SPY Project

Contribution to Deliverable xxx

Visual Descriptors for Video Indexing

- A State of the Art -

Département ARTEMIS

UMR CNRS 8145 MAP5

Author : Titus ZAHARIA

January 2012

Contents

1. Introduction

This document presents a brief overview of existing SoA techniques used in the fields of various video indexing and object recognition applications of potential utility for video-surveillance objectives. The first part of this document includes a brief recall of the existing ISO/MPEG-7 technologies (Section 2). Then, section 3 presents a different description approach, based on interest point descriptors and currently highly popular in various computer vision and object/event recognition applications. Finally, section 4 considers an emerging topic of research, which concerns the issue of detecting multiple instances of objects of interest in videos.

2. MPEG-7 descriptions of video documents

2. The MPEG-7 approach

The MPEG-7 standard offers support for a complete set of media types, including visual, audio, text and 3D data and aims at covering a wide field of applications, such as video search and retrieval, broadcasting and video on demand applications, but also more specialized ones such as video-surveillance.
MPEG-7 standardizes the following tools:

- A set of descriptors (Ds) corresponding to low-level descriptions related to features such as color, shape, texture and motion in the visual case.
- A set of description schemes (DSs), corresponding to higher level descriptions and combining other descriptors and/or description schemes, fully or partially instantiated.
- A description definition language (DDL), for expressing the individual Ds and DSs and creating interpretable and interexchangeable description documents of rich multimedia content. Let us already mention that the DDL adopted by MPEG-7 is basically the XML Schema language, with some minor specific extensions.
- System tools, defining description multiplexing, transmission, and synchronization mechanisms, coded representations (both textual and binary formats) for efficient storage and transmission, tools for management and protection of intellectual property in MPEG-7 descriptions, etc.

MPEG-7 aims at being a general-purpose description standard and thus defines a large variety of Ds and DSs, including low-level visual and audio descriptors, structural and semantic content description, abstract concepts, content management and production process, information about storage media and intellectual property, and so forth. Implementing and managing the complete set of Ds and DSs adopted by the standard within an indexing system would require a huge amount of software and hardware resources. However, specific applications do not require the totality of tools included in the standard.

2.1. MPEG-7 structural description of the AV content

The structural descriptions of the AV content play a central role within the MPEG-7 MDS (*Multimedia Description Schemes*) part of the standard [1], aiming at providing generic and flexible mechanism for structuring and representing the AV content. A particular attention has been paid to the management of complex video materials, which generally involve a huge amount of rich and heterogeneous information, corresponding to various media types. Defining appropriate multimodal structural DSs becomes then mandatory for efficiently accessing such an important volume of information.

MPEG-7 adopted a generic mechanism, able to manage multiple data decompositions within an unified manner and consisting of:

1. Defining an abstract class, Segment DS which specifies the generic elements that are common to all types of segments, such as ids, media locators, textual annotations, temporal references and generic subdivision mechanism.

2. Creating media specific segments, defined by inheritance from the Segment DS abstract class and integrating the low level descriptors appropriate to each media type. Concretely, the most popular media specific segments considered are the folowing: VideoSegment DS, AudioSegment DS, StillRegion DS, StillRegion3D DS, MovingRegion DS, VideoTextDS and AudioVisualSegment DS.

3. The definition of a generic mechanism for decomposing segments into sub-segments at the Segment DS level from which specific decomposition DSs corresponding to each type of segment are derived by inheritance, offers the premises of recursively creating hierarchical descriptions with multiple trees. Four decomposition types are currently supported:

- temporal (*e.g*. segmenting videos into scenes and shots), spatial (decomposition of a spatial region into subregions),
- spatiotemporal (decomposition of a video segment into moving regions, corresponding to foreground and background
- objects) and media (*e.g.* decomposing an audio-visual segment into an audio segment and a video segment).

The genericity of the segment-based approach offers an ideal framework for dealing with multiple and hierarchical table of contents, performing cross-modal queries and creating multigranular descriptions of the AV data (a segment may be represented by its own Ds or by the reunion of its subsegments Ds).

A video indexing web platform implementing the MPEG-7 structural description approach is proposed and described in [7].

As the low-level, media-dependent Ds include essential information for content-based retrieval, let us summarize now the MPEG-7 low-level visual tools.

2.2. MPEG-7 low-level visual descriptors

The MPEG-7 visual description tools [2] consist of basic structures (such as coordinate system, grid layout, time series, temporal interpolation mechanisms, multiview DS for 3D object characterization from multiple 2D projections) and descriptors that cover the following basic visual features: color,

texture, shape, motion, localization, and others. Each category consists of elementary and sophisticated descriptors. Table 1 presents the standardized visual descriptors.

Color	**Texture**
Color space	Edge orientation histogram
Dominant colors	Homogeneous texture (Gabor filters energy responses)
Color structure (histogram of stuctural elements)	Texture browsing (Tamura-like features)
GoF/GoP color (mean, median or min histogram for a group of video frames/pictures)	**Shape**
Color layout (DCT-based coded layout)	Region Shape (ART- Angular Radial Transform)
Scalable color histogram (Haar-transformed histogram)	Contour Shape (contour scale space)
Motion	3D-shape (3D shape spectrum)
Parametric motion	**Localization**
Motion trajectory	Region locator (polygonal region)
Camera motion (complete 3D camera modeling)	Spatio-temporal locator (set of polygonal regions)
Motion activity	**Others**
	Face recognition (eigenfaces)

Table 1. The MPEG-7 visual descriptors.

For a detailed description of MPEG-7 visual descriptors, with definition, representation, properties and associated similarity measures, the reader is invited to refer to [3], [4], [6].

In the following section, we propose a brief recall of some essential MPEG-7 descriptors and notably the 2D shape representations as well as the dominant color descriptor, which are useful for object-based identification and detection purposes.

