

AGIUS, H. AND ANGELIDES, M.C. 2005. COSMOS-7: Video-Oriented MPEG-7 Scheme for Modelling and Filtering of Semantic Content. *The Computer Journal 48*, 545-562.

ALLEN, J.F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM 26*, 832-843.

ALY, R., DOHERTY, A., HIEMSTRA, D. AND SMEATON, A. 2010. Beyond Shot Retrieval: Searching for Broadcast News Items Using Language Models of Concepts. In *Advances in Information Retrieval*, C. GURRIN, Y. HE, G. KAZAI, U. KRUSCHWITZ, S. LITTLE, T. ROELLEKE, S. RÜGER AND K. VAN RIJSBERGEN Eds. Springer Berlin / Heidelberg, 241-252.

AMIRI, A. AND FATHY, M. 2011. Video shot boundary detection using generalized eigenvalue decomposition and Gaussian transition detection. *Computing and Informatics 30*, 595-619.

AMRI, A. AND FATHY, M. 2010. Video shot boundary detection using QR-decomposition and gaussian transition detection. *EURASIP Journal on Advances in Signal Processing 2009*, 1-12.

ANGELIDES, M.C. 2003. Guest Editor's Introduction: Multimedia Content Modeling and Personalization, 12-15.

ANGELIDES, M.C. AND KEVIN LO, T.S. 2005. A video content independent mining algorithm for evolved rule-based detection of scene boundaries. *Ingénierie des systèmes d'information 10*, 81-99.

ANGELIDES, M.C. AND LO, K.T.S. 2005. *A video content independent mining algorithm for evolved rule-based detection of scene boundaries*. Lavoisier, Paris, FRANCE.

APACHE 2013. Hadoop.

APPIAH, K., HUNTER, A., DICKINSON, P. AND MENG, H. 2010. Accelerated hardware video object segmentation: From foreground detection to connected components labelling. *Computer Vision and Image Understanding 114*, 1282-1291.

AYADI, T., ELLOUZE, M., HAMDANI, T. AND ALIMI, A. 2012. Movie scenes detection with MIGSOM based on shots semi-supervised clustering. *Neural Computing and Applications*, 1-10.

AYVACI, A. AND SOATTO, S. 2012. Detachable Object Detection: Segmentation and Depth Ordering from Short-Baseline Video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*, 1942-1951.

BABENKO, B., MING-HSUAN, Y. AND BELONGIE, S. 2011. Robust Object Tracking with Online Multiple Instance Learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 1619-1632.

BABER, J., AFZULPURKAR, N. AND BAKHTYAR, M. 2011. Video segmentation into scenes using entropy and SURF. In *Emerging Technologies (ICET), 2011 7th International Conference on*, 1-6.

BAI, X., WANG, J. AND SAPIRO, G. 2010. Dynamic Color Flow: A Motion-Adaptive Color Model for Object Segmentation in Video. In *Computer Vision – ECCV 2010*, K. DANIILIDIS, P. MARAGOS AND N. PARAGIOS Eds. Springer Berlin / Heidelberg, 617-630.

BALLAN, L., BERTINI, M., BIMBO, A., SEIDENARI, L. AND SERRA, G. 2011. Event detection and recognition for semantic annotation of video. *Multimedia Tools and Applications 51*, 279-302.

BARTOLINI, I., PATELLA, M. AND ROMANI, C. 2011. SHIATSU: tagging and retrieving videos without worries. *Multimedia Tools and Applications*, 1-29.

BAŞTAN, M., ÇAM, H., GÜDÜKBAY, U. AND ULUSOY, O. 2010. Bilvideo-7: an MPEG-7- compatible video indexing and retrieval system. *MultiMedia, IEEE 17*, 62-73.

CANNY, J. 1986. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 679-698.

