Politehnica University of Bucharest

THESIS

Multimodal Advertising

Scientific Adviser
Prof. Paul-Alexandru Chiriţă

Author
Dan Bănică

- 2012 -

Contents

List of Abbreviations

UPB	University Politehnica of Bucharest
CS	Computer Science
VAST	Video Ad Serving Template
VPAID	Video Player Ad Interface Definition

List of Figures

Chapter 1

Introduction

Perhaps the biggest advantage that Online Advertising has over regular form of advertising is the fact that the publishing of Online Advertising is not bounded by geographic limits. Having the message reaching a global audience automatically turns into a bigger profit. Online advertising is a lot faster than regular forms of advertising as online ads can be sent to the audience as soon as the advertising campaign begins. Another advantage is the possibility to show to the customers only those ads that are relevant for them. For instance, the advertising system that Facebook uses only shows ads to those users that have interests related to the product that is presented. The traditional forms of advertising are reaching a more general category of receivers, thus it may not reach the people it was meant to reach in the first place. Yet another thing that makes Online Advertising better than regular forms of advertising is the fact that there are a lot of analytic tools that are used for the tracking and measuring of the effects of Online Advertising than similar tools for offline advertising. For example, it is impossible to find out exactly how many people have seen an ad which was made with offline advertising, however this is a trivial task in Online Advertising. The Online Advertising is a lot cheaper than regular forms of advertising. For example, an ad in a newspaper may cost several hundred dollars, while the payments for the Online Advertising are significantly smaller. Also, the payments for Online Advertising can be made on a performance basis (impressions, clicks or leads), which is not possible to do in offline advertising.

Given all the observations above, it is no wonder that 2012 would be the year that spending for online advertising would surpass the spending for print advertising. In 2011 for example, the budgets for online ads grew 23% in the United States and similar figures are forecasted for 2012 [35].

One of the most important trends in online advertising is represented by video advertising. The growing presence of online video is providing an incentive to advertisers and marketers to increase their budget in this direction. According to comScore, a leading authority in online video statistics, in February 2012, 179 million U.S. Internet users watched nearly 38 billion videos online. The average viewer watched 21.8 hours of online video content. Google sites, being primarily represented by YouTube.com ranked as the top online video content provider in February 2012 and also in the respective month video ads reached 50 percent of the total U.S. population. [36]

This steady increase demands for better methods and a better understanding of how to efficiently deliver the advertisement in these environments. Many of the current video-advertising systems rely on linear ads which capture the entire screen, interrupting the actual video. A different category of ads is represented by the overlays, which are usually inserted at predefined

Figure 1.1: Online vs Print Advertising

locations, in time and space. Although the overlays alter the user experience significantly less than linear ads, there are for sure situations where a predefined position results in an obtrusive placing, for example by covering some important elements of the scene, or through unfortunate color combinations. Such systems often miss better alternative for placing the ad, just by using a predefined, "one-fits-all" solution.

A particular challenge that video advertising poses is to choose the right moment to insert the advertisement. As opposed to other content containers, video explicitly possesses a temporal component which needs to be taken into account. From this perspective, placing an advertisement into a highly captivating scene may result in different effects regarding user perception of the brand than placing it into a less attracting one. Similarly, choosing the advertisement placement is important, when not relying on linear ads which actually pause the video and capture the entire screen, since the ad and the actual video content compete for the same, limited space represented by the video player.

The purpose of the current thesis is to tackle these new challenges that video advertising has and to implement a system where ads are both relevant and visually unobtrusive.

Chapter 2

Problem description

As mentioned before, this thesis presents a system for inserting advertisements into video. Although some of the strategies described here may apply to TV, for the sake of concreteness we will limit the description to the context of online advertising.

Generally, online advertising is done through ad networks. An ad network collects a large number of web-sites (publishers) and advertisers, and then each time a user visits one of the sites he receives some ads. Ideally, each of the mentioned participants benefits from this process: *the advertiser* increases his brand awareness (or directly his revenue); *the publisher* (web-site owner) receives money from the advertiser for showing the ad on his site, thus allowing him to offer *the users* free or better services. However, poor targeted or too intrusive ads may annoy the users, who will ignore the ads and even choose competitor web-sites that offer the services they need. Furthermore, this kind of ads can have a negative impact in the way the advertiser is perceived.

The two main problems that shall be tackled here are the ones mentioned above - ad matching (finding the most suitable ad) and unobtrusive ad insertion.

Besides these, there are other important aspects that need to be solved in online advertising, e.g. deciding what sum an advertiser will pay for a click on his advertisement. This is usually solved through a bidding process. Although solving such problems for video may represent fruitful research areas capable of generating approaches (e.g.: bidding on objects in the video) inexistent in traditional forms of advertising (like banner advertising or in-text advertising), such problems are explicitly outside the scope of my project.

2.1 Starting hypotheses

There are a few hypotheses that I will use while designing the system. Although they may seem obvious, and although there may exist some other hypothesis that will be implicitly made during solving the problem, I consider that stating the most important of them here will at least shed some light on the nature of the problem.

- While selecting the ad to be delivered, it is better if that ad is contextually related to the movie.

There are more justifications supporting this affirmation: most obvious, as the user is watching the movie, he does this for a reason - we may assume that he is interesting in the subject of the movie. Therefore trying to find an ad related to the movie is a good heuristic for trying to find

an ad that the user is interested in. Besides this, there are also more subtle reasons: it is more likely that the user will have a negative attitude towards something that is totally unrelated to the context. Psychological concepts like "priming" may be helpful here: it is known that a more positive attitude exists towards something that has been seen before. Also, targeted advertising has been shown to lead to improved conversion [37].

Furthermore, as an extension of this hypothesis, it may be stated that it is better for the ad to appear as close as possible to the moment of the scene that it is contextually related to.

- While placing an ad into a video, it is better to put it in less informative areas of the screen.

Although on very short term placing ads in important areas (like in the middle of the screen, on the face of main character) might lead to a high number of clicks, this is a terrible long-term strategy (both for the publisher and also for the advertiser). We shall try placing ads in less salient areas of the movie. Previous studies [38] have reported that intrusive advertising leads to negative attitudes and diminished intention to return to the site.

- While placing an ad into a video, it is better to put it during interesting scenes.

This hypothesis refers to the temporal placement of the ad (as opposed to the previous one that refers to the spatial placement). This idea is also suggested by [11], and the rationale behind this hypothesis is that during important scenes the users are more connected to the movie. However, it is essential that the spatial rule is respected; otherwise the entire effect of the scene may be destroyed because of the ad (which will annoy users).

Note that through the hypotheses formulated by now, the two problems that I am trying to solve seem independent. However, the following hypothesis leads to a connection between the two problems:

- While selecting and placing the ad to be delivered, it is better to have an ad that is visually similar to the video scene that surrounds it.

This means that delivering an ad that seamlessly integrates into the movie leads to better effect than delivering an ad that is totally out of context, distracting the user from viewing the movie. This idea is also supported by the intrusiveness discussion above, where we cited [38] as a work describing negative effects of intrusive advertising. Other case studies [39] have reported an immediate increase in revenue after changing the ads such that they blend better.

Chapter 3

Related work

There exists some prior research that can be grouped in some different more or less interrelated fields, which may be used in order to deliver an intelligent video advertising platform. I will analyze separately the following categories by presenting some articles from each:

a. Attention modeling

b. Intelligent advertising systems

c. Intelligently finding ads

d. Video comprehension (without attention modeling)

e. Understanding how users process information

However, besides the research fields that may lead to interesting algorithms and although my scope is not to deliver a fully standard compliant ad server, it is also important to be aware of the standards existing in the industry. There are a few documents published by IAB (Interactive Advertising Bureau) that regulate the online video advertising process. In the current context, these documents are important mainly because they give an overview about how video advertising works and also because they establish a terminology.

- IAB documents about video advertising

1 *Digital Video Ad Format Guidelines and Best Practices [27]*

The most common in-stream video ad formats are:

- Linear
- Non linear

Linear ads are those which capture the entire screen, stopping the main movie while they are playing. These are TV-like ads.

The main advantage of Non Linear ads is that the main content may also be played while showing the ad.

There are three types of linear ads:

- Pre-roll: they play before the movie. There may or may not exist a button to skip the ad.

- Mid-roll: they appear sometime during the main movie: the main movie is paused, the ad occupies the entire video player, and after it is played, the main movie is continued
- Post-roll: they play at the end of the movie.

There are two kinds of non-linear ads:

- Overlays: they appear somewhere over the video content.
- Non-overlays: they appear inside the player, but outside of the video content.

Overlays represent the main focus of this project.

2 *Digital Video Ad Format Guidelines and Best Practices [27]*

In this document a series of metrics that could be used to measure how well an ad performed are enumerated: impressions (how many times the ad was viewed), click-through (how many times the ad was clicked), completed (how many times the entire ad was viewed - without skipping any part of it), time spent (how much time was spent for viewing the ad - counting sections that were repeated), expanded (whether the ad was expanded), collapsed (whether the ad was collapsed).

3 *Video Ad Serving Template (VAST) [29]*

The document defines an XML protocol that should be used when serving an advertisement.

4 *Video Player Ad Interface Definition (VPAID) [30]*

As opposed to traditional web advertising, in video advertising the ad plays inside the video player. Therefore, for more interactive ads a protocol between the player and the ad is required. The VPAID document describes such a protocol by defining some methods that the ad must implement. The player calls these methods when certain events occur. Examples of methods are: *resizeAd, startAd, stopAd, pauseAd, resumeAd, collapseAd, expandAd.* This way, the ad may implement some special behavior when the size on which it appears on the screen changes (for example it may dynamically choose its layout depending on the available area, by implementing this functionality in the resizeAd method).

The figure below illustrates the last two protocols (step 2 is represented by VAST protocol and step 3 by VPAID):

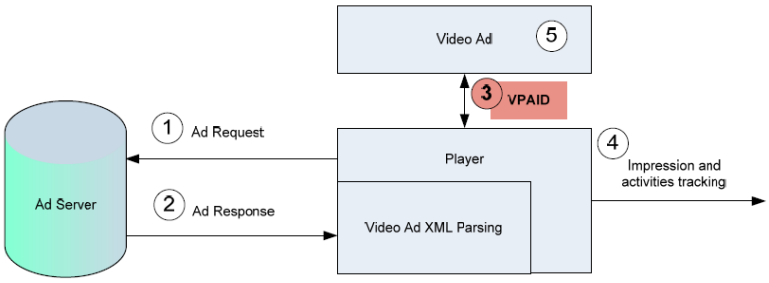

Figure 3.1: How video advertising works.

However, as stated before, my purpose is not to deliver a standard compliant video ad server, but rather to find better solutions for some of the particular aspects involved in video

advertising that are more related to the field of Artificial Intelligence. The above discussion about IAB documents was done just in order to present an overview of how video advertising works.

In the rest of this document I will focus on the two tasks that I have stated in the beginning: given a video, find suitable ads and suitable insertion points (in space and time).

I shall now proceed by describing some papers from different research fields that may be useful in developing intelligent algorithms for solving these tasks.

3.1 Attention modeling and applications

Modeling user attention is essential for ad placement because knowing where the user looks represents an essential clue.