2.3. MPEG-7 Shape Descriptors

Let us start with the Contour Shape (CS) descriptor [5] proposed by the MPEG-7 standard [2], [3], [4]. In order to obtain the CS descriptor, the first step is to extract the contour of the 2D shape. Further, this contour is successively filtered with a Gaussian kernel. Thus, a set of several contours in a Gaussian scale space are obtained. For each of them, the curvilinear positions of the inflexion points are computed. Finally, for each curve, the inflexion points are represented in the (σ, u) space (the Contour Scale Space - CSS), where σ represents the standard deviation used to generate the curve and u represents the curvilinear position of the considered inflexion point. Once the CSS

representation is obtained, the curvature peaks are determined. For each peak, the corresponding curvature value and position (expressed in curvilinear abscise) are retained as the CS descriptor. The associated similarity measure between two CSS representations is based on a matching procedure which takes into account the cost of fitted and unfitted curvatures peaks [4].

Let us note that the MPEG-7 Contour Shape descriptor offers the advantage of being intrinsically invariant to object's position and pose.

The second approach adopted is the MPEG-7 Region Shape (RS) descriptor, based on the 2D Angular Radial Transform (ART). In this case, the object's support function is represented as a weighted sum of 34 ART basis functions ($f_{m,n}$).

$$f_{m,n}(\rho,\theta M) = \frac{1}{\pi}\cos(\pi n\rho)e^{jm\theta}. \quad (1)$$

The decomposition coefficients constitute the descriptor. The distance between two shapes is simply defined as the L1 distance between the absolute values of the ART coefficients.

The MPEG-7 region shape descriptor requires preliminary object normalization into the unit sphere.

Let us also note that the MPEG-7 shape descriptors can also be exploited for performing 2D/3D object indexing, retrieval, as well as semantic categorization by exploiting the MPEG-7 Multiview DS structure or with the help of the methods such as those proposed in [9], [11].

2.4. MPEG-7 Dominant Color Descriptor

The MPEG-7 dominant color descriptor exploits a (over-)segmentation of the image, which can be achieved with arbitrary existing algorithms. Each region (or segment) determined is described by a unique, homogeneous color, defined as the mean value of the pixels of the given region. The set of colors, together with their percentage of occupation in the image (*i.e.*, the associated color histogram) are regrouped into a visual representation. More precisely, let $C_I = \{c_1^I, c_2^I, \ldots c_{N_I}^I\}$ be the set of N_I colors obtained for image I, and $H_I = (p_1^I, p_2^I, \ldots p_{N_I}^I)$ the associated color histogram vector. The visual image representation is defined as the couple (C_I, H_I). Let us note that some more sophisticated DCD-based approaches [1], [13], has also been introduced recently.

The query is by definition an object of arbitrary shape and is processed in the same manner in order to derive its visual representation.

The advantage of the DCD representation comes from the fact that objects with arbitrary numbers of colors can be efficiently compared by using, for example, the Quadratic Form Distance Measure, which can be re-written for arbitrary length representations as described by the following equation:

c_i^Q, c_j
(2)

where $H_Q = \left(p_1^Q, p_2^Q, \ldots p_{N_Q}^Q\right)$ and $H_I = \left(p_1^I, p_2^I, \ldots p_{N_I}^I\right)$ respectively denote the DCD histogram vectors of length N_Q, and N_I respectively associated to the query (*Q*) and candidate (*I*) images. The function *a*, describe the similarity between two colors c_i and c_j and is defined as:

$$a(c_i, c_j) = 1 - \frac{d(c_i, c_j)}{d_{max}} \quad (3)$$

where *d* is the Euclidean distance between colors c_i and c_j and d_{max} is the maximum Euclidean distance between any 2 colors in the considered color space (*e.g.*, for the RGB color space $d_{max} \cong 442$).

Let us note that each color region in a candidate image has a specific contribution to the global distance. Thus, the contribution of color c_j^I in an image *I* to the global distance between image *I* and query *Q* is defined as:

$$C\left(c_j^I, Q\right) = \sum_{i=1}^{N_I} a\left(c_j^I, c_i^I\right) p_j^I p_i^I - \sum_{i=1}^{N_Q} a\left(c_i^Q, c_j^I\right) p_i^Q p_j^I \quad (4)$$

The above-defined distance is used as a global criterion in the matching stage. In the case of object identification applications, the objective is to determine, in each key-frame of the considered video sequence, candidate regions visually similar with the query. Some solutions are introduced in [7], [10].

3. Interest point Descriptors

In the recent, years, among the most popular in the field of computer vision, image/video indexing, video-surveillance, object/event recognition are the interest point descriptors.

3.1 SIFT (Scale Invariant Feature Transform)

The SIFT descriptor proposed by Lowe [1] describes the local shape of a region using edge orientation histograms. The gradient of an image is shift-invariant: taking the derivative cancels out offsets. Under light intensity changes, *i.e.* a scaling of the intensity channel, the gradient direction and the relative gradient magnitude remain the same. Because the SIFT descriptor is normalized, the gradient magnitude changes have no effect on the final descriptor. The SIFT descriptor is not invariant to light color changes because the intensity channel is a combination of the R, G, and B channels.

SIFT descriptors are attached to the keypoints detected by using the Diffference of Gaussians method proposed in [1]. In order to better localize the keypoints a method for fitting 3D quadratic function to the local sample points to determine the interpolated location of the maximum is used [15]. Once the keypoints are localized, the unstable extrema with low contrast are rejected by using the Taylor expansion of the scale-space function.

