Analysis and interpretation of visual scenes through collaborative approaches

Sabin Tiberius Strat

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Sabin Tiberius STRAT

Thèse dirigée par **Patrick LAMBERT**
codirigée par **Dan Alexandru STOICHESCU**
et co-encadrée par **Alexandre BENOIT**

préparée au sein **laboratoire LISTIC et laboratoire LAPI**
et de **ED SISEO**

Analyse et Interprétation de Scènes Visuelles par Approches Collaboratives

Analysis and Interpretation of Visual Scenes through Collaborative Approaches

Thèse soutenue publiquement le **04 décembre 2013**,
devant le jury composé de :

Mme. Jenny BENOIS-PINEAU
Professeur de l'Université de Bordeaux 1, Rapporteur
M. Frédéric PRECIOSO
Professeur de l'Université de Nice Sophia Antipolis, Rapporteur
M. Bernard MERIALDO
Professeur, EURECOM Sophia Antipolis, Examinateur
M. Stéphane BRES
Maître de conférences de l'INSA de Lyon, Examinateur
M. Mihai CIUC
Maître de conférences de l'Université "Politehnica" de Bucarest, Examinateur
M. Patrick LAMBERT
Professeur de l'Université de Savoie, Directeur de thèse
M. Dan Alexandru STOICHESCU
Professeur de l'Université "Politehnica" de Bucarest, Co-Directeur de thèse
M. Alexandre BENOIT
Maître de Conférences de l'Université de Savoie, Co-Encadrant de thèse
M. Georges QUENOT
Directeur de recherche (CNRS), Laboratoire d'Informatique de Grenoble, Invité

UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI

TEZĂ

Pentru obținerea gradului de

DOCTOR AL UNIVERSITĂȚII POLITEHNICE DIN BUCUREȘTI

Specializarea: **Informatică**

Prezentată de

Sabin Tiberius STRAT

Teză coordonată de **Patrick LAMBERT**
co-dirijată de **Dan Alexandru STOICHESCU**
și co-incadrată de **Alexandre BENOIT**

pregătită în cadrul **laboratorul LISTIC și al laboratorului LAPI**
și al **ED SISEO și Școlii Doctorale "Electronică, Telecomunicații și Tehnologia Informației"**

Analiza și Interpretarea Scenelor Vizuale prin Abordări Colaborative

Analysis and Interpretation of Visual Scenes through Collaborative Approaches

Teză susținută public în data de **04 decembrie 2013**,
în fața juriului compus din:

Dna. Jenny BENOIS-PINEAU
Profesor, Université de Bordeaux 1, Raportor
Dl. Frédéric PRECIOSO
Profesor, Université de Nice Sophia Antipolis, Raportor
Dl. Bernard MERIALDO
Profesor, EURECOM Sophia Antipolis, Examinator
Dl. Stéphane BRES
Conferențiar, INSA de Lyon, Examinator
Dl. Mihai CIUC
Conferențiar, Universitatea Politehnică din București, Examinator
Dl. Patrick LAMBERT
Profesor, Université de Savoie, Director de teză
Dl. Dan Alexandru STOICHESCU
Profesor, Universitatea Politehnică din București, Co-Director de teză
Dl. Alexandre BENOIT
Conferențiar, Université de Savoie, Co-Supervizor de teză
Dl. Georges QUENOT
Director de cercetare (CNRS), Laboratoire d'Informatique de Grenoble, Invitat

Acknowledgments

To be written for the final version of the manuscript, after the defense.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 More and more multimedia data

In the last decade, our society has experienced significant advances in electronics and digital technology, with prices for consumer electronics and gadgets going down significantly. In the year 2000, digital cameras were rare, and those that existed offered limited performance in terms of resolution and maximum number of images that one could acquire. As for video cameras, recording was still done generally on magnetic tapes, which also served for storage of the video content. Camera phones were something completely unheard of at that time among most consumers. And of course, if someone had a "database" of images or videos, it simply consisted of many albums of photos on paper, negatives and slide film, or boxes full of video tapes recorded on various occasions.

As the years passed, technology improved to the level that digital cameras and camcorders of higher and higher quality have become affordable in all developed countries. It is in fact becoming difficult now (year 2013) to buy a mobile phone that *doesn't* have an integrated camera, because the costs of including one have decreased so much. Even phone cameras have progressed a lot, to the point that their quality is almost the same as that of compact digital cameras, and because most people carry their phone everywhere they go, they can take photos (or record videos) anytime and anywhere, at the simple push of a button.

This increase in the ease of acquiring images, videos or audio recordings has also been accompanied by an increase in resolution of image sensors. As of 2013, compact digital cameras have resolutions of around 10-15 Megapixels, enthusiast yet still relatively affordable cameras can go up to 24 Megapixels, while professional cameras with large sensors can have resolutions in excess of 40 Megapixels. The line between digital cameras and camcorders is also becoming more and more blurred, as most digital cameras sold in 2013 can also acquire high-definition video (1080 lines progressive (non-interlaced) at 30 or even 60 frames per second). Phone cameras too are getting better and better at capturing

images and videos, with a 2013 high-end smartphone being able to take 8 Megapixels photos and capture HD video.

At the same time, storage has also become much cheaper. If in the year 2000, a 40GB hard-disk was considered of large capacity, as of 2013, one can easily buy a 1TB hard-disk for a very affordable price. Such a hard-disk would allow storing in the order of 200000 high-quality still images (15 Megapixels with a low JPEG compression ration to give 5MB/image), or hundreds of hours of video (depending on the resolution, frame rate, compression algorithm and compression ratio used). Moreover, online storage of multimedia is now possible on websites such as Facebook, Instagram, Youtube etc., and users are in fact encouraged to upload, share and tag their content. Therefore, as of 2013, digital archiving has almost completely replaced analog archiving of still images, videos or audio recordings (multimedia), and many people have very large collections of multimedia files on their computers or on web servers.

1.2 The need to organize

With so many files, in order to be able to find a certain element later on, the user has to be extremely well-organized when adding new files to the database. However, many ordinary users' skills at organizing their personal collection of multimedia only go so far as to having a *"Various stuff"* folder on their computer, in which subfolders with "very suggestive" names with respect to their content are created, such as *"New folder (1)"*, *"New folder (2)"* or *"100NCD40"*. Afterwards, when the user wants to retrieve a particular photo or video, the strategy for many people consists in more-or-less randomly clicking on subfolders, hoping to find the desired element, which isn't particularly efficient.

This has led to the development of dedicated software which helps users to better organize their multimedia collections. In the case of photo organizing software (for example *Picasa*[1]), it usually allows the user to assign labels/tags or even captions to pictures, to add star ratings, to group pictures into albums and to easily move them to and from various folders, and to easily share photos on social networking websites. Browsing the database or searching for a particular picture can be done by entering desired tags, by specifying the album, and/or the date when the picture was taken etc. Geotagging is also increasing in popularity, as more and more cameras are equipped with GPS modules that can add the geographical location where the picture was taken to the picture's metadata. This information can also be used to filter only a subset of photos.

Such software can greatly help people to organize their multimedia collections, however one problem still remains: if the user wants to be able to search inside his database according to criteria more complex than just the date and time when the picture was taken and/or the geographical location, tags (indexing terms) of a higher semantic level are required. For example, searching for photos of "mother baking a cake" would require semantic tags such as "mother", "bake/baking", "cake", "kitchen" etc. to be used as search terms. In general, such tags need to be *manually-specified* by the user, but this is a tedious task when there are many files to annotate.

[1] `http://picasa.google.com/`

On-line databases experience similar problems, but to an even higher degree because of the numerous users that submit content. For example, on the Youtube[2] video website, 100 new hours of video are uploaded every minute by its users[3]. If these videos would not be properly annotated by the uploaders (with an adequate title and appropriate keywords), they would be impossible to find by others. Currently, this annotation must be done manually by the uploader. If we also demand the possibility of retrieving just a short sequence *inside* a video, the annotation needs to be even more detailed, at the sequence level, which requires even more effort from the part of the uploader.

It would be very helpful if these semantic tags could be assigned automatically, which brings us to the problem of automatic *semantic indexing* of multimedia content, which is the topic of this thesis. Using *computer vision* techniques, a computer could examine the image or video automatically and determine the semantic content: what the multimedia element is about, where the action takes place, who are the main characters and what do they do, what objects are present in the scene, if there are any unusual events etc. and the computer would then annotate the multimedia element with the corresponding keywords. Such an automatic annotation could be done in much more detail: for a long video, a human user might only be able to annotate the basic ideas, whereas a computer could annotate the different scenes of the video individually, so that not only the video could be searched, but specific scenes within the video (such as finding the moment when "Mother puts the cake in the oven").

Such a database management system, able to retrieve multimedia elements based on their *semantic* content and not just low-level tags (such as date, time and geographical location) is called a *Content-Based Multimedia Retrieval* system.

1.3 Examples of applications

A content-based multimedia retrieval system that can automatically assign semantic labels to database elements (performing *semantic indexing*), and later use these labels to help users search for specific elements, would find applications in many areas.

For example, users of the Flickr[4] or Picasa[5] on-line picture sharing websites, or of the Vimeo[6] or Youtube[7] video sharing websites would no longer need to manually set labels for the content they upload, in order to allow others to search for and retrieve this content. The same would also be true for multimedia collections stored on a personal computer and organized using such software.

In the case of stock photography websites (such as Getty Images and Dreamstime), which help uploaders to sell their photos (usually to advertisers), correct labels are even more important, as they maximize the sales of photos; again, computer vision techniques could aid in analyzing the content and tagging the photo automatically or suggesting new

[2] www.youtube.com
[3] http://www.youtube.com/yt/press/statistics.html
[4] www.flickr.com
[5] picasa.google.com
[6] vimeo.com
[7] www.youtube.com

tags to the user. Advertisers searching for photos would also benefit from this annotation via a "smart browser": this would allow them to view semantically-relevant photos easier, to view more images similar to a certain query image, to request images with the same keywords but in different contexts (in order to have a more diverse retrieval response from the system), to quickly navigate between semantically-related terms (and view associated images) and so on.

Television networks would also benefit from such video database organizing tools, as it can help them to archive their broadcasts automatically. At a later time, when a certain part of old material becomes relevant for current events, that old part can be searched and easily retrieved from the database thanks to annotations that were made automatically (such as who was the person being interviewed, what he/she was saying etc.), accompanied by ordinary metadata (the name of the show, the date and time etc.).

Regarding already-implemented applications of content-based multimedia indexing and retrieval, we can name a few popular examples.

The social networking website Facebook[8] allows users to tag their friends in uploaded photos, to facilitate searching for photos of certain people later on. However, tagging a friend in many photos is a tedious task. Facebook facilitates this by making the user manually tag his/her friend in just a few photos, and then employing automatic face detection and recognition software to suggest tags of the same person in other photos [9]. This constitutes a partially-automatic indexing system (the user needs to validate the proposed tags) for annotating pictures with the persons present in them, while retrieval can be done by requesting annotated pictures of that person later on.

Youtube does not yet (as of 2013) implement a system with automatic semantic indexing, however it implements *content-based copy detection*, which is a content-based video retrieval system used to combat piracy. Copyright holders (such as music or movie publishing agencies) can send to Youtube a copy of the content that they do not want pirated. When a regular user uploads a video, its content is automatically analyzed and compared to the database of copyrighted works. If the video is found to be (a part of) a copyrighted work such as a music video, a fragment of a commercial movie etc., Youtube contacts the copyright holder and lets it decide what to do: either block the video or use it to promote the original content (e.g. by overlaying links to where the viewer can buy the original DVD or music CD) [10].