CHAN, C. AND WONG, A. 2011. Shot Boundary Detection Using Genetic Algorithm Optimization. In *Multimedia (ISM), 2011 IEEE International Symposium on*, 327-332.
CHAO, L., CHANGSHENG, X., JIAN, C. AND HANQING, L. 2011. TVParser: An automatic TV video parsing method. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 3377-3384.
CHEN, Y., DENG, Y., GUO, Y., WANG, W., ZOU, Y. AND WANG, K. 2010. A Temporal Video Segmentation and Summary Generation Method Based on Shots' Abrupt and Gradual Transition Boundary Detecting. In *Communication Software and Networks, 2010. ICCSN '10. Second International Conference on*, 271-275.
CHIARCOS, C., NORDHOFF, S. AND HELLMANN, S. 2012. *Linked Data in Linguistics: Representing and Connecting Language Data and Language Metadata*. Springer.
CHOROŚ, K. AND PAWLACZYK, P. 2010. Content-Based Scene Detection and Analysis Method for Automatic Classification of TV Sports News. In *Rough Sets and Current Trends in Computing*, M. SZCZUKA, M. KRYSZKIEWICZ, S. RAMANNA, R. JENSEN AND Q. HU Eds. Springer Berlin / Heidelberg, 120-129.
CHRISTODOULOU, L., KASPARIS, T. AND MARQUES, O. 2011. Advanced statistical and adaptive threshold techniques for moving object detection and segmentation. In *Digital Signal Processing (DSP), 2011 17th International Conference on*, 1-6.
DAL MUTTO, C., DOMINIO, F., ZANUTTIGH, P. AND MATTOCCIA, S. 2012. Stereo Vision and Scene Segmentation.
DEL FABRO, M. AND BOSZORMENYI, L. 2010. Video Scene Detection Based on Recurring Motion Patterns. In *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*, 113-118.
DÖLLER, M., STEGMAIER, F., STOCKINGER, A. AND KOSCH, H. 2011. XQuery Framework for Interoperable Multimedia Retrieval. In *Grundlagen von Datenbanken*, 73-78.
DROPBOX 2013. Dropbox.
DUMONT, É. AND QUÉNOT, G. 2012. Automatic Story Segmentation for TV News Video Using Multiple Modalities. *International Journal of Digital Multimedia Broadcasting 2012*, 11.
DYANA, A., SUBRAMANIAN, M.P. AND DAS, S. 2009. Combining Features for Shape and Motion Trajectory of Video Objects for Efficient Content Based Video Retrieval. In *Advances in Pattern Recognition, 2009. ICAPR '09. Seventh International Conference on*, 113-116.
ELLOUZE, M., BOUJEMAA, N. AND ALIMI, A. 2010. Scene pathfinder: unsupervised clustering techniques for movie scenes extraction. *Multimedia Tools and Applications 47*, 325-346.
ERCOLESSI, P., BREDIN, H., SÉNAC, C. AND JOLY, P. 2011. Segmenting TV Series into Scenes Using Speaker Diarization. In *WIAMIS 2011:, 12th International Workshop on Image Analysis for Multimedia Interactive Services*.
FEI, W. AND ZHU, S. 2010. Mean shift clustering-based moving object segmentation in the H.264 compressed domain. *Image Processing, IET 4*, 11-18.
FROMME, J. AND UNGER, A. 2012. *Computer Games and New Media Cultures: A Handbook of Digital Games Studies*. Springer.
GHUFFAR, S., BROSCH, N., PFEIFER, N. AND GELAUTZ, M. 2012. Motion segmentation in videos from time of flight cameras. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, 328-332.
GRANA, C. AND CUCCHIARA, R. 2007. Linear Transition Detection as a Unified Shot Detection Approach. *Circuits and Systems for Video Technology, IEEE Transactions on 17*, 483-489.
GRUNDMANN, M., KWATRA, V., MEI, H. AND ESSA, I. 2010. Efficient hierarchical graph-based video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2141-2148.
GÜSGEN, H.W. 1989. *Spatial reasoning based on Allen's temporal logic*. International Computer Science Institute.
HALLER, M., KRUTZ, A. AND SIKORA, T. 2009. Evaluation of pixel- and motion vector-based global motion estimation for camera motion characterization. In *Image Analysis for Multimedia Interactive Services, 2009. WIAMIS '09. 10th Workshop on*, 49-52.
HAMEED, A. 2009. A novel framework of shot boundary detection for uncompressed videos. In *Emerging Technologies, 2009. ICET 2009. International Conference on*, 274-279.
HARIKRISHNA, N., SATHEESH, S., SRIRAM, S.D. AND EASWARAKUMAR, K.S. 2011. Temporal classification of events in cricket videos. In *Communications (NCC), 2011 National Conference on*, 1-5.
HEEJUN, H. AND JAESOO, K. 2011. An useful method for scene categorization from new video using visual features. In *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, 480-484.
HU, W.C. AND HSU, J.F. 2011. Foreground extraction-based video object segmentation using motion information and gradient compensation. *International Journal of Innovative Computing, Information and Control 7*, 4849-4859.
HUANG, S.N. AND ZHANG, Z.Y. 2010. Scene detection in videos using mutual information. *Applied Mechanics and Materials 34*, 920-926.
HUANG, Y.-F. AND TUNG, L.-H. 2010. Semantic scene detection system for baseball videos based on the MPEG-7 specification. In *Proceedings of the Proceedings of the 2010 ACM Symposium on Applied Computing*, Sierre, Switzerland2010 ACM, 1774285, 941-947.
HUI, C. AND CUIHUA, L. 2010. A practical method for video scene segmentation. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 153-156.
HUNTER, J. AND IANNELLA, R. 2009. The application of metadata standards to video indexing. *Research and Advanced Technology for Digital Libraries*, 514-514.