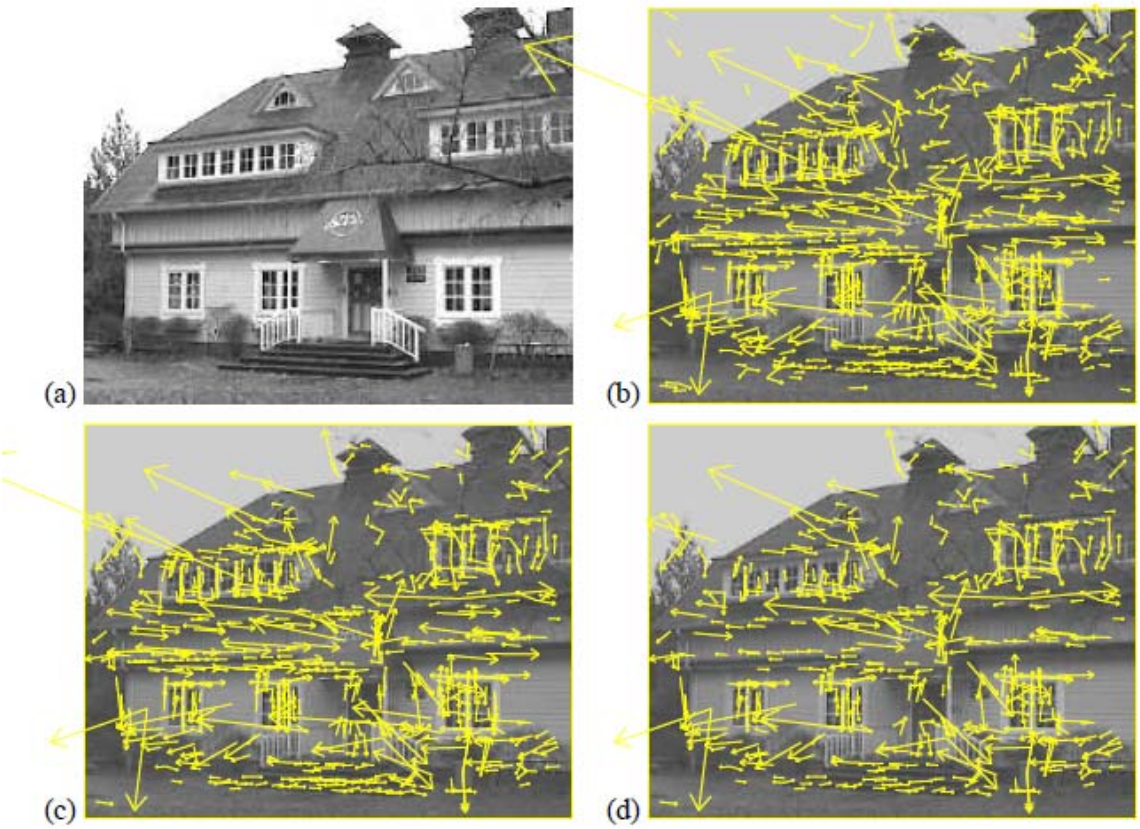

Figure 1. Keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

Figure 1 shows the effects of keypoint selection on a natural image. In order to avoid too much clutter, a low-resolution 233 by 189 pixel image is used and keypoints are shown as vectors giving the location, scale, and orientation of each keypoint (orientation assignment is described below). Figure 1 (a) shows the original image, which is shown at reduced contrast behind the subsequent figures. Figure 1 (b) presents the 832 keypoints at all detected maxima and minima of the difference-of-Gaussian function, while Figure 1.(c) illustrates the 729 keypoints that remain following removal of those with low contrast. In order to provide a higher stability, edge responses are removed by using a Hessian matrix Figure 4.(d)

By assigning a consistent orientation to each keypoint based on local image properties, the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation. The orientation is computed from the dominant directions of local gradients.

Figure 2 illustrates the computation of the keypoint descriptor. First the image gradient magnitudes and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation.

A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point. This is illustrated with a circular window on the left side of Figure 2. The purpose of this Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are far from the center of the descriptor, as these are most affected by miss-registration errors.

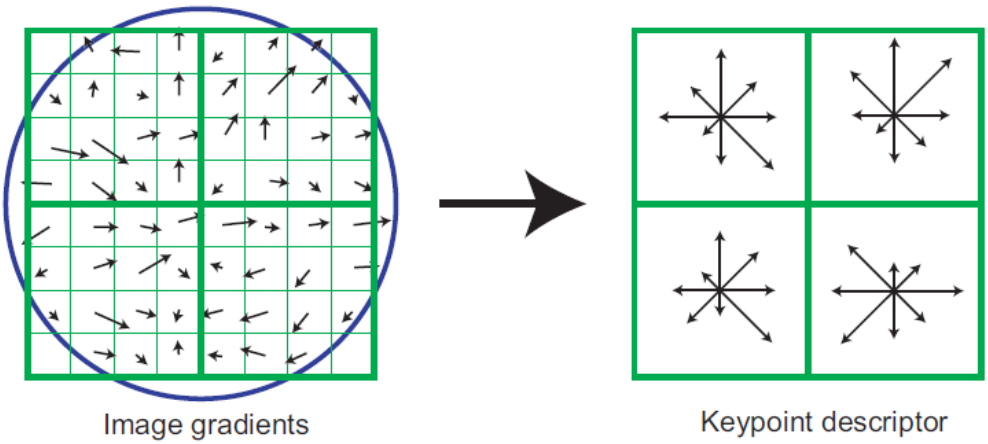

Figure 2. A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the best results are obtained when using 4x4 descriptors computed from a 16x16 sample array.

The keypoint descriptor is illustrated on the right side of Figure 2. It allows for significant shift in gradient positions by creating orientation histograms over 4x4 sample regions. The figure shows eight directions for each orientation histogram, with the length of each arrow corresponding to the magnitude of that histogram entry. A gradient sample on the left can shift up to 4 sample positions while still contributing to the same histogram on the right, thereby achieving the objective of allowing for larger local positional shifts.

It is important to avoid all boundary affects in which the descriptor abruptly changes as a sample shifts smoothly from being within one histogram to another or from one orientation to another. Therefore, trilinear interpolation is used to distribute the value of each gradient sample into adjacent histogram bins. In other words, each entry into a bin is multiplied by a weight of $1 - d$ for each dimension, where d is the distance of the sample from the central value of the bin as measured in units of the histogram bin spacing.

The descriptor is formed from a vector containing the values of all the orientation histogram entries, corresponding to the lengths of the arrows on the right side of Figure 2. The figure shows a 2x2 array of orientation histograms, whereas our experiments below show that the best results are achieved with a 4x4 array of histograms with 8 orientation bins in each. Therefore the feature vector for each keypoint will consist of 4x4x8 = 128 elements.