For Youtube videos, there are third-party providers of automatic rich tagging services. For example, the company Video Semantics[11] has enabled a video segmentation and tagging engine that *"allows content producers automatic insertion of the rich metadata into YouTube hosted videos. Keywords created by the Video Semantics software can include the topics in the video, related concepts, relevant categories or other information of interest to the viewer"*. Content producers can thus benefit from a higher viewability and exposure,

[8] `www.facebook.com`

[9] `https://www.facebook.com/notes/facebook/making-photo-tagging-easier/467145887130` retrieved on 30/09/2013

[10] `http://www.youtube.com/t/contentid`

[11] `http://www.videosemantics.com/site.php/`, retrieved on 30/09/2013

leading to higher ranks in search results[12]. Therefore, in this example, semantic indexing is assured by the company's tagging engine, while retrieval is managed by Youtube's search engine (which matches a user's search terms with the tags of a video).

Google Image Search allows users to search the web for images. Search queries can be formulated with keywords, but also by giving an example image, which makes Google search for visually-similar images. As of June 2013, their system is also capable of *semantic concept recognition* in images that have not been labeled by a human, being able to recognize more than 1000 image classes[13]. Thanks to the addition of semantic concept recognition, this system is now capable of both indexing (images on the web can be indexed with the recognized image classes) and retrieval (the classes of a query image are determined and used as search terms in the database).

These examples support the idea that automatic semantic indexing (semantic tagging) in the context of content-based multimedia retrieval is a very hot research topic in the computer vision community, as it has great potential for commercial applications. This motivates the work undertaken in this thesis, which consists in *exploring automatic semantic indexing algorithms on very large video databases*.

1.4 Context, goals and contributions of this thesis

Semantic indexing of video datasets, which is the focus of this thesis, is a research topic at the boundary of several fields, as shown in Figure 1.1. It requires knowledge of computer vision, image (or video) processing and analysis, machine learning and information fusion. The lines between these domains are not clearly defined, but we could say the following:

- Image/video processing and analysis tools are needed for extracting descriptions of a very low semantic level from the video (such as the dominant colors, the dominant contour orientations, the dominant motion directions etc.); these characterize the aspect of the video in a machine-understandable and compact form. The temporal segmentation of videos according to their temporal structure could also be included in this category, although it can also be considered a computer vision tool.
- Computer vision tools are used to aggregate the previous descriptors into representations (such as the Bag of Words descriptor which will be seen later in Section 2.3.4).
- Machine learning tools are used to train supervised classification algorithms. A classifier has the role of predicting whether or not a video belongs to a class (contains a certain semantic concept), based on the representation of a video from the previous point. The classifier is trained by giving it a set of annotated example videos, which enables it to automatically learn the rules for classifying other videos as well.
- At the end, information fusion strategies are needed to take advantage of complementary information coming from different sources.

[12] `http://www.prweb.com/releases/2013/7/prweb10968938.htm`

[13] `http://googleresearch.blogspot.fr/2013/06/improving-photo-search-step-across.html` retrieved on 30/09/2013

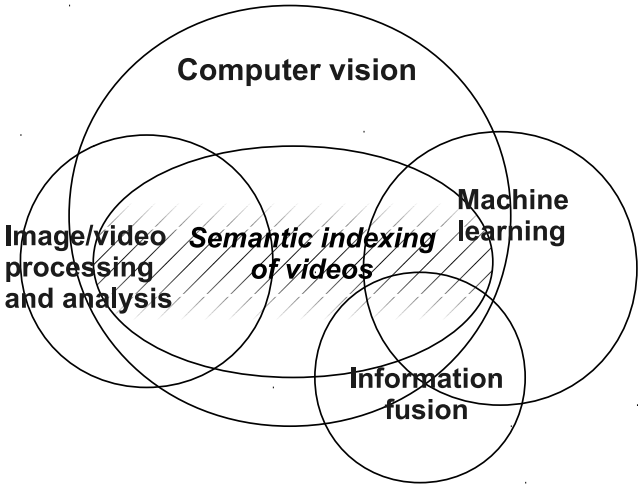


Figure 1.1: Concerned scientific areas of this thesis: semantic indexing of videos requires tools from computer vision, image or video processing and analysis, machine learning and information fusion.

As opposed to specific sub-domains of video analysis, such as event detection in surveillance videos, our goal is to devise an automatic semantic indexing system which is to be as generic as possible, able to deal with many semantic concepts of different types (not only events/actions, but also objects, characters, scene types etc.) and in very diverse contexts. A large number of target concepts means that concept detection algorithms specifically conceived for each concept are impractical, thereby motivating research into generic concept detection methods.

To this end, most of our work is done on the TRECVid Semantic Indexing datasets, which are very generic datasets for semantic concept detection in videos, however we also perform a few experiments on the KTH dataset for action recognition.

Within an automatic semantic indexing framework, the contributions of this thesis are in the image/video description and representation and in the information fusion domains. We do not construct an automatic semantic indexing system from scratch (this is a difficult task, that needs much more time than the duration of a thesis), instead we start from a state-of-the-art system [Ballas 2012b] (described in Section 2.2) used by our partners in the IRIM[14] French research consortium. We improve certain aspects of this system in a three-fold contribution consisting of the following elements:

- Our first contribution is in the image/video processing and analysis domain. We propose a method of *augmentic standard gradient-based image descriptors, such as SIFT [Lowe 2004a] or SURF [Bay 2008]*, in order to improve their genericity and precision at concept recognition. This method is based on *preprocessing the video frames with a model of the biological human retina* from [Benoit 2010]. SIFT/SURF descriptors are based on histograms of oriented spatial gradients of the light intensity, therefore, they are purely *spatial* descriptors. The retinal preprocessing improves the

[14] http://mrim.imag.fr/en/

overall results for concept detection, while also extending the descriptors by adding *spatio-temporal* behaviours. We discuss this method and its performances in detail in Chapter 3.

- Our second contribution is also in the domain of video processing and analysis. It consists in a *battery of trajectory descriptors, dedicated to representing the motion content* of videos. These trajectory descriptors are inspired from the existing state-of-the-art, with a few modifications, and are discussed in Chapter 4. Starting from standard SIFT descriptors, going through SIFT with retinal preprocessing and ending with trajectories, we progress from purely spatial, to spatio-temporal and in the end, to purely temporal descriptors. This brings us to our third contribution.

- Our third contribution is in the domain of information fusion. Because we now have descriptors with various properties, on top of which we applied supervised classifiers, we *exploit the complementarity* between descriptors by performing a *late fusion* of their supervised classifier ouputs. We compare several late fusion approaches and discuss their working principles and performances in Chapter 5.

The rest of the thesis is structured as follows: Chapter 2 presents the state of the art concerning the domains of our contributions, while Chapters 3, 4 and 5 describe our three-fold contribution. Chapter 6 concludes the thesis and opens the path for future developments.

CHAPTER 2

State of the art

Contents

2.1 Generalities about Content-Based Video Retrieval

The ultimate goal of a Content-Based Video Retrieval (CBVR) system is for a user to enter a query in human-understandable words, such as find the movie scene in which “Alice falls down the rabbit hole” (as a text query) and the system will be able to retrieve this scene from the “Alice in Wonderland” movie. Optionally, the user might specify additional constraints, such as whether the desired movie is with human actors or if it is an animation film, or from which year the desired movie is etc. The query model can even be extended to permit queries by multimedia samples (such as the Google Image search for visually similar images). The part of the CBVR system that insures the interaction of the user with the video database is called the *browsing and searching* part. This part would interpret the user’s query, transform it into a machine-understandable form (search terms) and conduct the search on the *indexed* video database.

In order for the database to be searchable by content, it must be *indexed* by content. This is the job of the Content-Based Video Indexing (CBVI) part of the CBVR system. When a new video is added to the database, the CBVI component would do the following:

1. determine the temporal structure of the video: the acts, scenes and shots of the movie; for example, it would identify the part when Alice falls into the rabbit hole as a separate scene or shot;

2. annotate the temporal elements of the video (such as the scenes or shots) with various keywords that illustrate the content of the scene or shot; for example, possible keywords for our scene would be "Alice" or "girl" (if the system is unable to identify the character), "falling", "tunnel/hole" and whatever magical items Alice sees doing her fall.

3. If the indexing system is very intelligent, it might even assemble the words characterizing the video into a short sentence such as "a girl falls down a tunnel with magical items flying by", constructing an automatic summary of the movie. This would constitute a very high-level semantic understanding of the video by the computer, but the state-of-the-art is not yet capable of such performances (or at least not on generic videos).

This thesis focuses on the second point above, the automatic assignment of keywords to video shots. Most of the work is performed on the TRECVid Semantic Indexing task, introduced in Section 2.6.3. The videos from this database are already divided into temporal elements (called *shots*) by an official TRECVid automatic shot segmentation tool [Smeaton 2010], giving shots with lengths between a few seconds to several tens of seconds. The index terms (the semantic concepts to detect) are also fixed: there is an official list of 346 various semantic concepts (examples in section 2.6.3). Therefore, what remains to be done (and the subject of our work) is to return, for each shot and each concept, the likelihood of the shot to contain the concept (expressed by a number between 0 and 1, where 0 means absence of the concept from the shot).

In more recent editions of TRECVid, an optional *concept pair* detection task was introduced, but we did not work on it. Also, new to the 2013 edition, an optional concept localization task was introduced, in which the moments in the shot when the concept is present and its spatial location in the video frames must also be specified [Over 2012]. We did not experiment with localization in our work.

In the following, Section 2.2 will describe a general framework of how semantic indexing on video datasets can be achieved. Even if we will give some particularities for the TRECVid Semantic Indexing dataset, the framework can be easily adapted to other datasets and different tasks. Because part of the contribution of this thesis is the development of spatio-temporal video descriptors, Section 2.3 will give an insight into existing video descriptors, with an accent on spatio-temporal ones, some of them used in TRECVid. Afterwards, Section 2.4 will show how information from multiple sources can be fused to augment concept recognition performances, which represents another area of contribution of this thesis. Base on this state of the art, Section 2.5 will point out the needs that we identified in the context of TRECVid and how we address them. Section 2.6 concludes this chapter with a description of a few popular video datasets for evaluating algorithms, with an accent on the TRECVid Semantic Indexing datasets on which we perform most of our experiments.

2.2 General framework for semantic indexing

There can be many solutions for the task of identifying semantic concepts in videos. One of the best-performing strategies at the moment consists in the following steps:

1. *extracting descriptors* from the video shots; descriptors characterize various aspects (and modalities) of the video, such as the dominant colors, dominant orientations, motion patterns, sounds, overlaid text etc.; more details will be given in Section 2.3;

2. training and applying *supervised classifiers* on the video shots, for each target semantic concept; during the training phase, based on a set of examples, supervised classifiers can automatically learn rules that allow them to distinguish between classes (whether or not a video element contains a target concept). After training, classifiers will be able to predict, based on the descriptor on which they were trained, whether or not a *new* shot contains the targer concept (alternatively, the classifier can give a score between 0 and 1, if a strict decision is not demanded);

3. *late fusion* of classification results (classification scores in the case of TRECVid), whereby the predictions from classifiers based on various descriptors are aggregated to improve reliability; this way, different "points of view" are taken into consideration, which generally leads to more reliable results; fusion strategies are discussed in more detail in Section 2.4;

4. optionally, further post-processing of results can be done, such as considering the temporal neighborhood of shots in the video, or the semantic relations between concepts;

5. evaluation of results: for the case of the TRECVid SIN task, as seen in Section 2.6.3, average precision is computed for each concept;

Particularities for TRECVid: Our participation at the TRECVid Semantic Indexing (SIN) task was done as part of the IRIM[1] group. The IRIM processing chain for semantic indexing is detailed below, and summarized in Figure 2.1. It follows the general framework stated previously.

The first step is *descriptor extraction*. The members of the group shared the descriptors that they computed on video shots, constituting a battery of several tens of various descriptors (and different parameter versions of them), such as color histograms, texture descriptors, SIFT/SURF Bag-of-Words descriptors, facial tracks, trajectories, audio descriptors, overlaid text in the videos, presence of various lower-level semantic concepts etc. [Ballas 2012b]. This ensured that the video shots were described in a very diverse way, so as to capture various aspects of the content. For most of the descriptors, only one (or several) keyframe(s) were analyzed instead of the entire video shot, to reduce computation time, using the official selection of keyframes from TRECVid [Ballas 2012b].