A Model of Saliency-Based Visual attention for rapid scene analysis [1]

One of the most influential work in attention modeling is that of L. Itti and C. Koch who are basically the initiators of this domain. This is why I will begin with this article that describes an overview of their system. They start from the premise that primates, in order to efficiently process visual scenes focus their attention only on important zones.

Multiple feature maps are generated for different type of features in the following way: given a feature (e.g.: intensity), a region is salient if the feature value in that region is very different compared to the surroundings. For each kind of feature, multiple maps are generated by using multiple resolutions (sizes): three different sizes are considered for center region, and for each of these, two sizes are considered for surroundings, resulting in a number of 6 maps for each feature.

The features used are intensity, color contrast (considering red-green and yellow-blue) and orientation (considering 0, 45, 90, 135 degrees local orientations). This way we obtain 6 maps for intensity, 12 for color and 24 for orientation. These maps are then combined into a single saliency map. Multiple strategies for combining the maps are described in [2].

A comparison of feature combination strategies [2]

Compares different methods of composing the saliency maps:

1. Simple summing: all maps are normalized to [0, 1] interval and then directly added. This is the baseline method and offers the poorest results.

2. Normalization: in this strategy, the maps which have a prominent maximum compared to the other local maxima in the same map become more important. In consequence, if according to one feature one point is very salient, then this feature becomes more important.

3. Iterative normalization: it is similar to the normalization described above, except that it is a more biologically plausible method, which sometimes even returns better results. The previous variant is not biologically plausible because it uses a global maximum, while the neurons in the cortex (responsible for analyzing visual signal) are known to be only locally connected. An iterative strategy is used, that attenuates the zones for which vicinities are similar. The procedure starts with a map which is normalized into [0, 1], and then it performs attenuation / intensification of some regions (in the end the map will not be in

[0, 1] anymore, therefore this method can be viewed as a procedure for offering different weights to different maps).

4. Learning weights: this weight updating strategy takes into account how well each map emphasizes the target. By training for a specific target, the maps that represent better that target will receive bigger weights (e.g.: if the target is red and the background is always green, then the color map will become more important).

The authors reported that simple summing always offers the worst results. Normalization has good performance, but Iterative Normalization is always comparable or better and besides that it is also biologically plausible. Learning weights is a good strategy only if the target is known beforehand. Any object has its own set of representative maps, therefore the generalization capabilities of this algorithm is poor.

Learning to predict where humans look [3]

This paper takes a data driven approach for computing the salience map. Eye tracking data was recorded for 15 users on 1003 images. In order to obtain continuous saliency maps from the eye fixation points, the image was convolved with a Gaussian kernel at each fixation point. A number of 10 positive and 10 negative example pixels were randomly extracted from each image for training. These points were obtained from each image by applying a threshold on the saliency map obtained from eye tracking data, considering top 20% pixels as positive examples (fixated) and bottom 70% as negative examples.

Three types of features were used: low level features (e.g.: contrast, orientation), mid-level features (a classifier was used to determine pixels on the horizon line), high-level features (Viola-Jones algorithm for face detection). Besides these, the observation that gazes tend to be concentrated at the center of the image was used by taking into account the position of the pixel. All these features were fed into a Support Vector Machine classifier.

A visual attention model for adapting images on small displays [4]

Here the authors use saliency in order to retarget images on different displays (e.g. mobile devices). They use an attention model that detects the Attention Objects (AOs). Each AO, is described by the following values: ROI (region of interest), AV (attention value), MPS (minimal perceivable size). ROI indicates the area where the object is (and it is represented as a rectangle); AV gives the importance of the object and MPS indicates how small the object can be made while still preserving its characteristics. First, they generate the saliency maps described in [1] and combine them using the iterative strategy described in [2]. When the saliency map is available, the AV at each pixel is computed by multiplying each pixels saliency with its brightness and then a Gaussian template is applied, knowing that people tend to look at the center of the image. The MPS of each salient region is heuristically computed (larger regions may be scaled more aggressively). Besides saliency, their attention model also incorporates top-down, semantic information by identifying faces and text. These are known to attract more attention (although may not be salient from a bottom-up perspective). The AV of each face takes into account the face size and position (larger faces and faces in the center in the image have higher AV). The MPS for faces is fixed at 25x30=750 pixels. Also, text is taken into account. The AV of a text region takes into account the text area and the aspect ratio. Text position is not important here. They use a rule-based system to give weights to these different sources. For example if there are faces with a very high AV, then the weight of the face component becomes large. For retargeting, authors try to optimize an "information

fidelity" formula that tells how much information is preserved - for each AO, if its size is too small (using MPS) then its AV is set to 0.

A computable Visual Attention Model for Video Skimming [5]

The salience model in [4] is extended for video analysis (an attention curve is generated) in order to do video skimming. Video skimming refers to extracting the highlights of a movie.

A novel region based image retrieval algorithm [6]

In this article saliency is used for region-based image retrieval. First, in order to calculate the salience at each pixel the following simple algorithm is used: any pixel x is compared to all pixels y in some neighborhood, taking into account color and orientation. The more different the feature in a specific pixel, the more salient it is. After this step, the salient objects are detected. In order to obtain these objects, the image is transformed in grayscale and segmented. Now using entropy theory, some segments with a high salience are selected. These segments are joined into objects.

Finally, in order to compute the score of an image in the database (this score is used to rank the images in the database, for performing the retrieval task), for each salient object in the original image, the most similar one in the current database image is detected and a similarity measure is recorded. These values are added, resulting therefore a score for each database image.

A principled approach to detecting surprising events in video [7]

This article also starts by emphasizing the important role of detecting surprising events in order for primates to survive. It defines a surprise framework to incorporate elements from two complementary domains: saliency and novelty. Saliency represents outliers in the spatial domain, while novelty works with the temporal domain. In the novelty framework, one traditional approach is to assume that each pixel's intensity comes from a mixture of Gaussians distribution. This way, in a scene containing trees waving in the wind, a new wind blow might not cause much novelty (as expected), whereas a pedestrian appearing in the scene would be successfully detected. Authors propose a different approach here, where the probability distribution is not known beforehand. Instead, a Bayesian framework is used: every new piece of information changes the probability distribution that models what the users expect to see. KL divergence between the distributions before and after the new information is seen is used as a measure of how much novelty the information contains.

A user attention model for video summarization [8]

There are two types of video summarization: static video abstract and dynamic video skimming. The former tries obtaining a set of images (key frames) from the video that are representative while the latter tries finding a set of sub-clips (video+audio) that represent the entire content (with a much smaller total length). This article proposes a framework for video sumarization (both static and dynamic). It treats a video as containing three types of information: Visual, Audio and Linguistic. Different saliency maps are generated from all these different sources and they are combined in order to generate an attention curve for the video. The attention curve is smoothed and the parts of the movie around local maxima are used as summary.

Image saliency mapping and ranking using an extensible visual attention model. [9]

Authors use saliency to evaluate the quality of a photo. They state that the more intense the saliency map, the more interesting is the picture. The model described in [1] and [2] is modified such that normalization is performed over the entire set of pictures instead of normalizing each picture individually.

The economics of attention. Maximizing user value in information-rich environments [10]

The general problem of finding a strategy to display n informational items, on a system that supports only k items at a time is tackled. This problem requires a tradeoff between exploration and exploitation and is solved starting from a formulation of the bandit problem.

3.2 Intelligent advertising systems

AdON [11]

The AdON system also tackles our two problems that appear in video advertising: ranking the ads from a list according to the contextual relevance and finding the position (time and space) for the advertisement. In order to solve the ranking problem, textual information is used (tags, closed caption of the video, OCR analysis). For positioning the ad two aspects must be treated: finding temporal position (it is best for the ad to appear in an interesting moment, when the audience is captured by the movie) and spatial position (it is important that the ad doesn't occlude the important parts of the scene, thus being unobtrusive). However, regarding position, the ad can be placed only at the top or bottom fifth areas. Finding interesting shots (for temporal positioning) is based on the motion intensity and shot duration. The longer and more intense the motion is, the more interesting the shot is. For spatial positioning, besides saliency map, also face and text detection is used. The intrusiveness of a position is measured by summing up the values of these maps over the period of time that an ad would appear (e.g.: 15 seconds) there. The product $Interestingness \cdot (1 - Intrusiveness)$ is used to rank the possible positions (spatial and temporal). In the end different strategies may be used to match the ranked ads and positions (e.g. place the best ad in the best position; or given a position also take into account the ad similarity with current shot).

AdImage [12]

AdImage [12] represents a system for contextual video advertising which consists of three parts: advertiser user interface, image matching system and ad scheduling algorithm. The UI allows the advertiser to specify an image (called adImage) that he wants to bid on (this is similar to AdWords, where the advertiser bids on keywords), the bid value, the total budget and the video ad that he wants to appear. After matching each adImage in the database with each frame of a uniformly sampled video a list of adImages and associated fitting scores will result. Selecting the ad is posed as a nonlinear optimization problem that takes into account the bid and budget of each advertiser while trying to maximize the total revenues. Note that there is also a temporal constraint between video ads, since each video ad is generally a few seconds long, therefore it might overlap with some other.

Contextual in-image advertising. [13]

The ImageSense system puts ads into images. In this context, they also tackle the two problems of matching relevant ads and finding the best position (however, only a spatial position needs to be found for images). In order to match relevant ads, they use three types of information: global textual information (obtained from the webpage that contains the image), local textual information (obtained from the block near the image) and local content relevance.

In order to extract the semantic structure of a web page, VIPS algorithm is used. The algorithm first finds the blocks using the page DOM, and then identifies horizontal and vertical separators between these blocks. The separators will determine a tree-like structure for the blocks in the web page.

After page segmentation, the local textual information can be extracted. It consists of three parts: T1 = title, description, T2 = expansion of words in T1 and T3 = concepts extracted from the content of the image. T2 is obtained by classifying words in T1 into a node of a predefined hierarchy which contains around 1000 words. The image content is also used (it is more important for the case when information in T1 is missing) for extracting additional keywords. A light ontology containing frequent elements in real-world is used (e.g.: animal, crowd, desert, sky, etc).

The ads also contain a textual description (title, keywords, description and hyperlink). Given an image with associated local and global textual information, the ads are ranked by using a similarity measure. One more aspect remains to be solved now - where to place the ads into the image. This is done by taking into account the saliency map of the image. First, the image is divided into $M \cdot M$ (M=5 is used) rectangular blocks and the total salience in each block is computed. Then, blocks are weighted such that the ones that are close to the center receive smaller score. In the end, for each candidate ad, a local content relevance is taken into account: the color similarity with neighboring blocks (using HSV) is computed.

Finally, because the search space is large, in order to find a good matching between image blocks and ads, a genetic algorithm is used.

Visual Contextual Advertising. Bringing Textual Advertisements to Images. [14]

The authors of [14] also try inserting advertisements into photos. They assume no surrounding text exists and use a generative model in order to tackle the problem. The main problem arises from the fact that images and ads are described in image space and word space respectively, so a direct matching is not possible as in traditional web -advertising. The joint distribution P(v, w, t) is estimated, where v denotes an image, w an word and t the advertisement. This distribution is then used in order to calculate P(t—v) and the ad t that maximizes this value is the one that will be selected. The main problem is estimating the joint distribution. Under an independence assumption this is reduced to the estimation of two factors. The first one is P($w|t$) - this value is obtained by dividing the number of occurrences of w in t by the total number of words in t. However, it is also smoothed with a term that considers the entire dataset.