3.2 Color SIFT Descriptors

3.2.1 HSV-SIFT

Bosch *et al.* [16] compute SIFT descriptors over all three channels of the HSV color model. This gives 3 x 128 dimensions per descriptor, 128 per channel. The H color model is scale-invariant and shift-invariant with respect to light intensity. However, due to the combination of the HSV channels, the complete descriptor has no invariance properties.

3.2.2 HueSIFT

Van de Weijer *et al*. [18] introduce a concatenation of the hue histogram with the SIFT descriptor. When compared to HSV-SIFT, the usage of the weighed hue histogram addresses the instability of the hue near the gray axis. Similar to the hue histogram, the HueSIFT descriptor is scale-invariant and shift-invariant.

3.2.3 OpponentSIFT

OpponentSIFT describes all of the channels in the opponent color space using SIFT descriptors. The information in the O_3 channel is equal to the intensity information, while the other channels describe the color information in the image. These other channels do contain some intensity information, but, due to the normalization of the SIFT descriptor, they are invariant to changes in light intensity.

3.2.4 C-SIFT

In the opponent color space, the O_1 and O_2 channels still contain some intensity information. To add invariance to intensity changes, [19] proposes the C-invariant, which eliminates the remaining intensity information from these channels. The use of color invariants as input for SIFT was first suggested by Abdel-Hakim and Farag [21]. The C-SIFT descriptor [20] uses the C-invariant, which can be intuitively seen as the normalized opponent color space $\frac{O_1}{O_3}$ and $\frac{O_2}{O_3}$. Because of the division by intensity, the scaling in the diagonal model will cancel out, making C-SIFT scale-invariant with respect to light intensity. Due to the definition of the color space, the offset does not cancel out when taking the derivative: It is not shift-invariant.

3.2.5 rgSIFT

For the rgSIFT descriptor, descriptors are added for the *r* and *g* chromaticity components of the normalized RGB color model, which is already scale-invariant.

3.2.6 Transformed color SIFT

The following color space transform is first applied :

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_R} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix} \quad (5)$$

with μ_C the mean and σ_C the standard deviation of the distribution in channel C computed over the area under consideration (*e.g.*, a patch or image). This yields, for every channel, a distribution where $\mu_C = 0$ and $\sigma_C = 1$.

For the transformed color SIFT, the same normalization is applied to the RGB channels as for the transformed color histogram. For every normalized channel, the SIFT descriptor is computed. The descriptor is scale-invariant, shift-invariant, and invariant to light color changes and shift.

3.3 SURF (Speeded-Up Robust Features)

SURF [17] detects interest points by using a basic Hessian-matrix approximation and the integral images, reducing computation time drastically. SURF describes the distribution of the intensity content within the interest point neighborhood, similar to the gradient information extracted by SIFT [1].

The SURF descriptor builds on the distribution of first order Haar wavelet responses in *x* and *y* direction rather than the gradient, exploits integral images for speed, and uses a 64-dimensional feature vector. This reduces the time for feature computation and matching, and has proven to simultaneously increase the robustness.

In order to ensure the invariance to image rotation, an orientation is computed for every interest point (Figure 3)

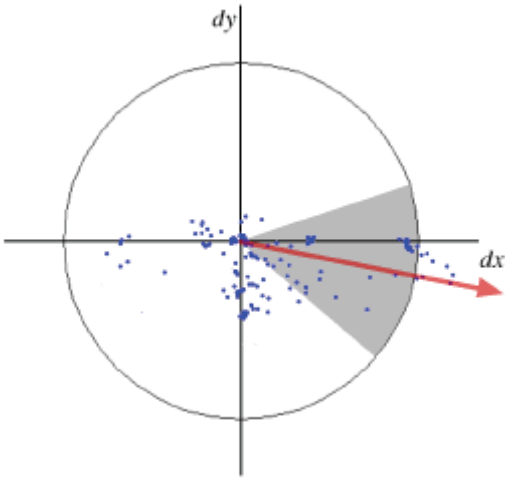

Figure 3. Orientation assignment: A sliding orientation window of size π/3 detects the dominant orientation of the Gaussian weighted Haar wavelet responses at every sample point within a circular neighborhood around the interest point.

For the extraction of the descriptor, a square region centered around the interest point and oriented along the orientation, is constructed. This region is split up regularly into smaller 4x4 square sub-regions. For each sub-region , Haar wavelet responses are computed at 5x5 regularly spaced sample points. The responses for the horizontal and vertical direction are summed up over each sub-region and form a set of entries in the feature vector (Figure 4). Thus, each sub-region has a four dimensional descriptor **v** for its underlying intensity structure

$$v = \left(\sum d_x , \sum d_y , \sum |d_x| , \sum |d_y| \right) \quad (6)$$

Concatenating this for all 4x4 sub-regions will result in a descriptor vector of length 64. The wavelet responses are invariant to a bias in illumination (offset). Invariance to contrast is achieved by turning the descriptor into a unit vector.

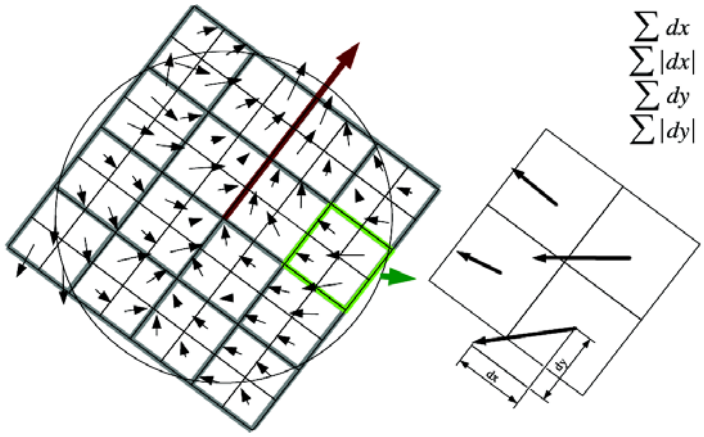

Figure 4. To build the descriptor, an oriented quadratic grid with 4x4 square sub-regions is laid over the interest point (left). For each square, the wavelet responses are computed. The 2x2 sub-divisions of each square correspond to the actual fields of the descriptor. These are the sums *dx*, |*dx*|, *dy* and |*dy*|, computed relatively to the orientation of the grid (right).