[1] Multimedia Information Modeling and Retrieval, `http://mrim.imag.fr/en/`

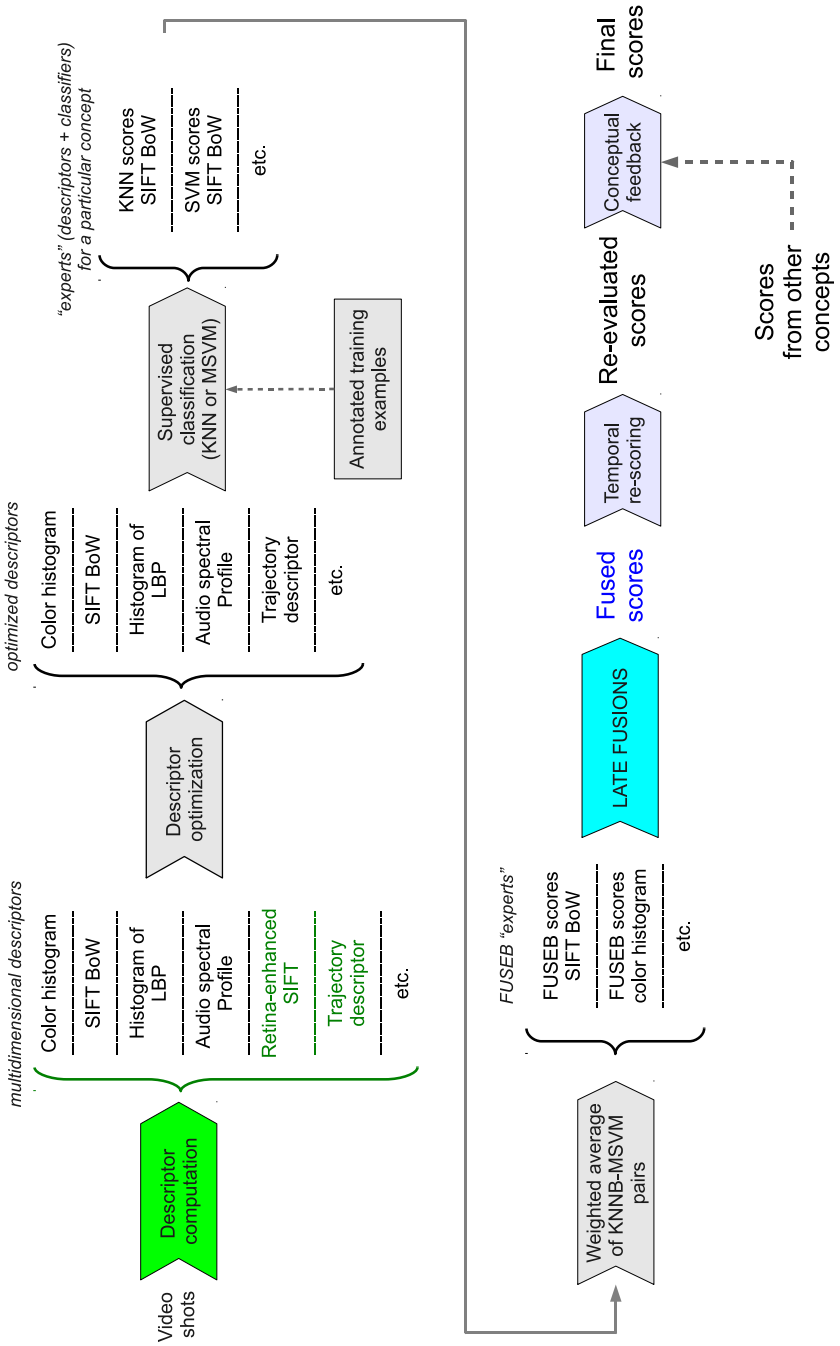


Figure 2.1: Processing chain for semantic indexing of the IRIM group [Ballas 2012b]. The figure illustrates the processign chain for a single concept, other concepts intervene only in the conceptual feedback step. First, multidimensional descriptors are extracted from the video shots, followed by descriptor optimization. On each (optimized) descriptor, a KNN and a MSVM classifier are applied. We call *"expert"* the combination of a descriptor with a supervised classification algorithm. Afterwards, all the experts are fused, and on the fusion result, temporal re-scoring and conceptual feedback are applied. Our contribution consists in a set of retina-enhanced SIFT BoW descriptors, a set of trajectory BoW descriptors and an automatic late fusion method.

Optionally, IRIM partner LIG[2] could *optimise* the resulting descriptors so that the supervised chassifiers would work better with them. This optimisation consisted in applying a *power transformation* to normalize the values of the descriptor dimensions, followed by *Principal Component Analysis (PCA)* to make the descriptor more compact, and at the same time, more robust [Safadi 2013].

The next step was to train and apply *supervised classification* algorithms on each of the descriptors. To this end, IRIM partner LIG used an implementation of the K-Nearest Neighbours (KNN) classifier [3] to generate, for each target concept and each video shot, a classification score (between 0 and 1) indicating the likelihood of the shot to contain the concept. An alternative to KNN was a multiple learner approach based on Support vector Machines (SVM), called MSVM [Safadi 2010]. MSVM gives better performance than KNN, but is more computationally expensive [Ballas 2012b].

At this point, for each video shot, for each concept and for each descriptor, we have the KNN and MSVM classification scores. The next step is a *late fusion* of classification scores for the current concept and shot, taking into consideration the scores from all possible combinations of descriptors and supervised classifiers. The late fusion is in effect similar to taking an "average opinion" from all the combinations of descriptors and supervised classifiers for the current shot and concept. Several fusions approaches are used, they will be discussed in more detail in Section 2.4 and in Chapter 5.

After the late fusion step, we dispose, for each concept, of the classification scores on all video shots. Because a concept that is present in a shot of a video also tends to be present in the neighboring shots of the same video due to temporal correlation, a *temporal re-scoring* of shots can be performed in order to take advantage of the temporal context. The approach is described in [Safadi 2011] and leads to an increase in average precision.

The last step undertaken in the IRIM group is applying *conceptual feedback* on the classification scores [Hamadi 2013]. This exploits the semantic relations between concepts by constructing a new descriptor with 346 dimensions (exactly the number of concepts), the i^{th} dimension of this descriptor being the classification score of the shot with the i^{th} concept. Supervised classification is applied on this descriptor as if it were a normal descriptor, and the resulting classification scores are re-fused with the previous results. Combined with temporal re-scoring, on TRECVid 2012, the authors report a 15% increase in mean average precision.

This approach, although illustrated on the TRECVid SIN dataset, can in fact be easily adapted to index other multimedia datasets as well. The main change required would consist in computing descriptors adapted to the type of multimedia content being analyzed (e.g. we will not compute motion descriptors on datasets of static images). Adaptations would also be required for the last two stages, the temporal and semantic context, in order to exploit a type of context that makes sense for the dataset in question.

[2] Laboratoire d'Informatique de Grenoble, `http://www.liglab.fr/?lang=fr`
[3] `http://mrim.imag.fr/georges.quenot/freesoft/knnlsb/index.html`

2.3 Descriptors for video content

Semantic concepts cannot be recognized directly from the video stream. One cannot train supervised classification algorithms directly on the video frames, for the following reasons: first, the possible variability of how the voxels (pixels in each frame) look is enormous. For example, considering small videos of only 320x240 pixels and 100 frames, with 3 color channels (RGB) and encoded on 8 bits (256 levels for each color channel), there would be almost $6 \cdot 10^9$ (5898240000 to be exact) possible videos. Of course, only a small fraction of these would actually make sense to a human, the rest being just noise. Second, such a video would be represented on 23 MB (23040000 bytes to be exact), which would overwhelm supervised classifiers. And we didn't even consider the sound from the video.

The solution is to extract *descriptors* from the video stream. Descriptors are representations of the video that are more meaningful (they try to encode only useful information) and much more compact (the small video above can be represented on a few bytes or a few thousands of bytes, depending on the descriptor, instead of the original 23 MB) than the raw video data.

Additionally, descriptors are usually conceived in such a manner as to be *robust* to various transformations (such as image translations, rotations, scale changes) or variations (of brightness, slight color differences), to small amounts of image noise or compression artifacts, to the exact spatial locations of elements in the image, to the sound volume etc. Robustness of descriptors is what allows supervised classifiers trained on top of these descriptors to generalize: from just a limited set of training examples, the classifier will be able to recognize semantic concepts in new videos, that can even be acquired under different circumstances than the training examples.

In practice, a compromise is always made between the robustness of a descriptor and its *discriminative power*: generally, the more robust a descriptor is, the less discriminative it is, and the less able to distinguish between different concepts. Ideally, the descriptor would be robust to uninteresting changes of the video (such as slight camera rotation, camera shake, lighting conditions etc.) but discriminative with respect to semantically-meaningful changes (such as how a person moves to execute an action).

Descriptors can represent different types of information from the video, such as colors, textures, shapes/contours, motion or audio. Some semantic concepts can be captured more efficiently by certain descriptors, for example, the color "green" can indicate vegetation, while certain motion patterns can indicate "dancing". Because the concepts in TRECVid are very numerous and varied, we therefore have interest in extracting as many descriptors and descriptor types as possible, and we will determine later which of them is more appropriate for which concept. In the following, we will give some examples of commonly used descriptors for video indexing.

2.3.1 Color descriptors

A very common way of representing color is with *color histograms*. When applied to a video, either the colors in all frames are examined, or, to speed-up descriptor extraction, only a few frames (or even a single frame, called the *keyframe*) are analyzed. IRIM partner

ETIS has contributed color histogram descriptors in the L*a*b* color space, quantized on 256, 512 and 1024 colors, computed on the keyframe of each video shot, with 1x1, 1x3, 3x1 and 2x2 spatial divisions of the keyframe [Gosselin 2008].

2.3.2 Texture descriptors

The texture on a surface gives information about the object that the surface belongs to. For example, a foliage texture indicates vegetation, different species of trees have barks textures in different ways, fish or snakes have scales on their skin, a leopard has spots etc.

Some examples of texture descriptors are:

- (histograms of) local binary patterns [Ojala 1996, Delezoide 2011, Zhu 2011];
- Gabor filter banks [Turner 1986];
- quaternionic wavelets [Gosselin 2008]

2.3.3 Audio descriptors

Mel-frequency cepstral coefficients (MFCC) represent the short-term power spectrum of a sound. IRIM partner LIRIS submitted a Bag-of-Words descriptor based on MFCCs [Ballas 2012b].

2.3.4 Bag of Words descriptors based on local features

There is a class of descriptors that characterize small, local parts of the image (image patches, *local features*), as opposed to descriptors that try to characterize the entire image. After characterizing the local features, an aggregation strategy is employed to characterize the entire image based on the local features.

Most commonly, the local features are aggregated using a simple, orderless model, called the *Bag of Words (BoW)* model (or *Bag of Visual Words*) [Csurka 2004]. The principle is illustrated in Figure 2.2.