Another term that had to be estimated was the probability P($f|w$), which denotes the probability of a particular feature given a word. This is estimated by using a database of annotated images from Flickr.

3.3 Intelligently finding ads

Advertising based on users photos [15]

Targeting is done using users photos (using photo sharing sites as flickr) - authors extract keywords from users' photos in order to serve better ads. For a given image, the keywords are extracted by searching it in a database (of about 2.4M images - which also contains surrounding text for each image). In order to solve the semantic gap, ODP taxonomy is used (Open Directory Project - http://dmoz.org). ODP contains a tree of concepts, and for each concept a list of web pages. Given the vector of terms for an image, a weighted list of concepts can be generated by applying cos-similarity between image terms and the pages associated to each concept (in order to compute the importance for each concept). Now, an image is described by a weighted list of general concepts instead of a weighted list of terms. The same process is applied on the terms of the ads in order to obtain a list of concepts for each ad. Then three ways of matching ads are evaluated: cos-similarity between terms (this is the baseline approach); cos-similarity between concepts and a mixture between these two (because sometimes simple terms are enough, while finding concepts introduces additional noise).

Finding Advertising Keywords on Webpages [16]

This is a detailed article that describes a machine learning approach for extracting keywords from a web-page. A keyword may be a phrase of 1-5 words. They used the following types of features: linguistic (POS tag), capitalization, hypertext (e.g.: whether the phrase belongs to the anchor text of a link), meta section features (e.g.: whether the phrase appears in the meta section of the page), title, URL, IR features (TF, IDF), location (first appearance within the page, relative to document length), length (the length of the sentence within the word appears, the length of the whole document), phrase length, query log (uses information from MSN query logs - represents queries that people are most interest in).

The detection of keywords starts with a candidate selection. Groups (phrases) of 1-5 consecutive words that appear in the page are taken into account as keywords (ignoring those that cross sentence boundaries). For each such phrase, all features above are computed, and then it receives a score by using a logistic regression trained on some manually annotated pages. An interesting result of this paper was that the MSN query logs are very important for such tasks.

Predicting Ads' ClickThrough Rate with Decision Rules [17]

The algorithm described here tries to estimate the click through rate of an ad in the context of sponsored search advertising. This number is directly related to the revenue generated by that ad through the formula CTR x CPC (where CPC is usually derived through a bidding procedure). Although the problem was studied before, one of the purposes of the system described is to generate human readable explanations and suggestions for improving the ad CTR. This is a reason why decision rules were used for classification. The probability for an ad of being click is modeled as a product of three factors, assumed independent: p_a (determined by the content of the ad), p_s (determined by the number of the current results page) and p_r (determined by the position of the ad within the results page). The value that the system tries to estimate is p_a. The two other factors (p_s and p_r) can be modeled by a single hidden variable (which could be interpreted as a variable that says whether the ad have been seen or not). In the first part of the article, a maximum likelihood approach is used in order to estimate all the three factors for ads which have been displayed at least 200 times. Then, the estimated values of p_a are fed into a machine learning algorithm that generates rules of the form "if conditions

then increase estimated CTR by α" (where α may be negative). Given an ad, if there exists a positive rule for which almost all conditions are met, then some of the conditions that are not yet respected may be used as suggestions.

Video Suggestion and Discovery for YouTube [18]

Starting from the notion of co-view, a graph having videos as nodes can be created. Using such a graph, a general algorithm for recommendations is generated. We can use the idea of co-view in our system, for example by transferring some features of one video to the ones that were also viewed by the same users.

On the impact of sequence and time in rich media advertising [19]

In [19] it is emphasized the problem of ad overloading and proposed a framework for sequentially inserting banners on a webpage. The sequence of ads is important and an irrelevant first ad may destroy the effect of the subsequent ones.

3.4 Video comprehension

Video retrieval and summarization [20]

Describes the main processes involved in video indexing and retrieval. These are:

- Video content analysis (mapping features from the video to semantic concepts). It is emphasized the fact that a multimodal analysis (visual, audio, text) would be more effective than taking into account only visual features.
- Video structure parsing (refers to video segmentation into individual scenes). A scene may be composed out of more shots, which are marked by a camera turn on/off. Shot detection can be performed by using only visual elements, however, while trying to merge the shots into scenes one could benefit from information provided by the other channels.
- Video summarization. Refers to creating an abstraction from the original video that would be much shorter, while still preserving important information that could be used for further analysis.
- Video indexing. Refers to methods of storing the content in order to efficiently retrieve the results of a query. This is a non-trivial problem, because we may want to query a large database not only by using keywords (but a set of visual features, for example).

ANSES - Summarization of video news [21]

ANSES generates summaries of video news. Each night, it captures video with subtitles from BBC news. Then, it segments the video into distinct stories and generates a summary for each one. First, in order to obtain the distinct news, video shots identification is used. In order to segment the shots, a histogram-based approach is used. Basically, a significant change in the color histogram indicates a shot change. However, using this simple approach a proper threshold has to be found. Instead, in order to detect whether a given frame is on the shot boundary, the ANSES system analyzes 16 frames either side of the current frame.

After this, for story segmentation, some shots have to be merged (it is assumed that story boundaries occur only on shot boundaries). The shot merging is performed by using the closed

captions (subtitles) contained into the video. First, named entities are recognized using GATE (General Architecture for Text Engineering). Then each text segment is compared to 5 neighbors each side. If two words are identical, then the similarity of the two text segments is increased taking into account the distance (the larger the distance, the smaller the increase) and the type of matching words (e.g.: if the word names an Organization, the bonus is higher than if it was a date or a simple noun). If the similarity between two text segments is high, then their shots are merged (and also all frames in between).

Segmentation of Lecture Videos Based on Text- A Method Combining Multiple Linguistic Features [22]

This article describes a method for segmenting video lectures. Techniques like the histogram-based segmentation cannot be applied here since the visual cues that such methods rely on are not present in video lectures. Their approach is to use textual information extracted from the audio track. A sliding window technique is used: when the similarity between the current position and an adjacent one is below a threshold, the corresponding position between the two windows is considered to be a boundary. There are more types of linguistic features, for example: noun phrases (NP), verb classes (VC), word stems (WS), topic words (TNP - which are NPs with more than one term). In the "verb classes" type, two verbs are considered identical if they are synonyms or hypernyms within two levels in WordNet. There is also a more complex feature that models combinations of noun-verb (NV). For each feature type, a cosine-similarity measure is computed between the two windows, and all these are weighted and added up. The weights are manually chosen, such that $S(TNP) > S(NV) > S(NP) > S(VC) > S(WS)$.

3.5 Understanding how users process information

Assessing the effects of animations in online banner ads [23]

In order to analyze the effects of advertising, understanding how ads are processed by the users plays an essential role. The hierarchy of effects provides a conceptual tool for modeling the consumer behavior. It states that a consumer moves stepwise, in a very ordered way, from product unawareness to actual purchase. The main stages are cognition, affect and conation. Using this framework, the authors state that animated ads are more effective than static ads. A few hypotheses were formulated and then an experiment was conducted in order to verify them. The hypotheses they made were: "An animated banner ad will have greater attention-getting capability than a static banner ad", "An animated banner ad will result in better memory than will a static banner ad", "An animated banner ad will generate more favorable attitude toward the ad", "An animated banner ad will have higher click through intention" and "Product involvement will play moderating roles on the effects of animated banner ads on attention, memory, attitude toward the ad and click through intention". So the first four hypothesis state that an animated ad will perform better in all phases that compose the hierarchy of effects, while the last one says that the performance is moderated by user involvement. A 2 (ad type: animated vs. static) x 2 (involvement: high vs. low) experiment was conducted on 55 subjects and it confirmed all of the above hypotheses.

Effects of attention inertia [24]

Users follow a path to the goal. This study reveals that more attention to ads is accorded in the beginning/ending of this path. Similar ideas may be extrapolated to video - as we are

interested in finding the moment when a video captures most user attention, the ideas in this document suggest that some good candidates may be the beginning/ending of a scene.

An intuitive model of perceptual grouping for HCI design [25]

Tries to explain the way humans group perceptually similar objects (e.g.: the buttons on a remote control that have the same color). Simple segmentation algorithms don't treat discontinuities. An initial approach to tackle this would be to apply some blurring (that would unify distinct regions), however, this is too dependent on background color / elements color. The solution was to move into a higher dimensional space, by adding another feature (as color or orientation) and perform blurring there.

Media Strategy vs. Content Strategy in Online Advertising [26]

Different strategies should be employed depending on the goal-directedness of the user. A user may be goal directed (if he is browsing with a specific purpose in mind) or not (i.e. browse only to kill time).

Under the ELM framework (which has as the essential factor the degree of elaboration that the user is expected to make on content) some hypotheses were proposed and then verified through an experiment. Basically, during an advertising campaign, the advertiser can change different variables in order to optimize his revenue. He may choose for example Media Strategies (making variations in ad presentation) or Content Strategies (e.g.: choosing between informational appeal and emotional appeal). The main hypothesis of this article states that goal directedness determines how effective Media Strategies and Content Strategies are. Specifically, Content Strategies will be particularly effective when the user is goal directed, whereas Media Strategies will be effective when the user is non goal-directed.

Chapter 4

Functionality overview

In order to give a context to the algorithm descriptions that follow, we first describe the main use cases of our system.

There are three kinds of users for our system:

- The regular user, who watches videos.
- A publisher, the one who comes with new videos.
- The advertiser, the one who uploads ads.

For simplicity, publishers and advertisers have been treated as a single entity "administrators".

In the following discussion these kinds of users will be reflected by a persona: Stephan will be a generic regular user, while Roger will be an administrator.

4.1 Main use cases

4.1.1 User

The user watches a video

1. Stephan types in the browser the URL of the website.
2. A list of available videos is displayed.
3. Stephan clicks one of the videos.
4. A webpage containing an embedded video player is shown (see figure below).
5. Stephan clicks play. During the video, ads are shown.

4.1.2 Publisher

The publisher uploads a video

1. Roger types in the browser the URL of the admin panel.
2. An authentication window is shown.

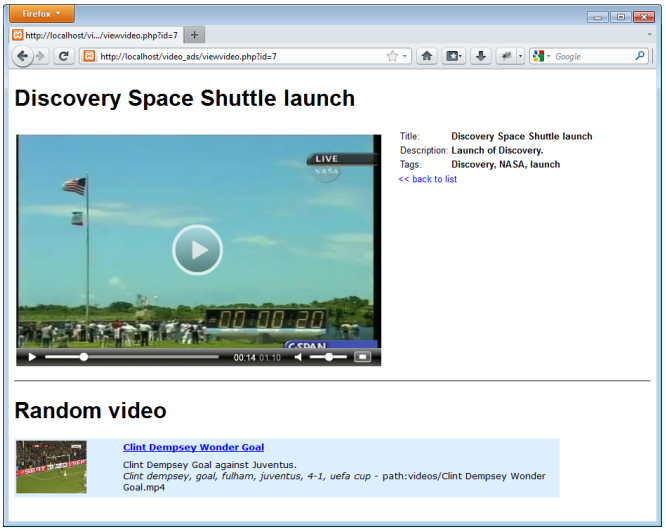

Figure 4.1: Watching a video

3. Roger enters his credentials.
4. The admin center web page is loaded in Roger's browser.
5. Roger clicks the "add video" link.
6. A web-form which allows Roger to introduce video information is shown.
7. Roger selects a local video file and enters additional upload information (tags).
8. Roger clicks the "Upload" button.
9. Roger is informed whether the video was successfully uploaded or not.