3.4 Keypoint matching techniques

3.43.1 SIFT

The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector.

An effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor. If there are multiple training images of the same object, then the second-closest neighbor is defined as being the closest neighbor that is known to come from a different object than the first, such as by only using images known to contain different objects. This measure performs well because correct matches need to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. For false matches, there will likely be a number of other false matches within similar distances due to the high dimensionality of the feature space. We can think of the second-closest match as providing an estimate of the density of false matches within this portion of the feature space and at the same time identifying specific instances of feature ambiguity. Such an approach will reject all matches in which the distance ratio is greater than 0.8, which eliminates 90% of the false matches while discarding less than 5% of the correct matches [1].

No algorithms are known that can identify the exact nearest neighbors of points in high dimensional spaces (128-dimensional feature vector) that are any more efficient than exhaustive search. Therefore, Beis and Lowe [22] propose an approximate algorithm, called the Best-Bin-First (BBF) algorithm. This is approximate in the sense that it returns the closest neighbor with high probability.

The BBF algorithm uses a modified search ordering for the k-d tree algorithm so that bins in feature space are searched in the order of their closest distance from the query location. This priority search order requires the use of a heap-based priority queue for efficient determination of the search order. An approximate answer can be returned with low cost by cutting off further search after a specific number of the nearest bins have been explored. Lowe *et. al* cut off search after checking the first 200 nearest-neighbor candidates. For a database of 100,000 keypoints, this provides a speedup over exact nearest neighbor search by about 2 orders of magnitude yet results in less than a 5%loss in the number of correct matches.

6.1.2 SURF

For fast indexing during the matching stage, the sign of the Laplacian (*i.e*. the trace of the Hessian matrix) for the underlying interest point is included. Typically, the interest points are found at blob-type structures. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. This feature is available at no extra computational cost as it was already computed during the detection phase. In the matching stage, features are compared only if they have the same type of contrast (Figure 5). Hence, this minimal information allows for faster matching, without reducing the descriptor's performance. Note that this is also of advantage for more advanced indexing methods. In the case of k-d trees, this extra information defines a meaningful hyperplane for splitting the data, as opposed to randomly choosing an element or using feature statistics.

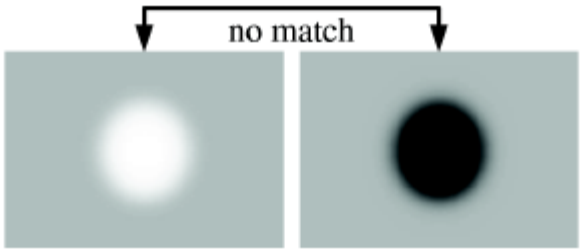

Figure 5. If the contrast between two interest points is different (dark on light background vs. light on dark background), the candidate is not considered a valuable match.

6.2 Words matching (adapted TF-IDF)

Considering the Bag-of-Words model described previously, some matching methods have adapted already existing text retrieval algorithms. The most popular is the Term Frequency – Inverse Document Frequency (TF-IDF) weighting scheme

6.2.1 TF-IDF weighting scheme

The TF-IDF weighting scheme can be considered as a statistical procedure. The main advantages of the TF-IDF method are its simplicity and efficiency, which explains its high popularity. The TF-IDF method also served as a starting point for some more algorithms, which propose some extensions/optimizations [23].

Essentially, TF-IDF works by determining the relative frequency of words in a specific document, normalized by the occurrences of the considered word within the entire document corpus. Intuitively, this measure determines how relevant a given word is in a particular document. Words that occur in a single or a small group of documents tend to have higher TF-IDF numbers than common words such as articles and prepositions [24].

The term frequency (TF) is by definition the number of occurrences of the considered term in the document. This amount is usually normalized to the total number of occurrences of terms of interest, in order to avoid bias related to the length of the document (the number of occurrences would be potentially higher in a page than in a paragraph).

For a document d_j and a term t_i the frequency of the term in the document is defined as:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{7}$$

where $n_{i,\ j}$ is the number of occurrences of the t_i term in d_j. The denominator is the number of occurrences of all terms in document d_j.

The *inverse document frequency* (IDF) is a measure of the importance of the term throughout the corpus. The IDF is obtained from the calculation of the logarithm of the inverse of the proportion of the corpus of documents containing the term:

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \quad \text{-(8)}$$

where $|D|$ is the total number of documents in the corpus and $|\{d_j : t_i \in d_j\}|$ is the number of documents where the t_i term appears ($n_{i,j} \neq 0$).

Finally, the TF-IDF weight is defined as the product of the two above-described measures:

$$tfidf_{i,\ j} = tf_{i,j} \cdot idf_i \quad (9)$$

Let us consider an example of corpus including 3 text documents as illustrated in Figure 6.

Document 1	Document 2	Document 3
His name is celebrated in the grove **that** trembles, and through the murmuring stream, winds prevail until the celestial arc, the arc of grace and consolation **that** his hand stretched into the clouds.	Just be distinguished two goals at the end of the career of one oak shading around the palm trees were to be seen in the glow of evening.	Ah! the weather of my poetry! the beautiful days I spent with you! The former, infinite joy, peace and freedom, the latter marked by a melancholy **that** also had its many charms.

Figure 6. Example corpus taken from the works of Friedrich Gottlieb Klopstock

The example focuses on document 1 (or $d_{1)}$ and the analyzed term is "that" *(t_1* = that). We will ignore the punctuation and apostrophes.

We will perform the following calculations:

$$tf_{1,1} = \frac{n_{1,1}}{\sum_k n_{k,1}} = \frac{2}{33}$$

We can notice that most terms appear once (20 words), *his*, *and*, *that* and *arc* appear 2 times and *the* appears 5 times. The denominator is 5 + 4 * 2 + 20. This sum is the number of words in the document.