In order to construct a BoW descriptor based on local features, the following steps need to be done:

1. choose a set of local features to characterize (choose the image subparts that we want to describe);
2. describe the image patches around the local features using a local descriptor (a descriptor for small image patches);
3. extract many local features from some “training” images, and cluster their descriptions into a *dictionary* of “visual words”, for example by using k-means clustering [Arthur 2007];
4. for an image that we want to represent, we extract and describe local features; then, we approximate each local description with its closest-matching dictionary word; the

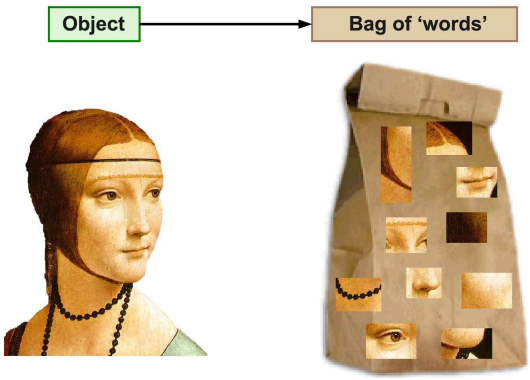


Figure 2.2: Basic principle of the BoW model: an image is represented as an orderless collection (a "bag") of subparts. The face is composed of two eyes, a nose, a mouth etc. The relative positions of these subparts are not taken into consideration. Image credit: *Li Fei-Fei, Rob Fergus, and Antonio Torralba. Recognizing and learning object categories - short course. 2009*

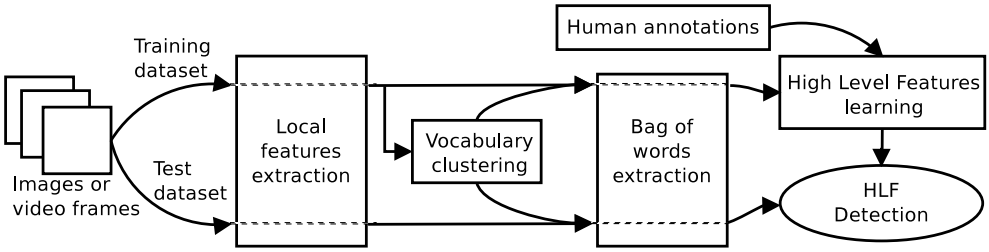


Figure 2.3: The state-of-the-art Bag-of-Words processing toolchain for semantic concept (High-Level Feature) detection

entire image is represented as a *histogram of visual words*, that says how often each type of local feature appears in the image;

The resulting histograms of visual words constitute the BoW descriptor. Supervised classification algorithms are applied afterwards, to find the link between different BoW histograms and different semantic concepts. The entire process is illustrated in Figure 2.3.

Compared to global representations, BoW have the advantage that they are robust to partial occlusion: the absence of a few elements out of many does not have a great impact upon the descriptor. Additionally, because the relative positions of the local features is not considered, invariance to viewpoint changes and global deformations is more easily obtained (as long as the method of describing each local feature is also invatiant to these changes). The BoW model has proven itself successful for image classification and object recognition[Csurka 2004].

Among derivatives of the Bag of Words model, we can name the *Bag of regions* model [Vieux 2012]: instead of working with highly-localized image features (small image patches), a segmentation algorithm is employed to divide the image into regions. Each region is described independently and the region descriptions are fed into the BOR model,

the other steps being similar to the Bag of Words framework. To facilitate understanding, it could be said that Bag of Regions is just like Bag of Words, however it does not use *local* features but *regional* (larger) features.

There exist other methods of aggregating local features into a representation of the entire image/video, such as correlogram features [Savarese 2006] which capture spatial co-occurrences of features, relative positions of local features [Sudderth 2005], spatial pyramid matching [Lazebnik 2006], or, for encoding the temporal structure of actions, actom sequence models [Gaidon 2011]. However, in the case of the TRECVid Semantic Indexing task, where annotations are only available for the entire video shot (we do not know where and when exactly the concept appears in the shot), we prefer to use the BoW model because of its simplicity and adaptability.

Next, we will discuss strategies of choosing and describing local features for the BoW model.

2.3.4.1 Choice of local features

Generally, there are two ways of choosing local features in images:

- with a *feature detector* which detects image patches with certain properties, such as high curvature (corners);
- by sampling the image regularly (such as every 5 pixels along the horizontal and vertical), along what is called a *dense grid*;

Either way, we end up with a selection of points from the image. Around these points, small image patches will be considered, and these patches will be described in a later step. Both the detection of features and the description of patches can be done at various spatial scales if desired, in order to account for the possible variation in the scale of objects and/or to capture information about both a more general shape and about minute details.

The work of [Tuytelaars 2008] gives a detailed review of the most common feature detectors. We mention here some of the most popular and some more recent detectors:

- the Harris corner detector, which chooses points that maximize a cornerness measure based on the second-moment matrix; it detects points with high spatial curvature; it is rotation-invariant (a point is still detected even if the image was rotated); the Harris-Laplace extension also detects the scale at which this high curvature is most evident, while the Harris-Affine extension can also deal with affine deformations (the object is deformed more along an axis than along another);
- the Hessian blob detector chooses points that maximize the determinant and the trace of the Hessian matrix; they tend to detect features that resemble more or less “spots”(“blobs”), hence the name “blob detector”; it is rotation-invariant; it also has extensions that deal with scale or affine deformations;
- the Difference of Gaussians (DoG) is an efficient implementation of a blob detector that finds extrema of the Laplacian of Gaussian (LoG);

- the SIFT [Lowe 2004a] (Scale-invariant Feature Transform) detector is based on the DoG detector, with additional constraints to discard low contrast points and points along edges; it is invariant to scale and rotation changes;

- the SURF [Bay 2008] (Speeded Up Robust Features) feature detector is also based on the Hessian matrix, but it approximates the Gaussian second-order partial derivative filters by box filters; integral images are used to compute the responses of the box filters, which make the SURF feature detector a fast implementation; it is also robust to scale and rotation changes;

- FAST [Rosten 2010] (Features from Accelerated Segment Test) is a corner detector based on comparing the value of a central pixel with those of pixels on a circle around the center pixel; it is very efficient, it is rotation-invariant and it also has an extension for scale-invariance;

- BRISK [Leutenegger 2011] (Binary Robust Invariant Scalable Keypoints) is a corner detector based on FAST, with an added capability to determine the accurate scale of a keypoint; it is reported to be even an order of magnitude faster than SURF in some cases;

- Good Features to Track (GFTT) employs the Harris corner detector, with additional constraints related to corner strength and distance between neighboring corners [Shi 1994a]; it is useful in videos, for motion tracking applications;

Properties which are often desired for feature detectors are invariance (or at least good robustness) to various image transformations: scale variations, rotations, affine deformations (such as from perspective changes).

For object category recognition, it has been shown in [Nowak 2006] that using dense features outperforms features from detectors, because many more features can be obtained from a dense grid than from detectors. Bag of Words models work better when a large quantity of features is available, as the BoW histogram of visual words is more populated and better represents the image content from a statistic point of view. Another reason for the improved performance of dense grids is that they insure a uniform coverage of the image, whereas a feature detector may focus only on certain zones where the detector gives a strong response. On the other hand, there is a possibility that features returned by feature detectors are more representative, as they are localized on spatial discontinuities; this is linked with the feature *description* mechanisms from the next step of the BoW processign chain, which usually deal with intensity gradients (the spatial appearance of such discontinuities).

Feature detectors can be impacted by degradations such as motion blur, compression artifacts or high noise levels, but in any case, they have a higher degree of repeatability than dense grids. However, the precise localization and repeatability of selected points plays only a secondary role in BoW performance. Nevertheless, it was shown in [Everingham 2010b] that combining interest points and dense grids yields an even better performance. From this remark, a hybrid approach has spawned for selecting features,

called *dense interest points*: the starting point is a regular grid of points, but the positions of the points are slightly shifted in the grid so as to maximize a cornerness measure [Tuytelaars 2010]. This approach exploits the benefits of both feature selection methods: the image is densely and uniformly covered as with dense grids, but the features are more localized on spatial discontinuities as with feature detectors, where feature descriptors based on intensity changes are more relevant.

2.3.4.2 Descriptors for local image patches

After choosing the image features to describe, an image patch is taken around each feature point. For each image patch, a descriptor is computed, and it is these descriptors which we discuss here. Like in the case of feature detectors, it is often desired of descriptors to be invariant (or robust, if true invariance is not possible) to image deformations (rotations, scale changes, affine transformations). If the descriptor itself is not invariant or robust to these transformations, but the feature detector was able to determine the scale and orientation, the image patch around the feature can be transformed so as to normalize scale, orientation and/or affine deformation. Here are a few of the most popular image patch (feature) descriptors and some more recent ones:

The SIFT descriptor: It is based on histograms of oriented gradients (HOG) (note that SIFT is both a feature detector and a descriptor). A 16x16 pixel neighborhood (image patch) is considered around the feature; in the case of approaches examining multiple scales, larger or smaller patches can be taken (but these patches will be re-scaled to the default value of 16x16 to allow computing the descriptor). Afterwards, the intensity gradients along x and y directions are computed, giving information about the module and orientation of the gradient vector in each pixel. The 16x16 patch is divided into 4x4 pixels smaller patches. On each smaller patch, a histogram of gradient orientations is computed (with orientation quantized on multiples of 45°), as in Figure 2.4, by summing the gradient modules that fall on each of the 8 orientations. An additional Gaussian weighting function is applied to give more weight to gradients closer to the feature point (the centre of the 16x16 patch). The 16 histograms of gradient orientations from the 4x4 subpatches are concatenated to form the SIFT descriptor of the patch (which has 16x8 = 128 dimensions)[Lowe 2004b].

The SIFT descriptor is conceived for describing grayscale image patches. When dealing with color, OpponentSIFT can be used, which consists in transforming the RGB image into an opponent color space, then computing the grayscale 128-dim. SIFT descriptor on each of the three color planes of the opponent color space, and in the end, concatenating the three descriptors into a 384-dimensional OpponentSIFT descriptor.

The SURF descriptor: SURF, too, is not only a detector, but also a descriptor. It is based on computing Haar wavelet responses on 4x4 square sub-regions of the image patch. The horizontal and vertical haar wavelet responses, d_x and d_y, are computed at 5x5 pixels regularly sampled points, for each square sub-region. Each sub-region is represented by a vector $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ formed of the sums of d_x and d_y and their absolute values inside the sub-region, weighed by a Gaussian centered on the feature point.

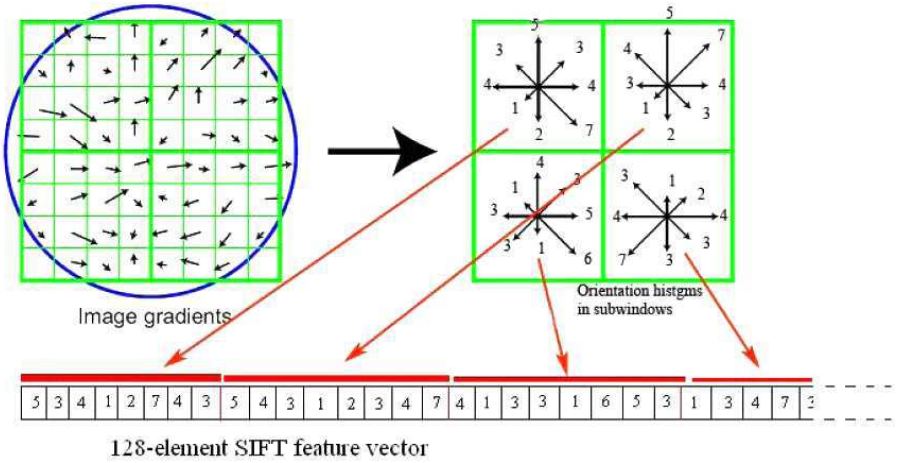


Figure 2.4: Computing the SIFT descriptor. For simplicity, only an 8x8 image patch is represented instead of 16x16. The module and orientation of the intensity gradient in each pixel are computed (left). 4x4 pixels subpatches are considered (left), and histograms of gradient orientations are computed on each subpatch (right) with 8 bins for orientation, with the gradient modules weighted by a gaussian function (represented by the blue circle). In the end, the histograms from the subpatches are concatenated to produce the 128-dimensional SIFT descriptor. Image source: `https://picasaweb.google.com/lh/photo/vyaYFzPsGz6RzldJnvEaDQ` adapted from [Lowe 2004b].

The v vectors from all sub-regions are concatenated to produce the 64-dimensional SURF descriptor [Bay 2008].

There is also a 128-dimensional version of SURF, in which the sums for d_x and $|d_x|$ are computed separately for $d_y < 0$ and for $d_y \geq 0$; similarly, the sums for d_y and $|d_y|$ are computed separately for $d_x < 0$ and for $d_x \geq 0$. The authors report that this version is more discriminative and not much slower to compute, but slower to match because of the increased length [Bay 2008].