The publisher deletes a video

1. Roger logs in to the admin center (steps 1-4 from the uploading video use-case).
2. Roger clicks the "Change video" link in the menu on the left.
3. A list of videos is shown, each one having a "delete" link aside (see figure).
4. Roger clicks the "delete" link.
5. Roger is informed that whether the video was successfully deleted.

4.1.3 Advertiser

The advertiser uploads an ad

1. Roger logs in to the admin center (steps 1-4 from the uploading video use-case).
2. Roger clicks the "add image" link

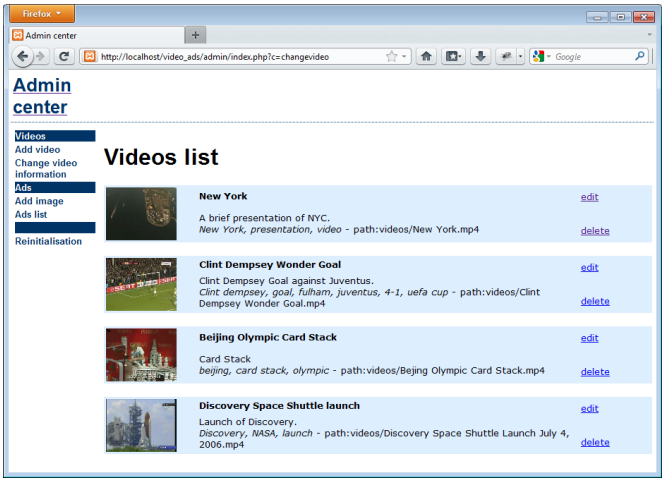

Figure 4.2: Administrator user interface

3. A web-form which allows Roger to introduce ad information is shown.
4. Roger selects a local image (the ad) and enters additional upload information (tags).
5. Roger clicks the "Upload" button.
6. Roger is informed whether the advertisement was successfully uploaded or not.

The advertiser deletes an ad

1. Roger logs in to the admin center (steps 1-4 from the uploading video use-case).
2. Roger clicks "Change ad" link.
3. A list of advertisements is shown, each one having a "delete" link aside.
4. Roger clicks the "delete" link.
5. Roger is informed whether the advertisement was successfully deleted.

I will briefly describe the operations involved by the use cases above. More details about the implementation will be provided in the following sections, this is only meant to offer an overview about when the actual computations are performed. Once a user uploads a video or an ad, some features are extracted (e.g.: saliency maps, color palettes), in order to easily insert ads into videos later. When a user wants to watch a video he receives a video stream which also includes advertisement. The matching between the video and the ad should be an operation with a very small cost since it is intended to be done in real-time. This is feasible because of the pre-computations that are done both for video and also for the ad, at upload time.

Chapter 5

Architecture

The main parts of the system are illustrated in Figure 5.1 below. The server-side processing part has three main modules, one corresponding to each type of users.

Once the publisher uploads a video, various computations are performed in order to be able to quickly retrieve this information when necessary. Similarly, when the advertiser pushes an ad, some features are extracted. Finally, when a viewer requests one of the videos, we need to be able to rapidly match the respective video with the corresponding ads, and also to find a good placement for the respective ads. This is the only module that needs to do calculations in real-time, right when the user chooses a video. All the others are supporting it in achieving this goal.

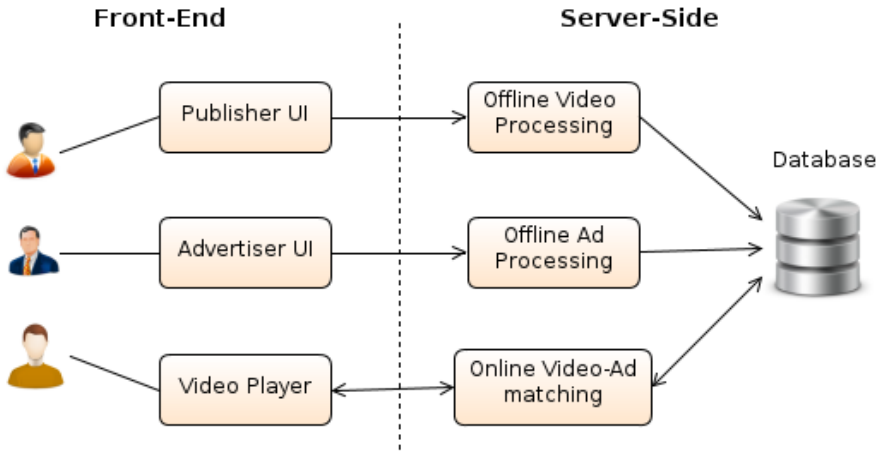

Figure 5.1: High-level architecture

At this point it is worth making a short discussion regarding the chosen architecture. While the offline video and ad processing modules perform some precalculations in order to make the online matching more efficient, one may note that the computations that need to be performed by the latter module are also non-trivial, since no connection is made at this point between particular ads and videos. It would be a legitimate question to ask whether we could avoid this module and incorporate its computations in the offline processing modules, and compute everything at video or advertisement upload. While it would be theoretically possible to invoke

the matching module after each video and advertisement upload, in order to have up to date information the chosen approach of computing part of the video-ad matching in real-time has a series of advantages:

- The chosen architecture allows incorporating user features (e.g.: what videos were viewed before), so that the matching takes the particularities of the user into account. It is obvious that this may not be pre-calculated, because of the large variety of possibilities.
- There are multiple, granular events that may happen, and requiring to have up-to-date matches after each of them may not scale well: ads may be deleted, they might expire, or their budget may have passed the limit.

To conclude this discussion, it would probably always be possible to find a different balance for the tradeoff between having up-to-date, personalized information in real-time on one hand, and performing as few calculations as possible when serving the video on the other hand. There is also room for more innovative techniques here, like having a cron job which would update the video-ad associations daily, and afterwards only requiring to perform the entire computation only if, for any reason, the respective association is not available anymore.

Next, I will briefly describe the main elements presented in Figure 5.1, as well as their sub-modules. This section will only describe the desired functionality, but will not detail the particular algorithms.

5.1 Front-End

The front-end consists of one module for each type of users, since each one has a different UI for performing his actions.

Publisher UI

The publisher is shown an UI which allows him to add new videos, along with a brief description/tags for that video. The few keywords inserted there may be useful for selecting proper ads, and they represent the essential element for which the advertisers may bid on.

Advertiser UI

The advertiser UI consists of a form that allows adding a new ad, along with few other information like the target URL (if the viewer clicks the ad) or the keywords that represent the desired context in which the ad should appear.

Video Player

The final user, the one who watches the videos along with the inserted ads online is shown a typical video content web site, being allowed to browse and search a list of videos.

5.2 Server-Side

As shown in Figure 5.1, the server side consists of Database, and three components: Offline Video Processing, Offline Ad Processing and Online Video-Ad Matching. We will briefly describe the role of each in the following subsections.

Offline Video Processing

The purpose of this module is to make the costful operations regarding video processing only one time, after the video is uploaded. It may include operations such as:

- Saliency calculation.
  We would like to find which regions of the original video are more likely to attract user gazes. Since these regions may represent essential spots of the video, we will try to avoid them when inserting the advertisement.
- Video segmentation.
  The structure of the video is another useful information that is extracted at upload-time. In order to avoid some cumbersome effects, we may discurage ads from passing from one scene to another.
- Scene intensity evaluation
  As stated in the problem description chapter, one of the hypotheses that we make is that it is better to have the ad into the intense scenes, as long as they do not overlap the important objects. While scene understanding is a difficult problem, various heuristics may be used in this regard, such as estimating the motion in the scene. Furthermore, as an improvement, different channels such as sound could be used here.
- Color palette extraction
  Color palette extraction refers to identifying a few most representative colors for the video. In order to have a visual pleasing effect when inserting an ad, it is preferable to select an ad that has the colors as close as possible to those of the video.
- Object detection
  Although not implemented in the current thesis, one possible improvement in order to have a better match between the video and the ads would be to automatically extract some semantic knowledge from the video, such as which objects are present there.

Offline Ad Processing

Similar to the offline video processing, this module is responsible for computing only one-time information about the ad. It is invoked after the advertiser pushes a new ad and it is responsible for storing into the database information about the advertisement. Although the offline processing modules that were already mentioned are invoked in a synchronous way, this is not a strict requirement. They might as well be added into a queue and processed later, in order to improve the publisher and advertiser experience.

Online Video-Ad Matching

This module is invoked when the final user chooses to view a particular video. When the request is made, the Video-Ad Matching module should find the ads which will be inserted into the given video, and also find some appropriate spatial and temporal positions to insert them. It basically consists of two sub-modules:

- Ad ranking module.
  Its purpose is to find those ads that are most related to the video. In the current implementation this also consists of two separated parts, since there are two criterions to look at, when evaluating the degree to which a given video and an ad fit together.

The first criterion to be taken into account is the semantic similarity. Advertisers may insert keywords along with their ads in order to express their preference toward certain kind of videos.

A second criterion that is considered is the visual similarity. As mentioned before, having the ad blend better into the video would result in a better overall experience for the user, also with positive implications for the publisher and advertiser.

- Ad scheduler.
  After having a list of ads ranked according to how well they fit into the current video, the next step is to find the spatial and temporal position of the ad. This is supported by the pre-processing that was done for both the ads and for the video, which save us from doing at this point the costful operations such as video saliency extraction. The specific algorithms that are used will be described in the next section.

Database

The system needs to store the videos and the ads, as well as the additional information. The additional information may be explicitly mentioned when uploading the video or the ad (such as video title, description, tags or advertisement keywords), or it may be computed, and stored in order to fasten later calculations.

Technologies

The main technologies that were used in implementing the architecture above are: Java, JNI, Xuggle, HTML 5, PHP, MySQL, WordNet.

- Xuggle.
  Xuggle is a Java library that allows uncompressing, modifying and re-compressing media files. We used Xuggle before the feature extraction step (to decode the video) and also in the end, when the results from the ad insertion module were available, in order to create a new video file which also contains ads.
- HTML, Javascript, CSS.
  We used these technologies to create the user interface, both for the regular user (who watch videos) and for the administrator.
- PHP, MySQL.
  MySQL and PHP are used in order to store video and ad information, and to implement the server side part for the operations presented in the user interface.

Alternative architecture

So far, we did not make any tradeoffs in favor of computation speed and maintained our algorithms as expressive as they could be, since this thesis aims to explore Artificial Intelligence approaches that might be used in the given problem. However, in a production environment it may be the case that we may need to cut some of the execution times in specific points. One sensitive such point is the computation that is done online, when the final user requests a video. Although we mentioned earlier in this section that it is theoretically possible to move part of the online computation in their pre-computation counterparts, but we have also enumerated the certain drawbacks that would be met. However, a conceptually different approach would have been to take into account that the online advertising domain is subject to certain standards.