INIGO, S.A. AND SURESH, P. 2012. General Study on Moving Object Segmentation Methods for Video. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 1*, pp: 265-270.
JACOBS, A., MIENE, A., IOANNIDIS, G. AND HERZOG, O. 2004. Automatic shot boundary detection combining color, edge, and motion features of adjacent frames, 197-206.
JACOBS, A., MIENE, A., IOANNIDIS, G. AND HERZOG, O. 2004. Automatic shot boundary detection combining color, edge, and motion features of adjacent frames. In *TRECVID 2004 Workshop Notebook Papers*, 197-206.
KANG, J.-H., KIM, C.-S. AND KO, E.-J. 2003. An XQuery engine for digital library systems. In *Proceedings of the Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, Houston, Texas2003 IEEE Computer Society, 827231, 400-400.
KANNAN, R., ANDRES, F. AND GUETL, C. 2010. DanVideo: an MPEG-7 authoring and retrieval system for dance videos. *Multimedia Tools and Applications 46*, 545-572.
KHATOONABADI, S.H. AND BAJIC, I.V. 2013. Video Object Tracking in the Compressed Domain Using Spatio-Temporal Markov Random Fields. *Image Processing, IEEE Transactions on 22*, 300-313.
KRISTENSEN, F., NILSSON, P. AND ÖWALL, V. 2006. Background segmentation beyond RGB. *Computer Vision–ACCV 2006*, 602-612.
KRULIKOVSKA, L., PAVLOVIC, J., POLEC, J. AND CERNEKOVA, Z. 2010. Abrupt cut detection based on mutual information and motion prediction. In *ELMAR, 2010 PROCEEDINGS*, 89-92.
KÜÇÜKTUNÇ, O., GÜDÜKBAY, U. AND ULUSOY, Ö. 2010. Fuzzy color histogram-based video segmentation. *Computer Vision and Image Understanding 114*, 125-134.
LADICKÝ, Ľ., STURGESS, P., ALAHARI, K., RUSSELL, C. AND TORR, P. 2010. What, Where and How Many? Combining Object Detectors and CRFs. In *Proceedings of the Computer Vision – ECCV 2010*2010, K. DANIILIDIS, P. MARAGOS AND N. PARAGIOS Eds. Springer Berlin / Heidelberg, 424-437.
LAVEE, G., RIVLIN, E. AND RUDZSKY, M. 2009. Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39*, 489-504.
LAWRENCE, E., NEWTON, S., CORBITT, B., LAWRENCE, J., DANN, S. AND THANASANKIT, T. 2012. *Internet commerce: digital models for business*. John Wiley & Sons.
LEZAMA, J., ALAHARI, K., SIVIC, J. AND LAPTEV, I. 2011. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 3369-3376.
LI, H. AND NGAN, K.N. 2011. Image/Video Segmentation: Current Status, Trends, and Challenges Video Segmentation and Its Applications, K.N. NGAN AND H. LI Eds. Springer New York, 1-23.