BRIEF, Binary Robust Independent Elementary Features, is a feature descriptor composed as a binary string computed using intensity difference tests. It is reported to give similar recognition performance as SURF, but with much shorter computation times. The descriptor is very compact because the bits are independent (a dimensionality reduction step as in PCA-SIFT is not needed), and combined with the use of the Hamming distance for descriptor comparisons (instead of the slower L_2 norm for SIFT-like descriptors), this gives very low descriptor matching times. The BRIEF descriptor itself is not invariant to scale and rotation changes, but this can be compensated for by the feature detector (which can detect the scale and orientation of the feature and choose the local image patch to describe accordingly); however, adding invariance to rotation reduces recognition performance, because the descriptor becomes less discriminant [Calonder 2010].

ORB, Oriented FAST and Rotated BRIEF, improves the FAST keypoint detector by adding a method for determining orientation, and also improves the BRIEF descriptor by making it rotation-aware [Rublee 2011].

BRISK, Binary Robust Invariant Scalable Keypoints, is a feature detector, descriptor and matcher. The detector is FAST-based, while the descriptor, from the same family as BRIEF, is composed as a binary string, formed by concatenating the results of brightness comparison tests. Only a limited number of points is used for brigtness comparisons, but in a specific sampling pattern, which can also give information about the orientation of the keypoint. Orientation information is then used to achieve rotation invariance. It is significantly faster than SIFT and SURF, while giving similar matching performance [Leutenegger 2011].

FREAK, Fast Retina Keypoint, is also a binary string descriptor based on comparing pairs of points around the feature point, similar to BRISK. Points for comparisons are taken on a circular pattern, with a higher density of points towards the centre. Gaussian smoothing is done for robustness to noise, with larger kernels farther away from the centre. This resembles the behaviour of the human retina, which has a higher resolution at the centre and whose output action potentials resemble the intensity comparisons of FREAK. The pairs of points whose comparisons form the descriptor are chosen so that they bring the most amount of information and have minimal correlation, and they are ordered according to their information contribution. This ordering has resulted in a pair comparison pattern which resembles a coarse-to-fine analysis, similar to the human retina. This ordering of

pairs results in an ordering of bits in the resulting descriptor, from most important to less important, which can be used to accelerate the descriptor matching step, by rejecting keypoints whose first 16 bytes are too different. Again, this is in tone with the coarse-to-fine idea, as the first bytes represent coarse spatial information. In feature matching tests, it outperformed SIFT, SURF and BRISK in terms of descriptor extraction time, matching time and number of correct matches [Ortiz 2012].

2.3.5 Descriptors for action recognition

When it comes to action recognition, descriptors based purely on spatial appearance are no longer informative enough. It becomes necessary to use descriptors that capture motion information, or that blend spatial and motion information together (*spatio-temporal* descriptors).

One of the first descriptors applied for action recognition was the Motion History Image (MHI) [Bobick 2001], which labels each pixel as having/not having motion (or as in how many frames did it experience motion recently); template matching is then used to recognize the action. However, the method cannot be applied in situations with camera motion or with cluttered scenes, being sensitive to parasitic movement and to occlusion. Other methods to recognize actions are based on detecting and representing the motion of human body parts, such as [Brendel 2010] and [Tran 2012].

Although not specifically dedicated to action recognition, we can also mention the work of [Tanase 2013], which extends the Bag of Words model by separating local features into two categories: features belonging to the (static) background and features corresponding to foreground objects in motion. Two histograms of visual words are thus constructed, one for static features and one for moving features, thereby separating information corresponding to the static and to the moving parts of a video. The authors then choose to concatenate the two histograms to form the video descriptor, but other strategies of exploiting these two types of information could be envisaged. For example, the BoW histogram of moving features can be used to detect objects that are usually in motion, while the other BoW histogram can be used for objects that are normally static. The results from the histogram of static features can then be considered as context information, and can be used to reinforce the results from the moving features histogram.

An interesting approach for action recognition is presented in [Rosales 1999]. Objects of interest (persons) are segmented using a continuously-updated background model. The object bounding boxes are then tracked across frames using Extended Kalman Filters, with adaptations that allow predicting and detecting occlusions (in order not to interrupt tracking when a short-time occlusion occurs). Tracking allows to align each object across frames and to construct object-centric representations using Motion History Images, from which the action can be recognized. The system is interesting because it employs feedback loops that improve processing on lower stages based on results from higher stages, treating in a unified manner the problems of tracking, trajectory estimation and action recognition. However, although the approach is well-suited for video surveillance contexts with a fixed camera and uncluttered scenes, unfortunately it would not work in TRECVid SIN, because the setting is too diverse and uncontrolled.

In general, methods that try to characterize video volumes as a whole are affected by occlusion and clutter (something passing in front of the action of interest changes its appearance). Local approaches, on the other hand, describe only small bits of videos (video features that are local in space and time) instead of large video volumes. They then use an aggregation strategy, such as the Bag of Words model, to construct the description of a larger video volume based on its small parts. As in the case of purely spatial descriptors, the BoW model ignores spatial and temporal relations. There exist models that also encode spatial and temporal relations, such as spatio-temporal pyramidal representations (but these impose a rigid definition of the space-time division) [Laptev 2008], or Actom Sequence Models that encode the temporal succession of action elements (action atoms, *actoms*) [Gaidon 2011].

There is a high diversity of spatio-temporal descriptors, but it can be noted that many of them describe the spatial appearance component with the aid of descriptors based on Histograms of Oriented Gradients (HOG), SIFT being a good example of a HOG-based descriptor. As for the motion component, the optical flow is often used, indicating the direction of motion in every pixel, which can be used to construct descriptors such as Histograms of Optical Flow (HOF). Motion can also be represented on longer time intervals by tracking the motion of points across many frames and constructing trajectories. Spatial appearance and motion can also be described at the same time, such as with HOG-3D descriptors based on gradient orientations in 3D (space-time) [Kläser 2012]. We will give some examples of spatio-temporal descriptors below, concentrating on local representations, as these are more appropriate for the diverse, unconstrained TRECVid context.

2.3.5.1 Spatio-temporal interest points

Some approaches detect local features that are distinctive not only in space, but also in time, *spatio-temporal interest points*, and then describe the spatio-temporal neighborhoods of these features [Laptev 2003, Ke 2005, Dollár 2005, Niebles 2008].

For example, in [Laptev 2003], spatio-temporal interest points are detected using an extension of the Harris corner detector to 3 dimensions (2D space + time). This gives features that are at the same time spatial corners, and experience a non-constant motion such as an abrupt change in motion direction. A spatio-temporal cuboid, as the ones in Figure 2.5, is then described with one or more descriptors, and the results are fed into a Bag of Words model for action recognition.

The approach is extended in [Laptev 2007] to make it invariant to the local constant-velocity component of motion; a spatio-temporal cuboid looks different when it undergoes acceleration around a zero local motion, or when the acceleration takes place while the spatial corner was undergoing uniform translation (the uniform translation will skew the spatio-temporal neighborhood). This brings robustness to camera motion or uniform object translation, at the cost of losing discriminative power in simpler scenarios without camera motion.

For describing spatio-temporal cuboids, the following types of descriptors were proposed in [Laptev 2007]:

- N-jets and multi-scale N-jets, which are spatio-temporal Gaussian derivatives up to

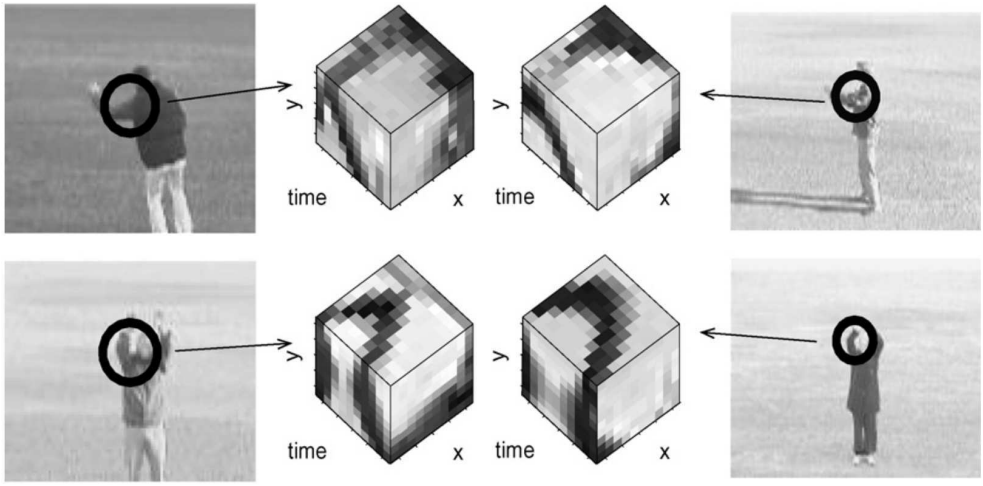


Figure 2.5: Examples of spatio-temporal video features detected by an extension to 3D of the Harris corner detector, with the spatio-temporal cuboids that will be described. Image credit: [Laptev 2007]

order N of the cuboid;

- histograms of first-order partial derivatives (intensity gradients in the spatio-temporal domain);
- histograms of optical flow;

Histogram descriptors were explored in both a position-independent way (a single histogram for the entire cuboid) or in position-dependent ways (the cuboid was divided according to a spatio-temporal grid and histograms were computed on the elements of the grid and then concatenated). Principal Component Analysis was also used optionally for dimensionality reduction. Upon testing on the KTH dataset, the ranking of the descriptors varied depending on whether or not position dependent or independent histograms were used, and whether or not PCA was used, but it can be said that generally, histograms of spatio-temporal gradients and of optical flow performed better than N-jets. Also, position-dependent histograms performed better than position-independent ones, because they described the cuboids in more detail and were thus more discriminative [Laptev 2007].

Histograms of Oriented Gradients (HOG) and Histograms of Optical Flow (HOG) were used to describe cuboids extracted from Hollywood movies in [Laptev 2008]. HOF performed better than HOG, but a combination of the two was shown to outperform both.

2.3.5.2 MoSIFT

The Motion Scale Invariant Feature Transform (MoSIFT) is a detector and descriptor for local video features that combines spatial appearance and motion information. The classical 2D SIFT detector is used to detect spatial features in he video frames. Afterwards, only spatial features that also experience significant optical flow are kept, discarding features that do not have enough motion.

For the description step, spatial appearance is described using the classical SIFT descriptor. But SIFT is made from histograms of oriented gradients, and the optical flow in a pixel also has a magnitude and an orientation, just like the intensity gradient. Therefore, a SIFT-like descriptor can be constructed from the optical flow field in the same manner as it is constructed from the image intensity gradient field. The static appearance part is adapted for rotation invariance, but the motion appearance part is not, because it is important to keep the motion direction unaltered as it constitutes an important cue for action recognition. The spatial appearance SIFT vector and the motion SIFT-like vector are concatenated to produce the 256-dimensional MoSIFT feature descriptor. A BoW strategy can then be used to aggregate the local features.

The descriptor was shown to outperform approaches based on spatio-temporal interest points on the KTH dataset, and it also outperformed 3D Histograms of Oriented Gradients on the TRECVid 2008 Surveillance Event Detection task [Chen 2009].

2.3.5.3 Trajectories of tracked points

Trajectories contain important information about motion in the video. Object centroids can be tracked and their motion described, although this does not give a lot of information for action recognition. Tracking body parts can give more information, as many human actions are characterized by a succession of body parts positions. Or either dense or sparse trajectories, not necessarily from body parts, can be constructed and described. Tracking local features (either from a dense grid or sparse) presents an advantage in unconstrained scenarios, because they are less sensitive to occlusion, viewpoint variations, variability of the objects/persons performing the actions and variability of context.

In [Vrigkas 2013], dense optical flow is computed on every frame of the video, from which *motion curves* (trajectories) are extracted. Motion curves belonging to the background are eliminated, based on whether or not the total optical flow along the curve is large enough (insufficient motion characterizes a background feature). Trajectories of varying lengths are allowed, and the Longest Common Sub-Sequence is used to compare two trajectories. The approach worked very well on the KTH dataset, with an accuracy of 96,71%.