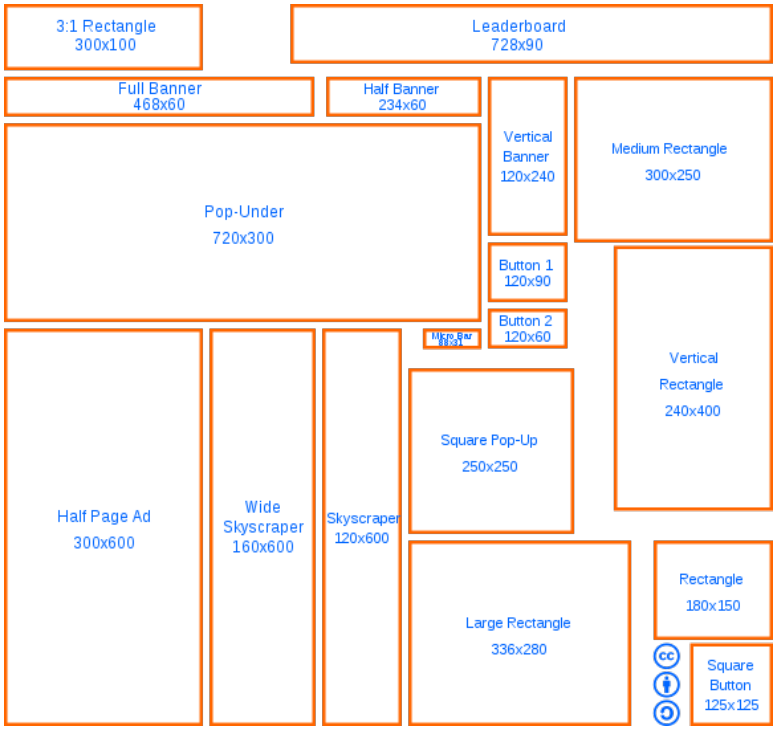

Figure 5.2: Standard Ad Sizes

One information that is very useful when looking where an ad fits is that there is a set of standard ad sizes, which are shown in Figure 5.2. Having this in mind, we may compute offline a few spots where each of these ad sizes fit for a certain video. Later, when a specific ad is chosen for a particular video, we will only look at the best possible placements for that ad size, and choose one of them (for example choose the one that also results in a favorable color combination with the specific ad). One restriction that this approach imposes is that we would be limited to the standard ad sizes.

Chapter 6

Implementation details

We will describe the algorithms involved in the research part of the project, that of placing ads into video. As already stated, there are two main research parts involved: selecting the proper ads from a larger database that are going to be inserted into the video and selecting a proper placement (both spatially - in unobtrusive areas, and also temporary - during important scenes) for those ads.

6.1 Selecting ads

Multiple approaches are possible to select the ads that best fit a given video. First, there is a semantic approach, which would try to match ads and videos based on content. Semantically related ads are more likely to be of interest for the user [37]. The semantic matching is difficult because ideally it would involve understanding the video. However, ads are usually annotated with some keywords in order to give hints regarding the kinds of video that the advertiser would like to appear in. Video may be tagged too, or some words could be extracted from the title. Even if a few words are available regarding the content of the video, matching the ads is still difficult (an appropriate ad may be ignored just because the words did not perfectly match) - possible approaches would be to use an ontology (e.g.: WordNet) and extract related words using the tags, or LSA, which would mean to obtain a similarity by analyzing how often two words appear together in a large corpus of texts. Different approaches for selecting the ads given the video may be imagined by taking into account the visual similarity (instead of semantic similarity) between the ad and the video. We used a hybrid approach in this thesis, first selecting a number of candidate ads based on semantic similarity, and afterwards filtering the list to use those with a visual similarity.

6.1.1 Semantic similarity

WordNet is a lexical database for English, but similar projects are being developed for other languages. Nouns, verbs, adjectives and adverbs are grouped by synonymy, in groups called synsets. A synset is actually an equivalence class, as replacing a word with another one in the same synset does not change the meaning of an utterance. Besides this grouping, WordNet also offers semantic relations between these synsets. These relationships vary depending on the part of speech. For nouns we have hypernyms (Y is a hypernym of X if and only if any X is also an Y), hyponyms (Y is a hyponym of X if and only if any Y is also an X), coordinate terms (X and Y are coordinate terms if they share a hypernym), holonyms (Y is a holonym of X if and only

if X is part of Y), meronyms (Y is a meronym of X if and only if X is a holonym of Y). Similar semantic relations are defined for verbs: hypernyms, troponyms, entailement, coordinate terms; adjectives: related nouns, similar to, participle of verb; adverbs: root adjectives. See [33] for more details. Wordnet also offers the polysemy count of a word: the number of synsets that a word is belonging to. If a word belongs to multiple synsets (i.e. homonymy) then some meanings are probably more frequent than others. This is quantified by frequency score (which was obtained by manually annotating large corpuses with the corresponding synset of each word).

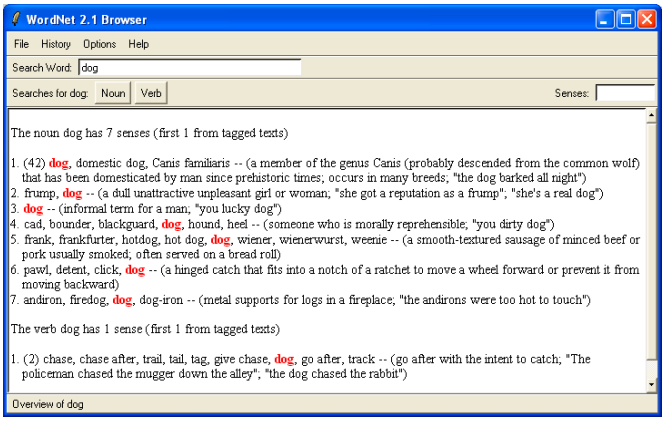

Figure 6.1: WordNet browser

A fundamental problem that may be answered by using an ontology such as wordnet is the semantic similarity of two words. Given all the available relations mentioned above, we may design a basic algorithm which takes into account the shortest path between the compared words. A path represents a sequence of words, each consecutive two words being connected by a similarity relation (e.g. holonymy). The issue with such an approach is that it treats uniformly all the relation types. More elaborate measures were developed around WordNet, one example bening the Resnik measure [34]. This aims to capture the similarity of two words by taking into account the information brought by the most specific concept that subsumes them. More specifically, given two concepts, C1 and C2, we are only looking at the "is-a" relations, going up in the hierarchy until we have found a common parent, P. We can measure how informative the subsuming concept is by summing up all frequency counts of the words representing or subsumed by that concept (i.e. if the hierarchy would contain one top concept which subsumes any other concept, then that top concept would have frequency 1, and informativity 0). Figure 6.2 shows an example hierarchy which has "taxi" and "airplane" as starting concepts. The arrows represent hypernymy ("is-a") relations. As shown, the lowest common anchestor for the two starting words is "vehicle", therefore the Resnik measure is in inverse ratio to the frequency of the "vehicle" concept. One advantage of using the Resnik measure which relies on term frequency against a shortest path is that it is independent of the terminology density around the concepts on the path (as an example, if only a few terms exist in a certain domain then this would make it possible to reach general concepts within a very small number of steps).

We have illustrated so far an algorithm that may be used to measure the similarity of two concepts. However, the problem that we need to solve is a different one. We are given a set of words/tags which describes a video and need to find an advertisement that is semantically related, based on the keywords that describe the advertisement. These keywords may actually represent words that would be targeted by the respective ad. Basically, in terms of information

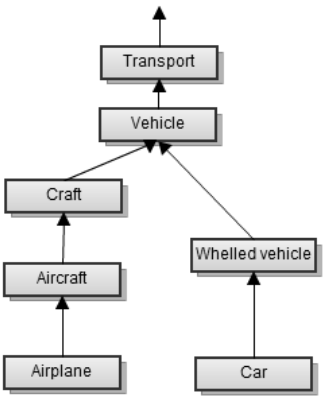

Figure 6.2: WordNet Is-A relations

retrieval, the video keywords represent the query, while the advertisement descriptions represent the documents. The problem that we are facing is a special case of an information retrieval problem that is typically tackled using a standard TF-IDF approach. We shall not use this measure because of some particularities that our data has.

- The documents only contain a few words.
- All the words in the documents are important (as an example, they all may be obtained through a bidding on keywords process).
- As the query also contains only a few words, it may be the case that no document contains a term in the query.

A first decision was to not use an IDF term. If multiple ads are targeting a certain keyword, we would not like to make it less important. Also, we would not have repeating keywords, so TF is not used either. We would therefore use a different similarity measure. We design the algorithm by first stating a key behavior that we would like it to follow:

- When two ads have the same number of keywords, the one that matches more keywords with the video is preferable.
- When two ads match the same number of keywords, the ad with a shorter description is preferable.

A formula that satisfies the above conditions would be $\frac{|Ad_{description} \bigcap Video_{description}|}{f(|Ad_{description}|)}$, where $Ad_{description}$ and $Video_{description}$ represents the set of keywords describing the ad, respectively the video and $f : \mathbb{N} \to \mathbb{R}$ is a non-decreasing function. Unless otherwise specified we used $f(x) = 1$ for this function. Finally, due to the scarcity of words that may describe both the ads and the video we would like a softer version of that formula, which would act like it when all words in video and ad description are either identical, or completely unrelated. We use the Resnik method described above in order to measure the word similarity. Instead of counting how many words are shared between the video and the ad, we would, for each word in the video description, take into account the most similar word in the ad description using the Resnik measure (which always yields a number between 0 and 1) as the weight. The formula obtained

is the following:

$$SemanticDistance(Video, Ad) = \frac{\sum_{V \in Video_{description}} \max_{A \in Ad_{description}} Resnik(V, A)}{f(|Ad_{description}|)}$$

where $f : \mathbb{N} \to \mathbb{R}$ is a non-decreasing function. Given a video, the similarity score of the description is calculated for every ad, and the ones with a higher score are more likely to be inserted into that video.

6.1.2 Visual similarity

.

Another variable that we take into account is the degree of visual similarity between the video and the ad. Intuitively, if the ad and the video only contain contrasting colors then they do not fit well together. The approach we used is based on extracting the color palette from the ad and also from a few frames of the video. We could imagine that the video queries the ads database with a list of colors (extracted from the video), and the most "similar" ads are retrieved. The intuition behind this approach is that by selecting visually similar ads the effect is less intrusive, leading to a more pleasant experience. The color palette of an image represents a set of representative colors for the image. We used a library compiled from the C implementation of the program available at http://kuler.adobe.com/.

Given a video and an ad, a distance is computed by using the ad and video color palettes. For each color from the video palette we only consider the most similar color in the ad - the most similar color is the one having the smallest Euclidian Distance in HSB space. In order to compute the ad distance, we add the logarithms of the distances defined above (between each color in the video palette and the most similar one from the ad color palette).

$$VisualDistance(Video, Ad) = \sum_i log(\min_j dist(pallete_{video}[i], pallete_{ad}[j]))$$

Adding the logarithms is equivalent to multiplying the quantities inside, but we used the logarithm in order to avoid numerical problems, because the HSV values are in [0, 1], so the product of distances may go to 0 fast.