LI, J., DING, Y., SHI, Y. AND LI, W. 2010. A divide-and-rule scheme for shot boundary detection based on sift. *International Journal of Digital Content Technology and its Applications 4*, 202-214.
LI, S.B., WANG, L.F. AND WANG, J.L. 2010. Video Segmentation Method Based on Film Script and Subtitle Information. *Computer Engineering 15*, 077.
LIENHART, R. 2001. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics 1*, 469-486.
LIN, G., ZHU, H., FAN, C. AND ZHANG, E. 2011. Object segmentation based on guided layering from video image. *Optical Engineering 50*, 097006-097006.
LIU, L., CHEN, R., WOLF, L. AND COHEN-OR, D. 2010. Optimizing Photo Composition. In *Computer Graphics Forum, 2010* Blackwell Publishing, 469-478.
MA, C., YU, J. AND HUANG, B. 2012. A Rapid and Robust Method for Shot Boundary Detection and Classification in Uncompressed MPEG Video Sequences. *Computer Science Issues, International Journal of (IJCSI) 5*, 368-374.
MA, Y. AND CHEN, Q. 2010. Stereo-Based Object Segmentation Combining Spatio-Temporal Information. In *Advances in Visual Computing*, G. BEBIS, R. BOYLE, B. PARVIN, D. KORACIN, R. CHUNG, R. HAMMOUND, M. HUSSAIN, T. KAR-HAN, R. CRAWFIS, D. THALMANN, D. KAO AND L. AVILA Eds. Springer Berlin / Heidelberg, 229-238.
MAHESH, K. AND KUPPUSAMY, K. 2012. Video Segmentation using Hybrid Segmentation Method. *European Journal of Scientific Research, ISSN*, 312-326.
MANJUNATH, B., SALEMBIER, P. AND SIKORA, T. 2002. *Introduction to MPEG-7: multimedia content description interface*. John Wiley & Sons Inc.
MEZARIS, V., PAPADOPOULOS, G.T., BRIASSOULI, A., KOMPATSIARIS, I. AND STRINTZIS, M.G. 2009. Semantic Video Analysis and Understanding. *chapter in "Encyclopedia of Information Science and Technology", Second Edition, Mehdi Khosrow-Pour*.
MEZARIS, V., SIDIROPOULOS, P., DIMOU, A. AND KOMPATSIARIS, I. 2010. On the use of visual soft semantics for video temporal decomposition to scenes. In *Proc. Forth IEEE Int. Conf. on Semantic Computing (ICSC 2010)*, 141-148.
MICROSOFT 2013. SkyDrive.
MIKA, P. AND GREAVES, M. 2012. Editorial: Semantic Web & Web 2.0. *Web Semantics: Science, Services and Agents on the World Wide Web 6*.
MITROVIĆ, D., HARTLIEB, S., ZEPPELZAUER, M. AND ZAHARIEVA, M. 2010. Scene Segmentation in Artistic Archive Documentaries. In *HCI in Work and Learning, Life and Leisure*, G. LEITNER, M. HITZ AND A. HOLZINGER Eds. Springer Berlin / Heidelberg, 400-410.
MOENS, M.F., POULISSE, G.J. AND VRT, M.M. 2012. State of the art on semantic retrieval of AV content beyond text resources.