Computing dense optical flow fields is computationally expensive, but computing optical flow for a small set of keypoints is much faster. Therefore, [Matikainen 2009] proposes to detect features with the Good Features To Track detector [Shi 1994b], and track them across frames using a classical Kanade-Lucas-Tomasi (KLT) tracker [Birchfield 2007]. These trajectory elements, called *trajectons*, are described using concatenated vectors of spatial derivatives (displacements in x and y from one frame to the next), to which an affine model of the local deformation along the trajectory can be added. The model was not made robust to scale variations, neither spatial nor temporal, and the fact that a motion can be captured starting from different moments was dealt with by considering the same trajectory several times, but with shifted starting and ending moments. The trajectories are fed into a BoW model, and Support Vector Machines with linear kernels are used for classification (LIBSVM, [Chang 2001]).

Trajectons were again used in [Wu 2011], where dense trajectories of points are ex-

tracted. This time, camera motion is dealt with by decomposing trajectories into their camera-induced component and object (person) induced component, without the need to perform an alignment of video frames. The approach gave 95,7% precison on the KTH dataset.

In [Wang 2011], dense trajectories are constructed by tracking points from a dense grid via dense optical flow fields. A fixed length of 15 frames is used for all trajectories (called *tracklets*) because the authors noted that representing trajectories at multiple temporal scales does not improve their results. The shape of a trajectory is encoded with a normalized vector of displacements. Additionally, trajectory-aligned descriptors are also computed: the local spatial appearance around a tracked point is represented with a Histogram of Oriented Gradients (HOG) averaged across the 15 frames, while local motion around the tracked point is represented with a Histogram of Optical Flow (HOF). A third trajectory-aligned descriptor is the Motion Boundary Histogram (MBH): spatial derivatives of the horizontal and vertical components of the optical flow are computed, and then histograms of orientations are constructed for these derivatives, giving rise to the MBH. Because MBH do not characterize the optical flow itself, but the relative motion between adjacent pixels, they are robust to camera motion. Dense trajectories have an advantage over tracking sparse points, because many more features are fed into the model, which is one of the reasons why the approach performs well on a variety of action recognition datasets (94,2% on KTH) [Wang 2011].

Similar dense trajectories and trajectory descriptors as in [Wang 2011] are used in [Jiang 2012], with the following differences: camera motion compensation is done by clustering motion patterns and describing trajectories *relative* to the three most important motion patterns, and relations between trajectories are encoded by considering trajectory pairs and describing the relative positions and relative motions of the members of the pair with respect to each other.

Instead of using dense trajectories, [Ballas 2011] employs a Difference of Gaussians detector to detect sparse points in frames, tracking being performed by matching SIFT descriptors of keypoints from consecutive frames. Trajectories are described using histograms of motion directions (first-order statistics), Markov Stationary Features (second-order statistics) and histograms of acceleration directions (for robustness to the uniform translation component of motion). Replacing displacement vectors with histograms of displacements gives robustness to the exact moment of the beginning of an action. Spatial appearance along the trajectory is also represented using the average SIFT descriptor along the tracked point.

In TRECVid, trajectories are employed mainly for the Surveillance Event Detection task. For example, [Xu 2012] use particle trajectories extracted directly from the MPEG stream, and [Little 2012] use a KLT tracker on Harris corners to construct 15-frame trajectories with HOG-HOF and MBH descriptors (as in [Wang 2011]) which are fed into a BoW model and classified with a SVM with a RBF (Radial Basis Function) kernel. There are few contributions employing trajectories in the Semantic Indexing task, because most of the concepts are not necessarily related to motion; [Ballas 2012a] have contributed BoW descriptors based on extracting dense trajectories and characterizing them with displacement vectors and histograms of displacement directions, as in [Ballas 2011], as part of their

participation in the TRECVid SIN task.

2.4 Information fusion strategies

Most often, combining information from several descriptors improves the correct recognition rates of semantic concepts. *Early fusions* combine descriptors before the classification step, while *late fusions* combine the outputs of supervised classifiers.

Early fusions can be as simple as concatenating two or more multidimensional descriptors. However, there are some issues with such a method: descriptor dimensions may have values in different ranges (causing certain dimensions to dominate the others) and they may also have varying numbers of dimensions (the descriptor with more dimensions dominates the others); additionally, descriptors may have varying importances for a certain concept, all of this requiring a careful weighting of the inputs. In [Zhang 2011], early fusion is performed by computing the distance between two videos as a weighted average of distances between different descriptors. In [Wang 2011], a multi-channel approach is used to combine a trajectory descriptor (shifts from one frame to the next) and trajectory-aligned descriptors (histograms of oriented gradients, histograms of optical flow, motion boundary histograms) as input for a SVM with a χ^2 kernel, by measuring the distance between videos as the average of distances between channels (input descriptors).

Late fusions can be as simple as averaging the output scores from classifiers based on different descriptors, or can be more complex, taking into account the inter-dependencies of classification scores from different sources like it is done with Choquet's integral [Cliville 2004]. An additional level of supervised classification can also be trained on the set of output scores from the previous classifiers, however this can lead to over-fitting which degrades results, and averaging output scores generally gives results just as good (or better) with less computational cost. In [Zhang 2011], late fusion is done by averaging output scores from classifiers applied on different descriptors, but in their approach, early fusion performs better than late fusion. They also experiment with a combination of early and late fusion (double fusion) which was shown to generally outperform both the early and late fusion. In general, late fusions perform best when the descriptors being fused are complementary, as it was shown by [Ng 2000].

There can also be intermediates between early fusions and late fusions. With regard to SVM classifiers, Multiple Kernel Learning (MKL) can be considered a sort of intermediate fusion. Instead of using a single kernel function for the SVM, several kernels can be combined (either working on the same data or on different data) to improve classification results [Gönen 2011]. For example, the multi-channel approach in [Wang 2011] can be regarded as a MKL problem.

Fusion strategies for detecting a concept can also concern themselves with how to deal with data imbalance problems (such as in TRECVid SIN, where most of the concepts have many more negative labeled examples than positives) or which features or descriptors are more relevant for that concept. [Zhang 2011] use a Sequential Boosting SVM inspired from bagging and boosting approaches. Bagging [Breiman 1996] means splitting the training database into several subparts (when there are many more training negatives than positives,

the positives may be kept common to all subparts) and training a classifier on each subpart; at recognition, the outputs from those classifiers are combined (averaged) to improve the result. Boosting strategies such as *AdaBoost* [Freund 1997, Schapire 1999] train a strong classifier by combining (through weighted average) results from many weak classifiers.

More specifically, *AdaBoost* is an iterative algorithm that works in the following manner: it starts by choosing the best weak classifier from the set of weak classifiers and applying it on a validation dataset, and including this weak classifier in the strong classifier. The misclassified examples by the weak classifier from this step are given more weight for the next iteration. At the next iteration, the weak classifier that minimizes the global error (the weighted sum of the errors for each example) is selected and added to the strong classifier. Again, weights of misclassified examples are increased and the process is repeated. Updating weights in this manner makes the next weak classifier focus on the examples that were incorrectly classified in the previous step. A very successful application of AdaBoost is in face detection, where weak classifiers based on simple Haar-like features are combined into a powerful (and fast) face detector [Viola 2004]. In TRECVid, late fusions based on AdaBoost have been used in [Cai 2007, Wu 2003, Tang 2008] among others.

In [Cao 2012], sets of classification scores are generated from a large number of video descriptors on which different classification algorithms are applied, and the classifier that yields the best result for each descriptor is retained and the resulting experts are combined in a late fusion approach.

A similar fusion context is described in [Strat 2012b], where three late fusion approaches for TRECVid SIN are compared. The fusion inputs are classification scores from two types of supervised classifiers [Ballas 2012b] applied on a battery of various descriptors (color, texture, BoW of local features, audio etc.). Since most of the descriptors are present in several versions (such as different vocabulary sizes for BoW descriptors), some of the descriptors are highly correlated. Because of this, all three approaches share a common idea: first, a descriptor *clustering stage* groups score sets into families based on similarity (such as grouping all scores from BoW descriptors); second, an *intra-cluster fusion* stage fuses the descriptors in each family; third, an *inter-cluster fusion* stage fuses the results from all families; score normalization steps may be included optionally between stages. One of the fusion methods uses a manual hierarchical grouping of input scores based on the type of descriptor and supervised classifier employed, while the two other approaches determine similarities automatically. The automatic approach contributed by us will be discussed in more detail in Chapter 5.

2.5 Proposed improvements

As stated previously, our semantic indexing experiments of video databases are conducted mostly as part of the IRIM group, which has put in place a well-performing framework [Ballas 2012b] for semantic concept detection in videos (see Section 2.2 for details). We therefore adopt this same framework in our experiments, because it already has put into place various tools dedicated to large-scale video indexing, that would have otherwise taken much too long to develop ourselves within the time span of a thesis, and would

have also demanded high computational resources that are not so easily available (access to the GRID5000 computing cluster). The supervised classification stage (see Figure 2.1) is the most essential tool that we use from IRIM to be able to perform semantic indexing on the TRECVid dataset, because training and applying supervised classifiers on such a large dataset requires significant computational power. We also take advantage of the optional descriptor optimizations, temporal re-scoring and conceptual feedback tools, since they can improve concept detection results. The availability of tools for determining average precisions for a set of classification scores also comes in very handy, as we can quickly get feedback related to the performance of our methods and adjust parameters accordingly.

Regarding our proposed contribution within this framework, based on the state-of-the-art that we have just done, we have identified the following needs for the problem of indexing generic videos with generic semantic tags:

- a need for improved *spatio-temporal descriptors* of video content, that would give better concept detection performance without excessive computational demands, and that would work not only with static concepts, neither just with motion-related concepts, but with *very generic* semantic concepts; this would allow indexing video databases with rich sets of semantic tags, that would in turn allow a user to formulate complex and diverse search queries and still obtain good results;

- it is unlikely that a single descriptor can fulfil the requirements above, therefore a set of complementary descriptors, some focusing on spatial aspects, some on temporal (motion) aspects, and even some that try to blend spatial and temporal information would be more suited; this brings us to the second identified need, that in order to benefit from the joint descriptor set, *information fusion strategies* adapted to the application framework and to the available descriptors have to be implemented;

The way we address these needs constitutes the three-fold contribution of this thesis:

- For generating improved, generic *spatio-temporal descriptors*, we build our work upon the classical Bag of Words framework utilizing SIFT or SURF local features. This framework already gives good results on databases of static images, and its application to video databses also performs good for concepts associated with particular spatial local features. Our contribution is to improve the concept detection performance of this framework and at the same time make it more generic, capable of encoding spatio-temporal information, all of this without a significant computational overhead. We do this by *preprocessing videos with a model of the human retina* [Benoit 2010] before extracting SIFT/SURF local features, as it will be seen in Chapter 3.

- Also with the goal of enriching spatio-temporal descriptions, we go one step further towards even more temporally-oriented descriptors, in the form of *Bags of Words of trajectories of tracked points*. We remain in the same Bag of Words framework (we just work with a different type of features, trajectories instead of local spatial SIFT/SURF signatures), because BoW has also been shown to work with motion features. The BoW model is simple to manage and does not require complicated

annotations for training (such as moments when an action starts and stops inside a video), which are unavailable on the datasets with which we experiment. An additional reason for keeping the BoW model is that we can reuse the same BoW tools developed for Chapter 3 and the same supervised classification stage from the IRIM group, therefore speeding up the development phase; this goes well with our goal of generic tools for video indexing, as it does not require yet another model to be developed and optimized. Additionally, this fulfils a dataset-specific need, because within TRECVid SIN, there are very few contributions utilizing motion descriptors, as they require vast computational resources on such a large database. We take advantage of our access to the MUST computing center of the University of Savoie to compute a rich set of trajectory BoW descriptors, which are detailed in Chapter 4.