However, in order to explain the intuition behind the above formula, it is easier to think of it as a product of distances. Having a product of multiple values for the distance we favor the case when one of the values is close to 0 - this can be interpreted as a softer version of an algorithm that takes into account only the most similar pair of colors.

6.1.3 Combining the semantic and visual similarities

So far we obtained two unrelated measures, each indicating how desirable a certain ad is. We need to combine these two values in order to select only a few ads for the given video. One approach would be to combine the two measures into a single number. Possible approaches would be to compute the sum, or the product of the given measures. Actually, any function f which has the semantic similarity and the visual similarity as parameters, that would satisfy $f(s_1, v_1) \geq f(s_2, v_2)$, whenever $s_1 \geq s_2$ and $v_1 \geq v_2$ would make a good, but arbitrary choice. A more principled approach would be to first establish a general form of the function that would mix the two similarity measures, and afterwards to learn certain values for the parameters. One example function would be $f(s, v) = w_1 \cdot s + w_2 \cdot v + w_3 \cdot s^{w_4} \cdot v^{w_5}$, where $\overline{w_{1..5}}$ are real-valued parameters that may be learned. We would first need to manually annotate the desired

mixed similarity value for multiple (semantic similarity, visual similarity) pairs, and afterwards find the $w_{1..5}$ values which minimize a certain type of error. One approach in finding these values would be a stochastic method such as simulated annealing, gradient descent or genetic algorithms. Another approach for this particular case would be to fix various values for w_4 and w_5 and, since we are left with a linear equation, use the fact that we then have a closed form which allows rapidly computing the rest of the weights (if using the least squares as the method for computing the error). This is a possible approach that may be used for obtaining a single value starting from the semantic and visual similarity. However, we used a simpler method that gives more control of which ads to select. First, the ads are sorted in descending order of similarity, and only a few of the top performing ads are considered for the next stage, where the final ad selection is based on the visual similarity score. We may control the outcome of this process by changing the number of ads to be considered after the semantic similarity scores are computed.

6.2 Placing ads

We associate a cost for each possible ad placement inside a video. The cost is given by the total important information in the original video covered by the ad, which cannot be seen anymore because of the ad. We use visual saliency in order to quantify how much information a region of the movie contains.

6.2.1 Saliency cost

The program uses the idea described in Guo et al [31] in order to detect how salient a region in an image is. First, the 2D Fourier Transform for the input image is computed, then the complex values which resulted are normalized such that each will have module 1, and in the end an Inverse Fourier Transform is performed. This way, components that appear frequently are reduced while discontinuities are enhanced. The explanation is that a frequent component in the spatial domain corresponds to a greater coefficient at the corresponding position in frequency domain - and therefore the normalization will lead to diminution of that complex number from frequency domain and to suppression of the periodical components. In order to use more than a single channel (or two - which would be obtained by using complex numbers) to detect discontinuities on, a generalized Fourier Transform is used (Quaternion Fourier Transform). Quaternions represent a generalizations of Complex numbers, taking the form $Q = a + b \cdot i + c \cdot j + d \cdot k$, where a, b, c, d are real numbers. The Quaternion Fourier Transform which we based our implementation on is described in Sangwine Ell [32]. Using quaternions we are able to integrate discontinuities over different maps (channels) in a natural way. Three of the four available channels are used for static image analysis, while the fourth incorporates temporal information (through the difference between frames):

- Channel 1: pixel intensity
- Channel 2: r-g. Sensitive to red-green contrast. Although the intensity might be the same, red areas contrast with green areas.
- Channel 3: y-b. Sensitive to yellow-blue contrast.
- Channel 4: intensity (t) - intensity (t-1). It models the motion in the given frame (compared to the previous one).

After running the above saliency algorithm for an image (a video frame), we obtain a map of values, each one representing the saliency of the corresponding pixel in the image. Because salience calculation is very costly, we only computed the salience for one frame every second. However, this is not a problem for most videos, as if one important object appears in the scene, it is usually caught by at least one of the key frames we considered.

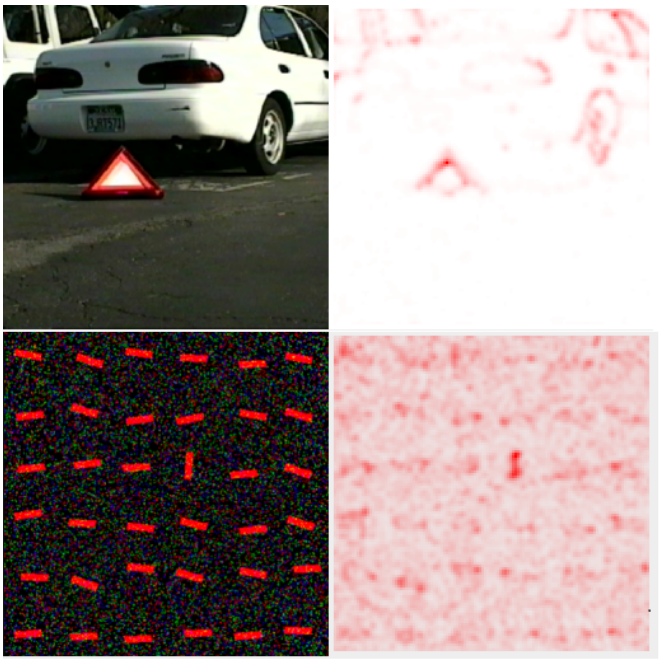

Figure 6.3: Saliency maps

The figure above shows the saliency we obtained for two test images. On the left side, the original images are presented, while on the right side the saliency map obtained is encoded such that higher salience is represented by a more intense red shade.

6.2.2 Finding the position with minimal cost

We formulate the optimization problem that we considered for placing an ad as minimizing the sum of pixel saliencies which are covered by the ad over the frames that it appears in (we only consider the frames which we calculated salience for, as described above).

In order to find the position that minimizes the above cost, we verify each possible placement (restricting the ad to start from one of the frames with computed saliency), and choose the one which leads to a minimal value.

It is essential to be able to efficiently compute the cost of a given placement. By using an auxiliary matrix which takes $O(W \cdot H \cdot T)$ to compute (where W and H represent the dimensions at which the video is processed, and T the length of the video), but which is computed only once, we can evaluate each placement in $O(1)$. There are $O(W \cdot H \cdot T)$ possible placements, therefore the overall complexity of this algorithm is $O(W \cdot H \cdot T)$.

The auxiliary matrix we construct in order to be able to obtain the cost of any placement in $O(1)$ is a three dimensional matrix which contains at position (x, y, t) the sum of the saliencies of all the pixels in the (1, 1) - (x, y) rectangle, for all frames from 1 to t. Using the inclusion-

exclusion principle, we are able to calculate the total salience in an arbitrary rectangle using only a few additions and subtractions.

$$Aux[X][Y][T] = \sum_{t=1}^{T} \sum_{x=1}^{X} \sum_{y=1}^{Y} Cost(x, y, t)$$

6.2.3 Placing multiple ads in a video

We explained so far how we can find the optimal position for an ad in a video. However, we will have multiple ads in the same video. We used a Greedy strategy in order to place multiple ads in a video: first, the optimal position is chosen for the first ad (ads are considered in the order given by the ad-selection part, which is based on the color palette). Afterwards each ad is inserted by only considering the positions that do not overlap with previously placed ads. We considered that two ads overlap if they appear in the same time, or the distance between the ads is below a predefined threshold (1 second). So we will not have more than one ad at a time.

6.2.4 Refining the cost using video segmentation

We used scene segmentation in order to avoid having ads that cross the boundaries between scenes.

The chart below illustrates the evolution of histogram change over consecutive frames. This change is usually small for continuous scenes but a large difference occurs at the boundary between scenes, therefore these values are useful for scene segmentation. The horizontal axis

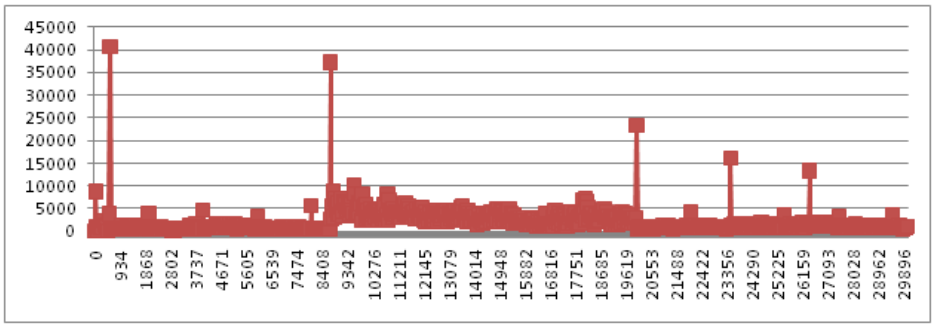

Figure 6.4: Histogram differences between consecutive frames

contains timestamps (in milliseconds), while the vertical axis represents the distance between the histogram for the frame at the current timestamp and the histogram of the previous one.

togram contains multiple bins, each of the bins counting the number of pixels that have a certain color. A histogram can be viewed as a point in a multidimensional space, each dimension corresponding to a bin. This way we can define the difference between two histograms as the Euclidian distance between the two corresponding points. Note that this definition of distance does not take into account the fact that some bins are actually strongly related if the corresponding colors are similar. It is possible for a certain pixel in the image to slightly change its color (only due to small changes in light, for example), and this may make the pixel belong to another bin of the histogram. However, although the two bins are close to each other, the distance definition does not take this into account. There are a few solutions for avoiding such problems: one would be to use a sophisticated distance measure which would account for the

bins similarity. Another approach would be to distribute a certain pixel across the color bins in the nearby. We could overlap a Gaussian distribution centered in the bin that matches the color of the pixel, and gradually decrease the pixel contribution to the more distant bins. This way, when a pixel slightly changes its color, we only get a small change for each bin; a large change instead produces a significant difference for each bin, as desired. Figure 6.5 illustrates the histogram update process, where bins representing similar colors are placed nearby on the Ox axis.

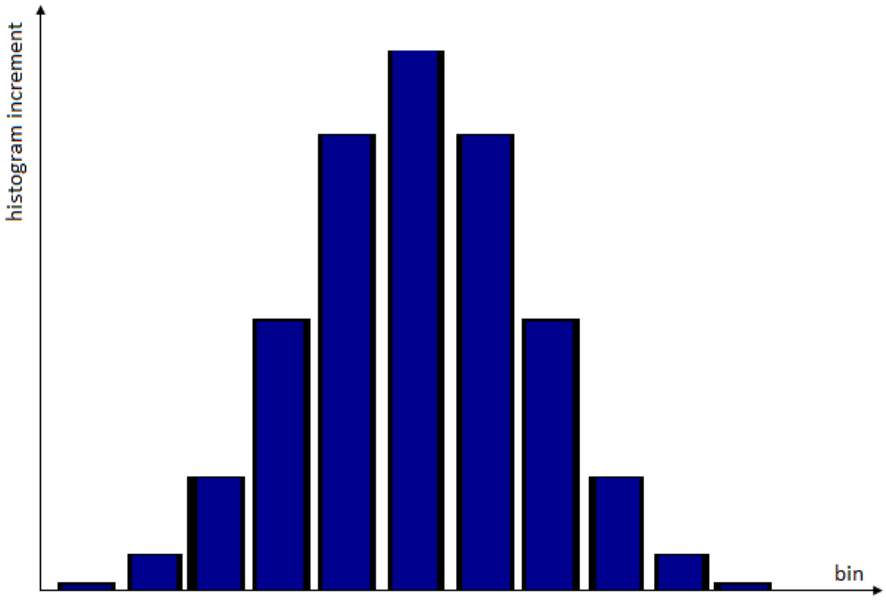

Figure 6.5: Soft increment of the bins for the histogram

A simpler approach (that provides good results) is to only use a few bins for each histogram: instead of having a bin for each R, G, B combination, we could group similar colors in a single bin. This is a robust method that provides good results in practice, and also has the advantage of being more computationally efficient, because we have to handle fewer bins. Figure 6.4 above shows the results obtained using this technique. The analyzed video contains 6 different scenes. The 5 separation points indeed correspond to the highest differences (y-values) in the chart (0.5 seconds, 8.7 seconds, 23.2 seconds, 16.1 seconds, 26.4 seconds). In order to obtain these separation points, we checked which differences are significantly above the average of their neighbors, taking into account 100 points before and 100 points after.