MOHANTA, P.P., SAHA, S.K. AND CHANDA, B. 2010. A heuristic algorithm for video scene detection using shot cluster sequence analysis. In *Proceedings of the Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, Chennai, India2010 ACM, 1924621, 464-471.
MOHANTA, P.P., SAHA, S.K. AND CHANDA, B. 2012. A Model-Based Shot Boundary Detection Technique Using Frame Transition Parameters. *Multimedia, IEEE Transactions on 14*, 223-233.
MPEG 2012. Information technology -- Multimedia content description interface -- Part 12: Query format. In *ISO/IEC 15938* International Standards Organisation.
NOMA, A., GRACIANO, A.B.V., CESAR JR, R.M., CONSULARO, L.A. AND BLOCH, I. 2012. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition 45*, 1159-1179.
OCHS, P. AND BROX, T. 2011. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 1583-1590.
PORIKLI, F., BASHIR, F. AND HUIFANG, S. 2010. Compressed Domain Video Object Segmentation. *Circuits and Systems for Video Technology, IEEE Transactions on 20*, 2-14.
POULISSE, G.-J., PATSIS, Y. AND MOENS, M.-F. 2012. Unsupervised scene detection and commentator building using multi-modal chains. *Multimedia Tools and Applications*, 1-17.
REN, W., SINGH, S., SINGH, M. AND ZHU, Y.S. 2009. State-of-the-art on spatio-temporal information-based video retrieval. *Pattern Recognition 42*, 267-282.
RICHARDSON, I. 2010. *The H. 264 advanced video compression standard*. Wiley.
ROSMAN, B. AND RAMAMOORTHY, S. 2011. Learning spatial relationships between objects. *The International Journal of Robotics Research 30*, 1328-1342.
RYOO, M.S. AND AGGARWAL, J.K. 2009. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer Vision, 2009 IEEE 12th International Conference on*, 1593-1600.
RYOO, M.S., LEE, J.T. AND AGGARWAL, J.K. 2010. Video scene analysis of interactions between humans and vehicles using event context. In *Proceedings of the Proceedings of the ACM International Conference on Image and Video Retrieval*, Xi'an, China2010 ACM, 1816109, 462-469.
SAKARYA, U. AND TELATAR, Z. 2010. Video scene detection using graph-based representations. *Signal Processing: Image Communication 25*, 774-783.
SAKARYA, U., TELATAR, Z. AND ALATAN, A.A. 2012. Dominant sets based movie scene detection. *Signal Processing 92*, 107-119.
SANG, J. AND XU, C. 2010. Character-based movie summarization. In *Proceedings of the Proceedings of the international conference on Multimedia*, Firenze, Italy2010 ACM, 1874096, 855-858.
SARMIENTO, A.S. AND LOPEZ, E.M. 2012. *Multimedia Services and Streaming for Mobile Devices: Challenges and Innovation*. Information Science Reference.
SEELING, P. 2010. Scene Change Detection for Uncompressed Video. In *Technological Developments in Education and Automation*, M. ISKANDER, V. KAPILA AND M.A. KARIM Eds. Springer Netherlands, 11-14.
SEUNG-BO, P., HEUNG-NAM, K., HYUNSIK, K. AND GEUN-SIK, J. 2010. Exploiting Script-Subtitles Alignment to Scene Boundary Dectection in Movie. In *Multimedia (ISM), 2010 IEEE International Symposium on*, 49-56.
SHAO, L., JI, L., LIU, Y. AND ZHANG, J. 2012. Human action segmentation and recognition via motion and shape analysis. *Pattern Recognition Letters 33*, 438-445.
SHARIR, G. AND TUYTELAARS, T. 2012. Video object proposals. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 9-14.
SIDIROPOULOS, P., MEZARIS, V., KOMPATSIARIS, I., MEINEDO, H., BUGALHO, M. AND TRANCOSO, I. 2011. Temporal Video Segmentation to Scenes Using High-Level Audiovisual Features. *Circuits and Systems for Video Technology, IEEE Transactions on 21*, 1163-1177.
SINGHAI, N. AND SHANDILYA, S.K. 2010. A Survey On:"Content Based Image Retrieval Systems". *International Journal of Computer Applications IJCA 4*, 22-26.
SMEATON, A.F., OVER, P. AND DOHERTY, A.R. 2010. Video shot boundary detection: Seven years of TRECVid activity. *Computer Vision and Image Understanding 114*, 411-418.
SNOEK, C.G.M. AND WORRING, M. 2009. Concept-Based Video Retrieval. *Found. Trends Inf. Retr. 2*, 215-322.
SOFOKLEOUS, A.A. AND ANGELIDES, M.C. 2008. DCAF: an MPEG-21 dynamic content adaptation framework. *Multimedia Tools and Applications 40*, 151-182.
SU, X., BAILAN, F., PENG, D. AND BO, X. 2012. Graph-based multi-modal scene detection for movie and teleplay. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 1413-1416.
SUBUDHI, B.N., NANDA, P.K. AND GHOSH, A. 2011. Moving objects detection from video sequences using fuzzy edge incorporated Markov random field modeling and local histogram matching. In *Proceedings of the Proceedings of the 4th international conference on Pattern recognition and machine intelligence*, Moscow, Russia2011 Springer-Verlag, 2026885, 173-179.
SUGARSYNC 2013. SugarSync.
TAPU, R. AND ZAHARIA, T. 2011. High Level Video Temporal Segmentation. *Advances in Visual Computing 6938*, 224-235.
TAPU, R. AND ZAHARIA, T. 2011. Video Segmentation and Structuring for Indexing Applications. *International Journal of Multimedia Data Engineering and Management (IJMDEM) 2*, 38-58.