- Retina-enhanced SIFT/SURF BoW descriptors and trajectory BoW descriptors constitute a set of complementary descriptors. In addition to these, for the TRECVid dataset, the IRIM group has made available to its members classification scores from a rich battery of additional diverse descriptors (color, texture, BoW of local features, audio etc.), creating opportunities for late fusion experiments. Because there are several tens of descriptors (therefore several tens of score sets, too) contributed by various teams and because each semantic concept has a different optimal combination of descriptors for the late fusion, we explore *automatic late fusion strategies* in Chapter 5.

Now that we have stated the lines of research of this thesis, we will give in the next section a short presentation of some popular video datasets used by the research community for comparing semantic concept detection methods. We will use such datasets (especially the TRECVid SIN datasets of various editions of the challenge) to show how our proposed methods can bring improvements compared to the state of the art.

2.6 Standard datasets for concept detection

In order to evaluate the detection of semantic concepts (objects, actions, scene types, movie genres, characters etc.) in images or videos, and to give a basis for comparing different approaches, standard datasets have been created and made publicly available. For static images, some examples are the Caltech 101 and Caltech 256 datasets [Fei-Fei 2007, Griffin 2007] for object recognition, and the Pascal VOC Challenge [Everingham 2010a] for object detection and recognition. For videos, some examples are the KTH dataset [Schuldt 2004] for action recognition in simple scenarios, the Hollywood 2 dataset [Marszalek 2009] for detecting and recognizing actions in movies, and the annual TRECVid challenges [Over 2012] that deal with very diverse semantic concepts (not just actions) in unconstrained videos.

2.6.1 The KTH human action dataset

This dataset consists of 6 actions (boxing, handclapping, handwaving, jogging, running, walking) performed by 25 people in 4 types of situations (outdoors, outdoors with scale

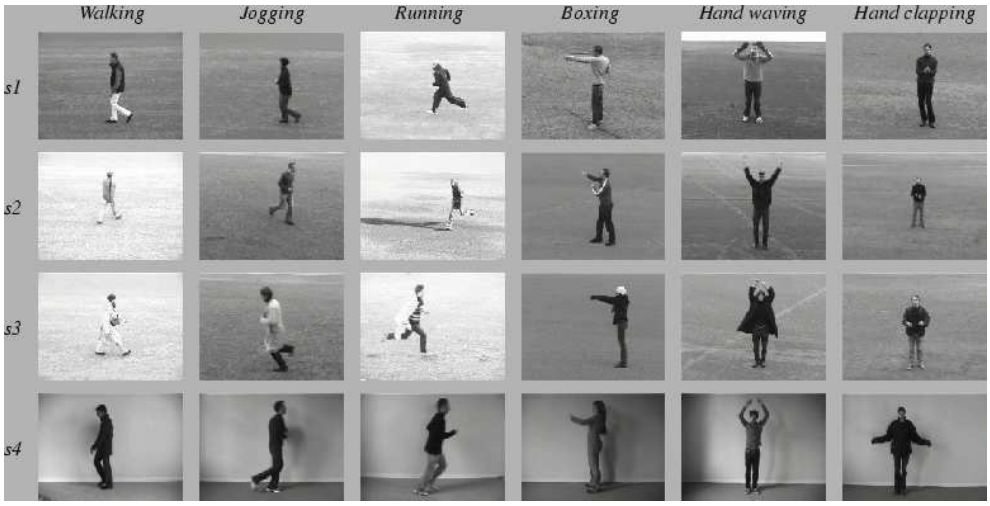


Figure 2.6: Example frames from the KTH dataset [Schuldt 2004]. Image source: http://www.nada.kth.se/cvap/actions/

variation, outdoors with different clothes and indoors), as in Figure 2.6. Each video file contains exactly one action, done by one person, in one situation, with the action being performed repetitively in video. The goal is to determine, for each video, which of the 6 actions is performed.

Because the actions and the situations are relatively simple, and because the actions do not need to be detected/localized (we already know that each video contains one of the actions, we just need to determine which one), it is easy with the current state-of-the-art to obtain good results, with precisions above 90% [Wang 2011, Laptev 2007].

2.6.2 The Hollywood 2 human actions and scenes dataset

This dataset contains short fragments from commercial movies. There is one part of the database dedicated to human actions and a second part dedicated to scene types. There are 12 classes of human actions and 10 classes of scenes distributed over 3669 video clips and approximately 20.1 hours of video in total[4] [Marszalek 2009].

Concerning the action part, each video sample contains at least one action (it sometimes contains more actions), and the action(s) do not necessarily occupy the entire temporal duration of the video sample. Also, there might be movie cuts during the video sample. Because of the much less constrained experimental conditions, this dataset is more challenging than the KTH dataset.

Concerning the scene types, there are 2 exterior scenes (House, Road) and 8 interior scenes (Bedroom, Car, Hotel, Kitchen, Living room, Office, Restaurant, Shop). Scene types allow improving the detection of actions by associating actions with their plauzible contexts [Marszalek 2009].

[4] http://www.di.ens.fr/~laptev/actions/hollywood2/

Figure 2.7: Example frames from the Hollywood 2 action dataset [Marszalek 2009] depicting the 12 action categories: Answering a phone, Driving a car, Eating, Fighting, Getting out of a car, Handshake, Hugging, Kissing, Running, Sitting down, Standing up. Image source: `http://www.di.ens.fr/~laptev/actions/hollywood2/`

2.6.3 The TRECVid challenge: Semantic Indexing task

TRECVid [Over 2012] is an annual challenge sponsored by NIST[5], with the goal of encouraging research in information retrieval by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results[6].

TRECVid proposes several tasks to participants, of which our team was interested by the *Semantic Indexing (SIN) task*. The dataset associated with this task, in the 2013 edition, is composed of $\approx$ 800000 short video fragments (called *shots*) of lengths varying between a few seconds to a few tens of seconds. Associated to these shots is a list of 346 various semantic concepts, which can be objects (Bus, Tree, Car, Telephone, Chair), actions (Singing, Eating, Handshaking), situations/scene types (Waterscape, Indoor, Kitchen, Construction site), abstract concepts (Science/technology), types of people (Corporate leader, Female person, Asian people, Government leader) or even specific people (Hu Jintao, Donald Rumsfeld). These concepts may or may not be present in a shot. The dataset is split in half, the first part for developing and fine-tuning concept detection algorithms, and the second part for testing and evaluating the performances of the task participants.

The goal of the challenge is that, from the testing half of the dataset, for each semantic concept, participants should return a ranked list of max. 2000 shots that are the most likely to contain the semantic concept in question (just like when using a search engine). The quality of the returned lists for each concept is evaluated (how well the relevant shots for that concept are concentrated towards the beginning of the list) by NIST and participants are then communicated their performances.

The evaluation measure used in TRECVid SIN is the *mean inferred average precision (infAP)* [Yilmaz 2006, Yilmaz 2008]. Basically, for a particular concept, the "average precision" is the average of the precisions obtained for various recall rates, a measure of how well the true positives of this concept are concentrated towards the beginning of the list of 2000 shots. The "mean average precision" is the mean of the previous result over all concepts.

The TRECVid SIN dataset is very challenging, for the following reasons:

- The videos come from a wide array of sources, of varying quality and content. They can range from professional news studio or news footage, sports events filmed by professionals or TV shows, to amateur videos recorded with a camera phone and a lot of camera motion. They can be from various environments, such as from inside a kitchen, from outside in the street, from the beach or from an exotic location. They can be acquired in various lighting conditions, ranging from a sunny day outdoors to a dark interior of a night club.

- The large amount of concepts to detect means that it is not practical to develop a special algorithm for each concept. Instead, a generic approach is used for all concepts, but it is not easy to develop a generic system that works well-enough with every concept.

[5] National Institute of Standards and Technology
[6] http://trecvid.nist.gov/

- Many concepts are quite rare in the dataset, they may only appear in a few tens of shots out of the total ≈ 800000, which poses a problem when training concept detection algorithms (when training supervised classifiers).

- For a shot to be considered as an occurrence of a concept, it is enough that the concept is present in at least one frame of the shot. However, the annotation only says if a shot contains or does not contain a concept, but it does not say *when and where* that concept appears. This poses a challenge because we do not know which information extracted from the shot is useful and which is irrelevant for the concept in question, which makes both training the detectors (training the supervised classifiers) and evaluating the test shots (applying the classifiers) more difficult.

Because of the large size of the database, many participants do not analyze the entire video shot. Instead, they analyze only one (sometimes several) *keyframe(s)* per shot, greatly speeding-up the analysis (of course, with the risk that the concept might not be in the chosen keyframe). To this end, TRECVid also provides an official selection of keyframes for each shot [Over 2012].

Most of our experiments are performed on various editions of the TRECVid SIN task, because the large diversity of target semantic concepts and contexts in which these concepts can appear constitutes an ideal test for generic semantic indexing algorithms, which is the goal of this work. In Chapter 3 we experiment on the TRECVid SIN 2010, 2011 and 2012 editions, showing that our retinal preprocessing approach gives reproductible results across datasets. In Chapter 4, we experiment with trajectory Bag-of-Words descriptors on the 2012 edition, but we also reuse part of these descriptors for the information fusion approaches from Chapter 5 tested on the 2013 edition. As for Chapter 5, we perform experiments both on the 2012 and the 2013 editions of the dataset.

While most of the studies are done on the complex TRECVid SIN dataset, we also use the KTH dataset to validate the motion dedicated descriptors. Further studies for trajectory descriptors should be performed on other motion-dedicated datasets of intermediate complexity such as Hollywood 2, but already these two datasets, illustrating two extreme scenarii of action detection/recognition (highly-restrained versus completely unrestrained context) show the potential of our methods.

Chapter 3

Retinal preprocessing for SIFT/SURF-BoW representations

Contents

As we have seen in Section 2.3.4, SIFT/SURF BoW descriptors generally perform well in object or scene detection and recognition applications, and they are usually the best performing individual descriptors in TRECVid SIN [Over 2012]. They also scale well to large databases, which is an additional reason why the are used so often. However, they can be negatively impacted by image disturbances such as noise and compression artifacts. Moreover, they are lacking when it comes to encoding spatio-*temporal* information, which makes them less relevant for concepts related to motion.

The Human Visual System, on the other hand, exhibits certain spatio-temporal behaviours which are useful in image processing applications. For example, the human retina not only contains photosensitive cells, but also applies a series of low-level processing steps on the signals coming from these cells. These processing steps regulate the mean local luminance coming from the photoreceptors, reduce high frequency spatio-temporal noise, enforce local contrasts without increasing noise and detect moving elements [Hérault 2010].

The goal of this chapter is to take advantage of certain properties of the Human Visual System to augment SIFT/SURF BoW descriptors, by making them more robust to image degradations, more sensitive to spatial details, and also by making them sensitive to spatio-*temporal* information instead of only spatial information.

Bio-inspired models become more and more involved in computer vision. For example, in the domain of local spatial feature description, the recent FREAK feature descriptor

[Alahi 2012], or the approach from [Ali 2011], propose to replace the classical SIFT features with bio-inspired features for image representation. They allow very efficient image description, however they are not designed to be robust against classical image artifacts (noise, compression, luminance range, etc.). Compared to these approaches, we propose to enhance visual information prior to the image description step. In our approach, image description consists in extracting SIFT/SURF features on a dense grid, but any other local feature could be involved (such as FREAK).

Other, more global human visual system models have also been proposed, such as [Itti 1998, Le Meur 2006b, Redi 2011a]. They include parts of the retina and of the first visual cortex areas. These models are mostly dedicated to visual saliency analysis and can be involved in applications of visual quality perception assessment. However, they do not all support luminance range adaptation nor do they manage temporal information. In addition, their significant computational cost compromises their use in frame by frame image analysis in large video databases.

Finally, other bio-inspired models, such as [Reinhard 2005, Mantiuk 2005, Benoit 2010], focus more on the properties of the human retina. They are mainly targeted at image filtering applications, such as image compression and detail enhancement. They also have a lower computational cost. This corresponds more to our requirements, however, only [Benoit 2010] takes into account the temporal filtering properties and the effects of the peripheral vision occurring in the human retina. Regarding peripheral vision, it is presumed that it plays a role in reflex eye movements [Sprague 1965], therefore it can be useful for focusing the analysis on low-level salient areas of the visual scene.