The term "that" does not appear in the second document, thus:

$$idf_1 = \log \frac{|D|}{|\{d_j : t_1 \in d_j\}|} = \log \frac{3}{2}$$

We will obtain:

$$tfidf_{1,1} = \frac{2}{33} \cdot \log \frac{3}{2} \approx 0.1059$$

For other documents:

$$tfidf_{1,2} = 0 \cdot \log \frac{3}{2} = 0$$

$$tfidf_{1,3} = \frac{1}{32} \cdot \log \frac{3}{2} \approx 0.0546$$

The first document appears as "most relevant".

6.2.2 TF-IDF and multimedia content retrieval

In video retrieval, each key frame features a set of “visual words”, just like a text corpus has a set of words it contains. We adapt the *tf-idf* scheme and create array of visual words for each key frame and use a cosine similarity measure for retrieving the best match.

Cosine similarity is a measure of similarity between two vectors of *n* dimensions by finding the cosine of the angle between them. Given two vectors of attributes, *A* and *B*, the cosine similarity, θ, is represented using a dot product and magnitude as:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \tag{10}$$

The cosine similarity will range from 0 to 1, since the tf-idf weights cannot be cannot be negative.

The tf-idf scheme offers a simple yet efficient algorithm for matching textual querries with concepts included in the description. However, the tf-idf has also some limitations. In terms of synonyms, tf-idf does not make the jump to the relationship between words. For example, if the user wanted to find information about, the word “stone”, tf-idf would not consider documents that might be relevant to the query but instead use the word “rock”. For large video collections, this could present a serious problem.

In [25], Sivic et al.adopted *tf-idf*, while most of the other works chose *tf* directly [26], [27]. In [28], binary weighting, which indicates the presence and absence of a visual word with values 1 and 0 respectively, was used. Generally speaking, all the weighting schemes perform the nearest neighbor search in the vocabulary in the sense that each keypoint is mapped to the most similar visual word (i.e., the nearest cluster centroid).

For visual words, directly assigning a keypoint to its nearest neighbor is not an optimal choice, given the fact that two similar points may be clustered into different clusters when increasing the size of

visual vocabulary. On the other hand, simply counting the votes (e.g. *tf*) is not optimal as well. For instance, two keypoints assigned to the same visual word are not necessarily equally similar to that visual word, meaning that their distances to the cluster centre are different. Ignoring their similarity with the visual word during weight assignment causes the contributions of two keypoints equal, and thus more difficult to assess the importance of a visual word in an image.

In order to tackle the aforementioned problems, in [29], Agarwal and Triggs proposed to fit a probabilistic mixture model to the distribution of a set of training local features in the descriptor space, and code new features by their vectors of posterior mixture-component membership probabilities. This method, although interesting, involves a training stage which is not very efficient.

Jiang *et. al* [30] propose an straightforward *soft-weighting* approach to weight the significance of visual words. For each keypoint in an image, instead of searching only for the nearest visual word, they select the top-N nearest visual words. Suppose we have a visual vocabulary of *K* visual words, we use a *K*-dimensional vector $T = [t_1,\ldots, t_k, \ldots,t_K]$ with each component t_k representing the weight of a visual word *k* in an image such that

$$t_k = \sum_{i=1}^{N} \sum_{j=1}^{M_i} \frac{1}{2^{i-1}} sim(j,k) \qquad (11)$$

where M_i represents the number of keypoints whose i^{th} nearest neighbor is visual word *k*. The measure *sim(j, k)* represents the similarity between keypoint *j* and visual word *k*. Notice that in the equation above, the contribution of a keypoint is dependent on its similarity to word *k* weighted by $\frac{1}{2^{i-1}}$, representing the word is its i^{th} nearest neighbor. Empirically it has been found that $N = 4$ is a reasonable setting.

We can notice that *tf-idf* provides a good starting point for visual words matching and there are many other different approaches improving it.

4. Multiple video object detection

While an increasing number of solutions have provided a variety of satisfying results for concept detection in videos **[31]**, retrieving different instances of the same object in video sequences still remains a challenge. The main difficulty is related to the specification of semi-global image representation that need to be considered, together with the elaboration of efficient partial matching strategies. In addition, variations in visual appearance and object's pose have to be taken into account appropriately. This relatively recent topic of research has been considered in the TRECVID 2010 and 2011 evaluation campaign, under the so-called instance search task, and TRECVID work is currently ongoing for the 2012 edition.

Related work includes two types of approaches, including (possibly dense) interest points as well as local regions.

Currently, interest points are among the most popular tools for object recognition and classification for both images and videos.

Early approaches for object retrieval, using interest points, have been developed by Sivic and Zisserman in their Video Google system **[33]**. In this case, SIFT descriptors **[34]** are extracted from video keyframes with the help of two types of overlapping image patches: Harris-Affine **[35]** regions, based on interest point neighborhoods, and so-called Maximally Stable Extreme Regions (MSER) **[36]**. The *bag-of-words* technique is used for achieving fast and efficient retrieval of objects interactively selected by the user with the help of a bounding box. Other applications involving interest points include scene classification and image understanding (*e.g.* **[37]**, **[38]**). Interest points yield a high repeatability, *i.e*. they can be extracted reliably and are often identified in other images where the same object/scene appears.

However, the number of interest points extracted from an image varies a lot with the image content (from a few hundred to several thousands).

Starting from the method proposed in **[39]**, Li *et al*. **[40]** group points of interest in graphs by using Delaunay triangulations. They take in consideration different geometric constraints with the goal of characterizing the geometric properties of the neighborhood of each node. Moreover each node has to be represented as an "affine" combination of its neighboring nodes. The obtained model is then matched to different scenes in order to determine the object of interest.

Aiming to improve the accuracy of the process by injecting more spatial localization information in the visual representation, a different scheme, based on a dense sampling of the image with a regular grid (possibly defined over a range of scales) is proposed in **[41]**, **[42]**, **[43]**. Such approaches prove to be particularly useful for stereo matching [12]. On the downside, dense sampling cannot reach the same level of repeatability as obtained with interest points, unless sampling is performed extremely densely, in which case the number of features becomes unacceptably large.