TIAN, Z., XUE, J., LAN, X., LI, C. AND ZHENG, N. 2011. Key object-based static video summarization. In *Proceedings of the Proceedings of the 19th ACM international conference on Multimedia*, Scottsdale, Arizona, USA2011 ACM, 2071999, 1301-1304.
TJONDRONEGORO, D.W. AND CHEN, Y.P.P. 2010. Knowledge-Discounted Event Detection in Sports Video. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 40*, 1009-1024.
TORRES, R.S., FALCÃO, A.X., GONÇALVES, M.A., PAPA, J.P., ZHANG, B., FAN, W. AND FOX, E.A. 2009. A genetic programming framework for content-based image retrieval. *Pattern Recognition 42*, 283-292.
TRYON, C. 2012. 'Make any room your TV room': digital delivery and media mobility. *Screen 53*, 287-300.
TSAO, H.H. 2011. DCT Based Fast Object Detection and Segmentation Design for Compressed Video and Implementation on Embedded System. In *Electronic and Information Engineering* National Yunlin University of Science and Technology, Douliu City, Yunlin County, Taiwan.
TSINGALIS, I., VRETOS, N., NIKOLAIDIS, N. AND PITAS, I. 2012. Anthropocentric descriptors and description schemes for multi-view video content. In *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean*, 133-136.
TUZEL, O., PORIKLI, F. AND MEER, P. 2006. Region Covariance: A Fast Descriptor for Detection and Classification. In *Computer Vision – ECCV 2006*, A. LEONARDIS, H. BISCHOF AND A. PINZ Eds. Springer Berlin Heidelberg, 589-600.
VAN DEN BERGH, M. AND VAN GOOL, L. 2012. Real-time stereo and flow-based video segmentation with superpixels. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, 89-96.
VAZQUEZ-REINA, A., AVIDAN, S., PFISTER, H. AND MILLER, E. 2010. Multiple Hypothesis Video Segmentation from Superpixel Flows. *Computer Vision – ECCV 2010 6315*, 268-281.
VIJAYAKUMAR, V. AND NEDUNCHEZHIAN, R. 2012. A study on video data mining. *International Journal of Multimedia Information Retrieval 1*, 153-172.
VISSER, A. 2011. On the ambiguation of Polish notation. *Theoretical Computer Science*.
VROCHIDIS, S., MOUMTZIDOU, A., KING, P., DIMOU, A., MEZARIS, V. AND KOMPATSIARIS, I. 2010. VERGE: A video interactive retrieval engine. In *Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on*, 1-6.
W.S. ANDERSON, P. 2004. AVP: Alien vs. Predator Twentieth Century Fox Film Corporation.
WANG, H.H., MOHAMAD, D. AND ISMAIL, N. 2010. Semantic Gap in CBIR: Automatic Objects Spatial Relationships Semantic Extraction and Representation. *International Journal Of Image Processing (IJIP) 4*, 192.
WEIMING, H., NIANHUA, X., LI, L., XIANGLIN, Z. AND MAYBANK, S. 2011. A Survey on Visual Content-Based Video Indexing and Retrieval. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41*, 797-819.
WILSON, K.W., DIVAKARAN, A., NIU, F., GOELA, N. AND OTSUKA, I. 2010. Method for detecting scene boundaries in genre independent videos Google Patents.
WOLF, M. AND WICKSTEED, C. 1998. Date and time formats. *W3C NOTE NOTE-datetime-19980827, August*.
XU, W. AND XU, L. 2010. A novel shot detection algorithm based on graph theory. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, V3-628-V623-630.
XUE, L., LI, C., WU, Y. AND XIONG, Z. 2009. VeXQuery: an XQuery extension for MPEG-7 vector-based feature query. In *Advanced Internet Based Systems and Applications* Springer, 34-43.
YONGQUAN, X., WEILI, L. AND SHAOHUI, N. 2009. A Simple and Fast Segmentation Approach for Sport Scene Images. In *Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on*, 466-470.
ZAJIĆ, G.J., RELJIN, I.S. AND RELJIN, B.D. 2011. Video Shot Boundary Detection based on Multifractal Analisys. *Telfor Journal 3*, 105-110.
ZENG, X., ZHANG, X., HU, W. AND LI, W. 2010. Video Scene Segmentation Using Time Constraint Dominant-Set Clustering. In *Advances in Multimedia Modeling*, S. BOLL, Q. TIAN, L. ZHANG, Z. ZHANG AND Y.-P. CHEN Eds. Springer Berlin / Heidelberg, 637-643.
ZHU, Q., XIE, Y., GU, J. AND WANG, L. 2012. A New Video Object Segmentation Algorithm by Fusion of Spatio-temporal Information Based on GMM Learning. In *Advances in Automation and Robotics, Vol. 2*, G. LEE Ed. Springer Berlin Heidelberg, 641-650.
ZHU, S. AND GUO, Z. 2012. An Overview of Video Object Segmentation. In *Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on*, 1019-1021.
ZHU, S. AND LIANG, Z. 2011. Semantic scene segmentation for advanced story retrieval. *Information Technology Journal 10*, 98-105.