The timestamps associated with the boundaries between scenes are used in the ad placement heuristic: the cost function was multiplied with a penalty factor when an ad crosses such a boundary.

6.2.5 Placing the ads in high interest scenes

Another heuristic derived from the above histogram method is used in order to estimate high interest scenes. Some of the scenes generating a high interest are highly dynamic, resulting in faster changes in the histograms from a frame to the next one. In the figure above, such a scene is the one delimited by the second and the third separation points (8.7 sec - 23.2 sec).

The average histogram change in a specific scene is taken into account by incorporating it into the cost function (a lower cost results when placing an ad strictly inside a more interesting scene).

6.2.6 Experimenting new ways of placing ads

We created a new method of presenting the ads by allowing them to change their position in the movie. Changing the position may allow an ad to avoid overlapping salient regions that could not otherwise be avoided. The procedure was to set a speed variable for each ad, penalizing large velocities. The cost function is now defined as

$$Cost = (\alpha + \nu^{\beta}) \sum_{t=T_0}^{T} \sum_{x=1}^{X} \sum_{y=1}^{Y} salience(x, y, t).$$

The main problem of this approach is that the efficient solution that used dynamic programming can't be used anymore. Lacking an efficient procedure for computing the optimal solution (the one that minimizes the above cost), we relied on a stochastic algorithm that tries various random placements (starting moment, position, speed), evaluates the cost for each of them, and selects the best performing one.

- Applying CSS Shaders so that the inserted advertisement blends into the video

Delivering the ads to the final users may be done via HTML5, using the <video>tag. This allows the use of an innovative, under development technology called CSS Shaders, which defines "a filter effects extensibility mechanism and provide rich, easily animated visual effects to all HTML5 content". More specifically, it allows applying vertex and fragment shaders directly on HTML elements.

There are two types of effects we can apply so that the ad fits better into the video:

- Spatial transformations: move the corners of the ad so that it fits better into the context, and then apply the affine transformation which makes the rectangular ad become the one with the new corners.
- Color transformations: one simple effect is to gradually increase the transparency of the ad as we are getting toward the edges, making thus a smooth transition to the ad.

Chapter 7

Experimantal Results

We shall make a qualitative evaluation of the results generated by our program. in order to have quantitative results, the system should be run in a production environment and the performance of the ads (e.g. CTR) obtained while using our approach should be compared to that obtained using other approaches. Having an accurate measure of whether the inserted ads have a positive impact is difficult to obtain. Ideally, one should consider some groups of subjects and follow their behavior with regard to the brands over a longer period of time.

We first separately evaluate the performance of two main parts of the system: choosing the semantically similar ads, and placing the ads in an unobtrusive way. Later we will also evaluate the performance of the entire system. The evaluation dataset consists of 5 videos with different topics, each of which having between one and five minutes length. We afterwards selected 25 ads such that there are at least 5 ads that fit well with each video. Note that it may be the case that some ads could be representative for more videos. Each video and each ad were tagged with a set of 2-5 keywords.

Evaluation of the semantic similarity matching

For the semantic similarity evaluation we manually scored each ad selection with a value from 1 to 3, where a score of 1 is assigned when the selected ad is completely unrelated or undesirable for the topic of the video, while a score of 3 is assigned if the ad is directly related to the video. Five advertisements were selected for each of the five videos, so there is a potential perfect match, since as we mentioned there are at least 5 advertisements that fit each video. Table 7.1 below shows the average scores for each of the videos.

Video	Score
Video 1	2.4
Video 2	2.0
Video 3	2.0
Video 4	1.8
Video 5	2.2
Average	2.08

Table 7.1: Average ad selection score per video

The average score over all the 25 ads is 2.08. The graph below shows the evolution of

precision, depending on what is considered to be the acceptance level. For example, if the advertisements with a score of 2 or more are considered correctly selected, then the precision value is 0.64.

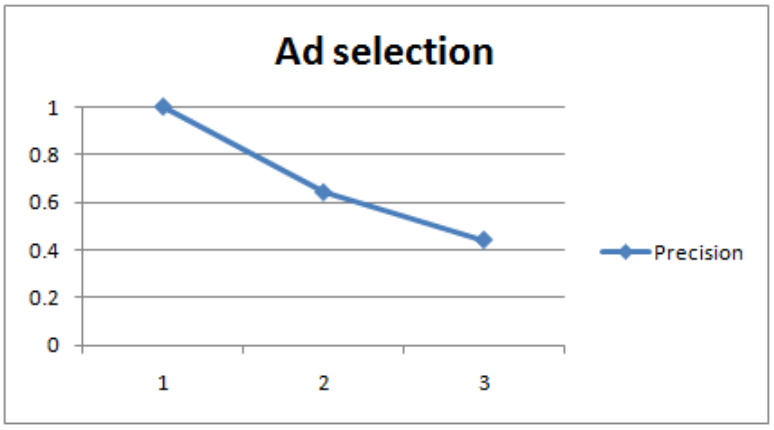

Figure 7.1: Precision for different acceptance levels

Evaluation of the advertisement placement

Similar to the evaluation of the semantic similarity matching, we assigned a score from 1 to 3 for each ad placement. An ad placement scoring 1 point is one that significantly alters the viewer experience (e.g. by occluding important elements of the scene), while an ad scoring 3 is an ad whose placement leads to a pleasant effect. This score ignores the specific ad which is actually shown and only takes into account the spatial position within the video. Table 7.2 below shows the average scores for the placements within each video.

Video	Score
Video 1	2.4
Video 2	2.4
Video 3	2.6
Video 4	2.2
Video 5	2.4
Average	2.4

Table 7.2: Average ad placement score per video

The average score over all the 25 ads is 2.4 (out of 3.0). The graph in Figure 7.2 below shows the evolution of precision, depending on what is considered to be the acceptance level. All the ad positions were scored 1 or above (since 1 is the minimum value), 88% were scored 2 or above and 52% scored 3.

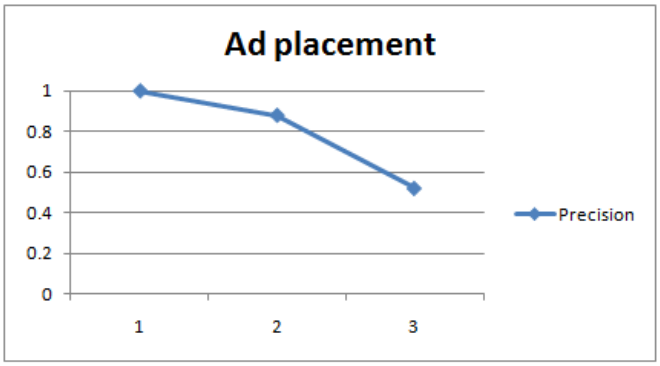

Figure 7.2: Precision for different acceptance levels

Evaluation of the overall results

Finally, we assigned an overall score from 1 to 5 (note the wider range) for each inserted advertisement. This is intended to take into account the overall effect that the advertisement has, considering both how related it is to the video content, and also how well it blends into the video.

Table 7.3 below shows the average values for each video, and also the average value for all the ads.

Video	Score
Video 1	3.8
Video 2	3.8
Video 3	3.4
Video 4	2.6
Video 5	3.0
Average	3.32

Table 7.3: Average overall score per video

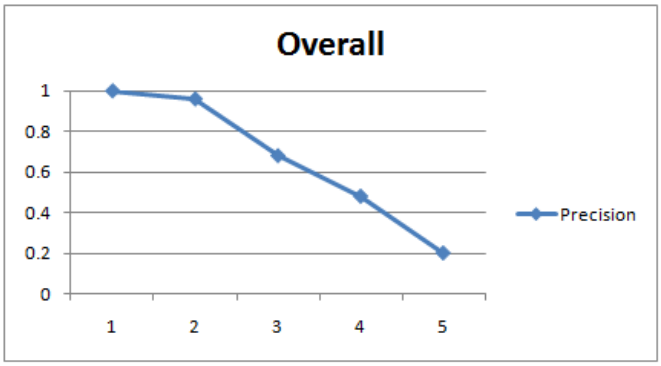

Figure 7.3: Precision for different acceptance levels

We illustrate below some ads that were well placed in the videos. In each of the cases, the ads avoid the important elements of the scene, and most of the times this could not have been achieved by traditional systems that insert ads at predefined locations.