In order to combine the advantages of both schemes, Tuytelaars **[44]** has recently introduced the dense interest points, starting from densely sampled image patches and then applying for each feature a local optimization of the position and scale within a bounded search area. The outcome of this process is a set of interest points on a semi-regular grid, densely covering the entire image as is the case with dense sampling, but with repeatability properties closer to those of standard interest points.

Browne and Smeaton **[45]** propose a different approach. In order to perform character retrieval in animated videos, they use a number of templates of each object to be detected and a matching procedure to compare each image against the available templates. In this case, templates are represented by all the yellow parts of the faces of the cartoon characters from "The Simpsons" series.

In **[46]**, authors generate for each keyframe a hierarchy of regions represented by a Binary Partition Tree. Various visual descriptors are extracted from each region and used to create visual codebooks. Another region-based approach is proposed in [17], where frames are divided in rectangular cells forming a grid and the descriptors of each cell are used. Histogram-based descriptors (*e.g*. HSV histogram, MPEG-7 Edge Histogram, Wavelet histogram) are here used in order to cluster the cells into a Bag of Features to be compared with a dictionary.

Gould *et al*. **[49]** propose to combine appearance-based features computed on superpixels **[48]** patches with relative location priors in a two stage classification process. Malisiewicz *et al*. **[50]** have shown that using image segmentation is efficient to improve the spatial support for object detection and recognition.

In **[51]**, the authors construct "region adjacency graphs" of pre-segmented objects and retrieve similar objects with the help of a new graph matching method based on an improvement of relaxation labeling techniques. In **[52]** image segments resulted from different segmentation algorithms are used as primitives to extract other features (*e.g.* color, texture and interest points) and for training detection models for a predefined set of categories.

Finally, let us mention the approach introduced in **[53]**. Here, a different, region-based representation is proposed. The idea is to represent the image as a dense map of (overlapping) regions.

The approach proposed in [7], [10] adopts a region-based representation strategy, which involves an over-segmentation of the image. Here, arbitrary segmentation methods can be considered, since the goal is to achieve independency with respect to the adopted segmentation procedure. The obtained representation is described with the help of an extended version of the MPEG-7 dominant color descriptor (DCD) **[54]**. Finally, two matching strategies, one based on a greedy strategy and the other on simulated annealing optimization, are proposed in order to retrieve similar objects of interest.

The advantage of region-based approaches comes from the possibility of directly exploiting the connectivity information (*i.e.* adjacency between regions), which can be highly useful in the matching stage.

5. References

[1] (ISO/IEC 2002) ISO/IEC 15938-3: 2002, MPEG-7-Visual, Information Technology – Multimedia content description interface – Part 3: Visual, 2002.

[2] (ISO/IEC 2003) ISO/ IEC 15938-5, Information technology - MultimediaContent Description. Interface - Part 5: Multimedia Description Schemes. 2003.

[3] B.S. Manjunath & Salembier, P.& Sikora, T. (2002). Introduction to MPEG-7: Multimedia Content Description Interface, John Wiley & Sons, Inc., New York, NY.

[4] Bober, M. (2002). MPEG-7 Visual Shape Descriptors, IEEE Transaction on Circuits and Systems for Video Technology, Volume 11, Issue 6, pp. 716-719.

[5] Mokhtarian, F.& Mackworth, A.K. (1992). A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 789-805.

[6] T. Zaharia, F. Prêteux, « Descripteurs visuels dans le standard MPEG-7 », dans A. Mostefaoui, F. Prêteux, V. Lecuire, J.-M. Moureaux (Ed.), Gestion des données multimédias, Traité IC2 – Série Informatique et Systèmes d'Information, Hermès-Lavoisier, Paris, France, pp. 225-282, Mai 2004.

[7] A. Bursuc, T. Zaharia, F. Prêteux; "Retrieval of Multiple Instances of Object in Videos," *Proc. 18th International Conference in Multimedia Modeling* (MMM2012), *Lecture Notes in Computer Science* (LNCS) 7131/7132, Klagenfurt, Austria, January 2012.

[8] A. Bursuc, T. Zaharia, F. Prêteux, "OVIDIUS: a Web Platform for Video Browsing and Search," *Proc. 18th International Conference in Multimedia Modeling* (MMM2012)/*Video Browser Showdown*, *Lecture Notes in Computer Science* (LNCS) 7131/7132, Klagenfurt, Austria, January 2012.

[9] R.D. Petre, T. Zaharia, "3D models-based semantic labeling of 2D objects", *International Conference on Digital Image Computing: Techniques and Applications* (IEEE eXplore), Noosa, QLD, Australia, December 2011.

[10] A. Bursuc, T. Zaharia, F. Prêteux, "Detection of Multiple Instances of Video Objects," *Proc. IEEE/ACM .International Conference on Signal Image Technology and Internet-based systems*, (SITIS 2011), Dijon, France, Novembre 2011.

[11] R. D. Petre, T. Zaharia, "3D Model-based Semantic Categorization of Still Image 2D Objects", International Journal of Multimedia Data Engineering and Management (IGI Global), Vol. 2, Issue 4, pp. 19-37, 2011.

[12] Yang, N.C., Chang, W.H., Kuo, C.M., Li, T.H.: "A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval", Journal of Visual Communication and Image Representation, Vol. 19, Issue 2, February 2008, pp. 92-105.

[13] Zin, T.T., Tin, P., Toriu, T., Hama, H.: "Dominant Color Embedded Markov Chain Model for Object Image Retrieval," 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009, pp.186-189, 12-14 Sept. 2009.