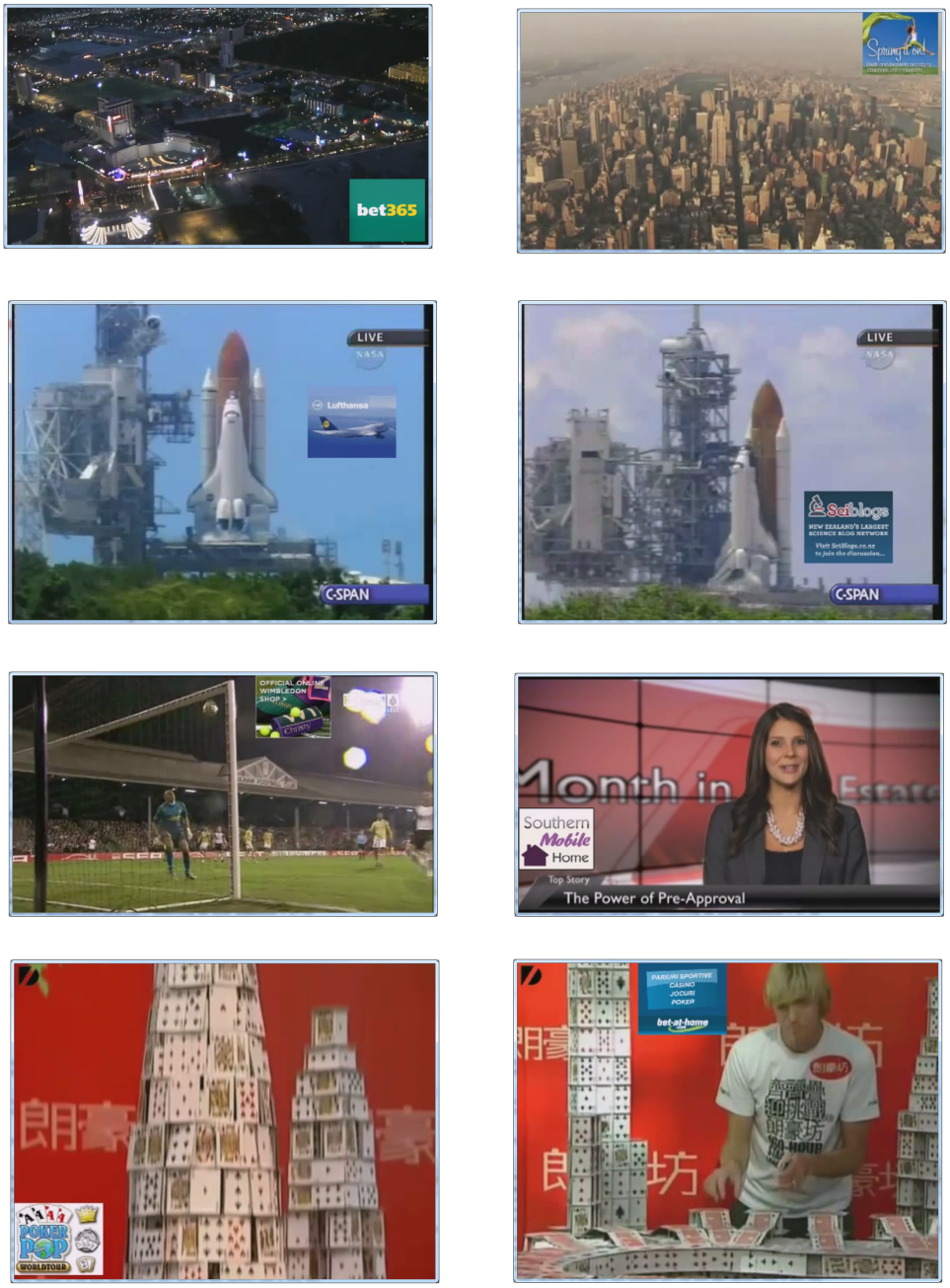

Figure 7.4: Sample ads

Chapter 8

Conclusions and Outlook

We described a system that places ads into videos in an innovative way. Two main problems were considered: finding the most suitable ads and finding optimal placements - both in space (where to insert the ad) and in time (when to insert the ad).

When considering which ad to place into a video we take two factors into account: first the semantic similarity and then the visual similarity, in order to make sure that the ad blends well into the video. For the semantic similarity, the problem was formulated as an information retrieval task, where the video represents a query for the ads database. However, we noted that our situation has a few particularities, most important of which being the scarcity of words available that describe each ad. In order to still be able to retrieve semantically related ads even when we do not have a perfect match between the video and ad keywords, we used a semantic similarity measure based on the Wordnet ontology. As mentioned, we also took into account the visual similarity between the video and the ad. This was done by extracting the color palette from both the video and the ad and defining a distance between the two.

When choosing the placement of the advertisement the main factor is given by video saliency analysis. A salient region is a region that is likely to attract a regular user's attention. The video saliency represents very useful information, since we would like not to occlude any of the important elements of the scene when showing an ad. We designed an efficient algorithm which allows estimating the total salience of a series of frames in $O(1)$ time, requiring a one-time preprocessing task to be done beforehand. A key advantage of the approach is that the preprocessing task may be done offline, being independent of the ad size. The cost of each possible placement is also refined by a series of other heuristics, such as to avoid having ads that pass scene boundaries, or inducing a preference towards having ads in interesting scenes, to make sure that the audience is involved.

Some improvements or alternative directions include working with different advertisement presentation techniques, such as ads that change their coordinates in such a way that they avoid the important objects, as opposed staying at a fixed position inside the player. A possible approach in this direction would be to penalize rapid movements or direction changes in order to prevent a random noisy behavior. However, a principal issue with such an approach is that the algorithm which allowed rapid computations would not be applicable anymore. Another direction for improvement would be to enrich the semantic data around the video or the ad by using machine vision techniques, such as object detection. This represents a related, active field of research and any advancement in that domain would potentially provide useful information for an intelligent advertising system.

Bibliography

[1] L. Itti; C. Koch, E. Niebur *A model of saliency-based visual attention for rapid scene analysis*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 20, no. 11, pp.1254-1259, Nov 1998

[2] L. Itti, C. Koch, *A comparison of feature combination strategies for saliency-based visual attention systems*. In Human Vision and Electronic Imaging. 1999

[3] T. Judd; K. Ehinger, F. Durand, A. Torralba. *Learning to predict where humans look*. Computer Vision, 2009 IEEE 12th International Conference on , vol., no., pp.2106-2113, Sept. 29 2009-Oct. 2 2009

[4] L. Chen, X. Xie, X. Fan, W. Ma, H. Zhang, H. Zhou. *A Visual attention model for adapting images on small displays*, MSR-TR-2002-125, Microsoft Research, Redmond, Washington. 2002

[5] Zhang Longfei, Cao Yuanda, Ding Gangyi, Wang Yong. *A Computable Visual Attention Model for Video Skimming*, Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on , vol., no., pp.667-672, 15-17 Dec. 2008

[6] Zeng Zhi Yong, Liu Shi Gang. *A Novel Region-Based Image Retrieval Algorithm Using Hybrid Feature*. Computer Science and Information Engineering, 2009 WRI World Congress on , vol.6, no., pp.416-420, March 31 2009-April 2 2009

[7] L. Itti, P. Baldi. *A principled approach to detecting surprising events in video*. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on , vol.1, no., pp. 631- 637 vol. 1, 20-25 June 2005

[8] Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. *A user attention model for video summarization*. In Proceedings of the tenth ACM international conference on Multimedia (MULTIMEDIA '02). ACM, New York, NY, USA, 533-542. 2002.

[9] Wolf Heiko, Deng Da. *Image saliency mapping and ranking using an extensible visual attention model based on MPEG-7 feature descriptors*. Discussion Paper 2005/10. Department of Information Science, University of Otago, Dunedin, New Zealand. 2005.

[10] A. Bernardo, Huberman ,Fang Wu. *The economics of attention: maximizing user value in information-rich environments*. In Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising (ADKDD '07). ACM, New York, NY, USA, 16-20. 2007.

[11] Jinlian Guo, Tao Mei, Falin Liu, Xian-Sheng Hua. 2009. *AdOn: an intelligent overlay video advertising system*. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09).

[12] Wei-Shing Liao, Kuan-Ting Chen, and Winston H. Hsu. *AdImage: video advertising by image matching and ad scheduling optimization.* In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08). 2008.

[13] Tao Mei, Xian-Sheng Hua, and Shipeng Li. *Contextual in-image advertising. In Proceeding of the 16th ACM international conference on Multimedia (MM '08).* ACM, New York, NY, USA, 439-448. 2008.

[14] Yuqiang Chen, Ou Jin, Gui-Rong Xue, Jia Chen, Qiang Yang. *Visual Contextual Advertising: Bringing Textual Advertisements to Images.* Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010

[15] Xin-Jing Wang, Mo Yu, Lei Zhang, Wei-Ying Ma. *Advertising based on users' photos.* Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on , vol., no., pp.1640-1643, June 28 2009-July 3 2009

[16] Wen-tau Yih, Joshua Goodman, Vitor R. Carvalho. *Finding advertising keywords on web pages.* In Proceedings of the 15th international conference on World Wide Web (WWW '06). ACM, New York, NY, USA, 213-222. 2006.

[17] Krzysztof Dembczynski, Wojciech Kotlowski. *Predicting Ads' Click-Through Rate with Decision Rules.* WWW. 2008.

[18] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, Mohamed Aly. *Video suggestion and discovery for youtube: taking random walks through the view graph.* In Proceeding of the 17th international conference on World Wide Web (WWW '08).

[19] Benoit Baccot, Omar Choudary, Romulus Grigoras, Vincent Charvillat. *On the impact of sequence and time in rich media advertising.* In Proceedings of the seventeen ACM international conference on Multimedia (MM '09).

[20] Sebe, Lew, Smeulders. *Computer Vision and Image Understanding.* Computer Vision and Image Understanding. Vol. 92, no. 2-3, pp. 141-146. 2003.

[21] Marcus J. Pickering, Lawrence Wong, Stefan M. RĂŒger. *ANSES: Summarisation of News Video.* Lecture Notes in Computer Science, 2003.

[22] Ming Lin; J.F. Nunamaker, M. Chau, Hsinchun Chen. *Segmentation of lecture videos based on text: a method combining multiple linguistic features.* Proceedings of the 37th Annual Hawaii International Conference on , vol., no., pp. 9 pp., 5-8 Jan. 2004.

[23] Chan Yun Yoo, Kihan Kim, Patricia A. Stout. *Assessing the Effects of Animation in Online Banner Advertising: Hierarchy of Effects Model.* Journal of Interactive Advertising. 2004.

[24] Jyun-Cheng Wang, Rong-Fuh Day. *The effects of attention inertia on advertisements on the WWW.* Computers in Human Behavior, May 2007.

[25] Ruth Rosenholtz, Nathaniel R. Twarog, Nadja Schinkel-Bielefeld, Martin Wattenberg. *An intuitive model of perceptual grouping for HCI design.* In Proceedings of the 27th international conference on Human factors in computing systems (CHI '09).

[26] K. Wang, T.G Wang, C. Farn. *Media Strategy vs. Content Strategy in Online Advertising: Exploring the Influence of Consumers' Goal-Directedness for Web Navigation.* PACIS 2007 Proceedings.

[27] *Digital Video Ad Format Guidelines and Best Practices.* http://www.iab.net/media/file/IAB-Video-Ad-Format-Standards.pdf

[28] *Digital Video In-Stream Ad Metrics Definitions.* http://www.iab.net/media/file/DV_In-Stream_Metrics_Definitions.pdf

[29] *Digital Video Ad Serving Template.* http://www.iab.net/media/file/VAST-2_0-FINAL.pdf

[30] *Digital Video Player-Ad Interface Definition.* http://www.iab.net/media/file/VPAIDFINAL51109.pdf

[31] Chenlei Guo, Qi Ma, Liming Zhang. *Spatio-temporal Saliency detection using phase spectrum of quaternion fourier transform.* In Proceedings of CVPR'2008.

[32] Sangwine, S.J.; Ell, T.A. *Hypercomplex Fourier transforms of color images.* In Proceedings of International Conference on Image Processing, 2001

[33] G.A. Miller. *WordNet: A Lexical Database for English.* Communications of the ACM, 1995.

[34] P. Resnik. *Using Information Content to Evaluate Semantic Similarity in a Taxonomy.* In Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995.

[35] *US Online Advertising Spending to Surpass Print in 2012.* http://www.emarketer.com/PressRelease.aspx?R=1008788

[36] *comScore Releases February 2012 U.S. Online Video Rankings.* http://www.comscore.com/Press_Events/Press_Releases/2012/3/

[37] H.Beales. *The value of behavioral targeting.* Network Advertising Initiative, 2010.

[38] H. Li, S.M. Edwards, J. Lee. *Measuring the intrusiveness of advertisements: Scale development and validation.* Journal of Advertising, 2002.

[39] *Unobtrusive ads can boost revenue.* http://adsense.blogspot.ro/2006/05/unobtrusive-ads-can-boost-revenue.html