[14] D.G. Lowe, “Distinctive Image Features from Scale Invariant Keypoints,” International Journal on Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.
[15] M. Brown and D. Lowe. Invariant features from interest point groups. In BMVC, 2002.
[16] A. Bosch, A. Zisserman, and X. Munoz, “Scene Classification Using a Hybrid Generative/Discriminative Approach,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 4,pp. 712-727, Apr. 2008.
[17] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In ECCV, 2006.
[18] J. van de Weijer, T. Gevers, and A. Bagdanov, “Boosting Color Saliency in Image Feature Detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pp. 150-156, Jan. 2006.
[19] J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and H. Geerts, “Color Invariance,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 12, pp. 1338-1350, Dec. 2001.
[20] G.J. Burghouts and J.M. Geusebroek, “Performance Evaluation of Local Color Invariants,” Computer Vision and Image Understanding, vol. 113, pp. 48-62, 2009.
[21] A.E. Abdel-Hakim and A.A. Farag, “CSIFT: A SIFT Descriptor with Color Invariant Characteristics,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1978-1983, 2006.
[22] Beis, J. and Lowe, D.G. 1997. Shape indexing using approximate nearest-neighbor search in high-dimensional spaces. In Conference on Computer Vision and Pattern Recognition, Puerto Rico, pp. 1000-1006.
[23] Salton and Buckley, Term-weighting approaches in automatic text retrieval. Information Proceeding and Management, p.513-523, 1988
[24] Juan Ramos, “Using TF-IDF to determine word relevance in document queries”, http://www.cs.rutgers.edu/~mlittman/courses/ml03/iCML03/papers/ramos.pdf , last accessed August 2009
[25] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In ICCV, 2003.
[26] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: An in-depth study. In INRIA Technical Report RR-5737, 2005.
[27] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In IEEE CVPR, 2006.
[28] E. Nowak et al. Sampling strategies for bag-of-features image classification. In ECCV, 2006.
[29] A. Agarwal, and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In ECCV, 2006.
[30] Y.-G. Jiang, C.-W. Ngo, and J. Yang, “Towards optimal bag-of-features for object categorization and semantic video retrieval,” in Proceedings of the ACM International Conference on Image and Video Retrieval, pp. 494–501, Amsterdam, The Netherlands, 2007.
[31] Snoek, C.G.M., Worring, M.: "Concept-Based Video Retrieval", Foundation and Trend in Information Retrieval, Vol.2, No.4 (2008), pp. 215-322.
[32] Smeaton, A. F., Over, P., and Kraaij, W.: “2006. Evaluation campaigns and TRECVid”; In Proc. 8th ACM International Workshop on Multimedia Information Retrieval (USA, October 26 - 27, 2006). MIR '06. ACM Press, New York, NY, pp. 321-330.
[33] Sivic, J. and Zisserman, A.: “Video Google: A text retrieval approach to object matching in videos”, IEEE International Conf. on Computer Vision (ICCV’03), 2003.
[34] Lowe, D.: “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision (IJCV), 2(60):91–110, 2004.
[35] Mikolajczyk, K. and Schmid, C.: “An affine invariant interest point detector”, European Conference on Computer Vision, Springer-Verlag, 2002.

[36] Matas, J., Chum, O., Urban, M. and Pajdla, T.: “Robust wide baseline stereo from maximally stable extremal regions”, British Machine Vision Conference (BMVC’2002), pp. 384–393, 2002.

[37] Fergus, R., Perona, P. and Zisserman, A.: “Weakly supervised scale-invariant learning of models for visual recognition”, Int. Journal of Computer Vision, 71(3):273–303, 2007.

[38] Leibe, B., Leonardis, A. and Schiele, B.: “Robust object detection with interleaved categorization and segmentation”, IJCV, 77(1-3):259–289, 2008.

[39] Jiang, H., Drew, M.S., Li, Z.: “Matching by linear programming and successive convexification”, IEEE Trans. PAMI,29:959–975, 2007.

[40] Li, H., Kim, E., Huang, X., He, L.:"Object matching with a locally affine-invariant constraint", IEEE International Conf. on Computer Vision and Pattern Recognition (CVPR 2010), pp.1641-1648, 2010.

[41] Fei-Fei, L. and Perona, P.: “A bayesian hierarchical model for learning natural scene categories”, IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[42] Tola, E., Lepetit, V. and Fua, P.: “A fast local descriptor for dense matching”, IEEE International Conf. on Computer Vision and Pattern Recognition (CVPR2008), 2008.

[43] Tuytelaars, T. and Schmid, C.: “Vector quantizing feature space with a regular lattice”, IEEE International Conf. on Computer Vision (ICCV2007), 2007.

[44] Tuytelaars, T.: "Dense Interest Points", IEEE International Conf. on Computer Vision and Pattern Recognition 2010 (CVPR2010) , pp. 2281-2288.

[45] Browne, P., Smeaton, A.F.: "Video retrieval using dialogue, keyframe similarity and video objects," IEEE International Conf. on Image Processing (ICIP 2005), pp. III- 1208-1211, 11-14 Sept. 2005.

[46] Foley, C., *et. al*: “TRECVID 2010 Experiments at Dublin City University”, TRECVid 2010 - Text REtrieval Conference TRECVid Workshop, Gaithersburg, MD, Nov. 2010.

[47] Gorisse, D., et. al: “IRIM at TRECVID 2010: Semantic Indexing and Instance Search”, TRECVid 2010 - Text REtrieval Conference TRECVid Workshop, Nov. 2010.

[48] Ren, X., Malik, J.: Learning a classification model for segmentation. IEEE International Conf. on Computer Vision (ICCV’03), vol. 1, pp. 10–17 ,2003.

[49] Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: “Multi-class segmentation with relative location prior”, International Journal on Computer Vision, 2008.

[50] Malisiewicz, T., Efros, A.: “Improving spatial support for objects via multiple segmentations”, British Machine Vision Conference (BMVC’2007), 2007.

[51] Chevalier, F., Domenger, J.P., Benois-Pineau, J., Delest, M.: “Retrieval of objects in video by similarity based on graph matching”, Pattern Recognition Letters 28(8): 939-949, 2007

[52] Vieux, R., Benois-Pineau, J., Domenger, J.-P., Braquelaire, A.: “Segmentation-based multi-class semantic object detection”, Multimedia Tools and Applications, pp. 1-22, 2010

[53] Kim, K., Grauman, K.: “Boundary Preserving Dense Local Regions”, IEEE International Conf. on Computer Vision and Pattern Recognition 2010.

[54] Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., Yamada, A.: “Color and Texture Descriptors,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 6, pp. 703-715, June 2001.