Therefore, considering our image enhancement and salient area extraction needs, plus the low computational cost requirement, we choose *Benoit et al.*'s model [Benoit 2010] as our video preprocessing step before extracting SIFT/SURF features. This retinal model presents interesting properties for filtering out undesired image artifacts (compression artifacts, noise etc.) and gaining robustness to luminance variations. More precisely, one of the retinal outputs (called the parvocellular channel) allows the enhancement of spatial details and artifact reduction. Additionally, another output of the model (called the magnocellular channel) can be used to manage temporal information by selecting only regions of interest associated with transient information. Such transient signals consist both of low-level spatial saliency which occurs when discovering a new visual scene, and also (and in a greater degree) of motion saliency areas.

In the next section, we will explain the behaviour of the human retina model from [Benoit 2010]; the interested reader may also refer to Annex A for the inner workings of the model. Section 3.2 will show how we exploit the retina model behaviour to augment SIFT/SURF BoW descriptors.

3.1 Behaviour of the human retina model

3.1.1 The parvocellular channel

The human retina has two well-known data channels (also called pathways). The first is the *parvocellular* channel, which processes spatial details and colors. It has a high resolution

in the center of the visual field, where it constitutes the foveal vision. It normalizes colors, enhances local contrast, responds well to temporally-sustained signals, while smoothing out fast temporal variations.

An interesting property is that at the onset of a spatio-temporal event (such as "opening the eyes" to discover a new visual scene, or when an object appears or moves in the image), the retina exhibits a *transient state*. During this transient state, only low spatial frequencies are transmitted (a blurry image, but with high signal-to-noise ratio); this is because the appearance of a new object or scene is a high temporal frequency event, and the parvocellular channel attenuates spatial details at high temporal frequencies. But if the object (or the new scene) remains stationary, the parvocellular response stabilizes and the retina enters a *stable state*. During the stable state, the parvocellular channel will start to transmit (and enforce) spatial details.

This *coarse-to-fine* processing model is not unlike what happens in the Human Visual System: when examining a new scene, the retina supplies the brain with a coarse, low-resolution image, to get a general idea of the scene content; only afterwards does it supply more spatially-detailed information.

In the retinal model that we use, the parvocellular channel is implemented as a sequence of color images with enhanced spatial details, corrected colors (with respect to the color temperature), enhanced details in the shadows and also reduced noise and reduced video compression artifacts.

An example of the effect of the parvocellular channel on a video can be seen in Fig. 3.1, in which a TV presenter is talking. We present the input and parvocellular channel at respectively 5 (Fig. 3.1a,3.1c) and 40 frames (Fig. 3.1b,3.1d) after the beginning of the visual scene processing (the initialization of the retina). Depending on the temporal constants chosen for the retinal filters, the transient state usually lasts between 10 and 20 frames. The coarse-to-fine effect can be observed on the slowly moving journalist and background clouds, where details are better enhanced later, while the global mean luminance energy decreases. The clouds are barely visible in Fig. 3.1a and 3.1b, but they are more clearly visible in Fig. 3.1d, as the parvocellular channel enhances details, and the mean luminance energy has also decreased compared to Fig. 3.1c.

Regarding model limitations, following human behaviours, the parvocellular channel cannot perfectly remove all data corruption. It properly cleans the noise introduced by low quality image sensors, by filtering-out high-frequency spatio-temporal signals. Regarding compression artifacts, they cannot all be eliminated when the compression is too severe. In extreme cases, block effects are not completely cleaned, they are only smoothed. Therefore, some corrupted data is still transmitted to the next processing stage. However, output signals still benefit from the other properties of the retina.

Another limitation is that the retina also introduces a certain degree of motion blur, therefore the spatial representation of moving features from the parvocellular channel will be degraded more or less, depending on their speed. This effect can be diminished by using smaller values for the temporal constants of the retinal spatio-temporal filters, at the cost of lower noise removal.

(a) Input frame after 5 frames from start

(b) Input frame after 40 frames from start

(c) Parvocellular output after 5 frames from start

(d) Parvocellular output after 40 frames from start

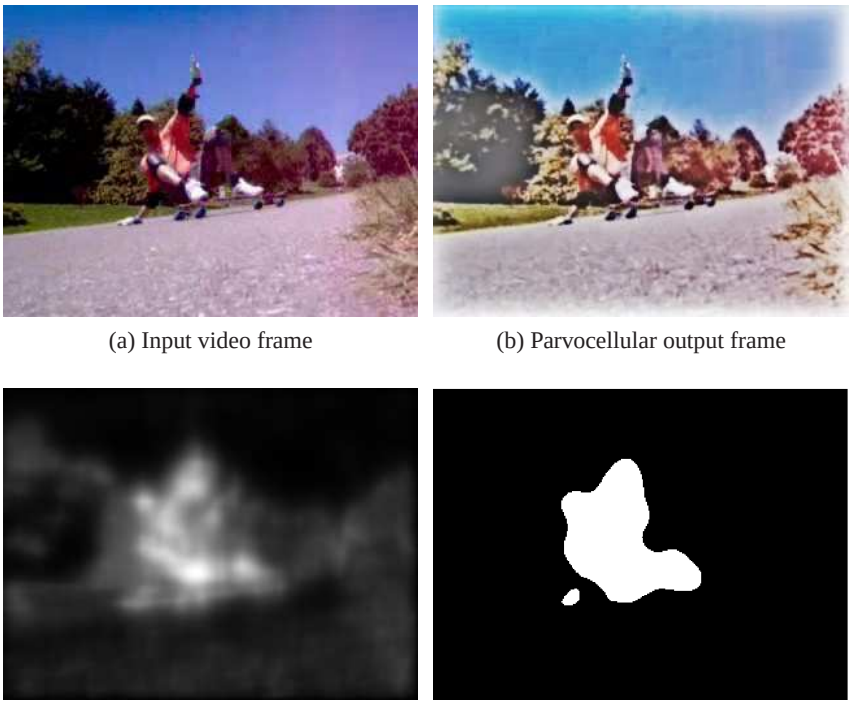
(e) Magnocellular channel output after 5 frames from start

(f) Magnocellular channel output after 40 frames from start

(g) Segmented salient blobs after 5 frames from start

(h) Segmented salient blobs after 40 frames from start

Figure 3.1: Retinal preprocessing example, after respectively 5 and 40 frames since the start of the preprocessing (the initialization of the retina). After 5 frames, the retina is still in its transient phase: the parvocellular channel passes a large amount of luminance and details are not yet enhanced too much, while the magnocellular channel fires on large spatial structures. After 40 frames, the retina is in its stable state: the parvocellular channel passes less luminance and enhances spatial details, while the magnocellular channel fires mainly on moving areas (the presenter's face). The segmented interest blobs are obtained by processing the magnocellular output: after 5 frames, we select potential spatially-interesting areas, while after 40 frames, we select mainly moving areas



(a) Input video frame (b) Parvocellular output frame

(c) Magnocellular output frame (d) Segmentation mask

Figure 3.2: Retinal outputs and area-of-interest detection on a TRECVid video.

3.1.2 The magnocellular channel

The other well-known channel is the *magnocellular* pathway. It does not distinguish between colors, but it is sensitive to spatio-temporal events. It deals mainly with the peripheral vision, giving strong responses to transient signals (quick spatio-temporal changes of light intensity, motion) and weak responses to slow-varying signals.

This channel has two interesting effects: the low spatial frequencies (from the luminance information) are briefly transmitted and a strong response is generated on spatial boundaries until the end of the retina's *transient phase*. This allows it to be used as a detector of potential *spatial* areas of interest. After the retina reaches its stable state, the response stabilizes, only firing on moving parts, therefore the channel acts as a transient area detector and more generally as a *motion* detector.

We also implement the magnocellular pathway in our model, as a sequence of gray-level images, and we use it as a low-level spatio-temporal regions of interest detector during the first seconds of a visual scene observation.

An example of the magnocellular channel response is given is Figures 3.1e and 3.1f. In Figure 3.1e, the retina is in its transient phase and the magnocellular channel passes a lot of luminance information and low spatial frequency components. In Figure 3.1f, the retina is in its stable state and the magnocellular channel only responds to moving elements (the presenter's head and her lips), which are related to motion saliency.

Regarding model limitations, severe block effects can impact transient area detection.

In such a case, using this channel as a salient area detector can lead to false detections. This drawback has to be compared with classical interest point and corner based detectors, which would also respond to such artifacts. Here, at least the temporal smoothing effect of the retina would lower the quantity of wrong interest point detections. As a result, some irrelevant image features are still transmitted, but in a lower amount that with classical approaches.

A second example of retinal outputs is given in Figure 3.2, this time only for the stable state. Here, the camera is almost static and the skater is moving. The parvocellular channel increases local contrast, generating halos visible especially around the trees, and also introduces slight motion blur, because of the spatio-temporal filtering. The magnocellular image and the associated segmentation mask highlight the skater's motion. We will describe such segmentation masks in the next section.

3.1.3 Area of interest segmentation

Not all the areas in a video are interesting for describing semantic concepts, and if a Bag of Words representation would take into account local features from uninteresting areas, the BoW descriptor would be "polluted" by irrelevant features. This motivates us to experiment with focusing the feature extraction step only on *salient* (and hopefully more relevant) areas, so that the BoW descriptors would perform better. We wish to consider both spatially-salient and temporally-salient areas.

3.1.3.1 Choice of saliency model

When talking about human visual saliency models, one usually refers to high computational cost algorithms such as [Itti 1998, Le Meur 2006b]. Such models aim at precisely modeling the retinal and visual cortex processing for identifying areas of interest. The involved retinal models are generally focused on the parvocellular channel. Such high-level saliency has been proposed in [Redi 2011a, Usman Niaz 2011] and is used to adjust the importance of features extracted from keyframes, by weighting each feature according to its saliency. In [de Carvalho Soares 2012] a fuzzy saliency model is used to weigh local features in a Bag of Words framework, in [González Díaz 2013] salient regions are used to extend object detection in egocentric vision, [Moosmann 2006] proposes saliency maps built on-line by the image classifier for object categorization, while [Vig 2012] experiment with various saliency models including recorded human eye movements for weighting local features.

In our context however, we want to investigate the use of an area of interest detector dealing with a lower-level saliency and with a *much lower computational cost*. As seen previously, the retinal properties are such that the *magnocellular channel* can be used as a *detector of low-level spatial and temporal saliency* (it detects spatio-temporal events), with low computing requirements. This leads us to design a new strategy for low-level salient blob segmentation from the magnocellular channel. This approach differs from classical human visual saliency models [Itti 1998] [Le Meur 2006b], but proposes an interesting balance between computational cost and detection performance. From a biological point of view, low-level visual processing occurring before the visual cortex levels has long been

presumed to play a role in reflex eye movements. This type of saliency is supposed to be processed mainly at the superior colliculus level [Sprague 1965].

Therefore, following this idea, but in a much simplified version, we use the magnocellular channel as a detector of low-level spatial and temporal potential areas of interest, in a low-cost bottom-up approach. We call this "saliency" for readability, even though it is of a lower level than classical saliency models. This will enable us to gather local features only from potentially more interesting areas of the videos, as we will see later in Section 3.2.2. Compared to [Redi 2011a, Usman Niaz 2011], our algorithm simply selects features from salient areas, each of them being considered with equal importance, instead of accurately weighting each feature by its saliency value.

3.1.3.2 Segmentation algorithm

The aim of the proposed segmentation algorithm is to select areas of high local transient energy at the magnocellular output. This can either be done by simple thresholding of the magnocellular output, or through a center-surround analysis, as a difference of spatially isotropic Gaussians. However, we want our segmented areas to be stable in time, avoiding fast variations of size and shape. Also, we want to avoid accidental segmentations due to residual noise left after the previous retinal processing. Therefore, we propose to use a cascade of non-separable spatio-temporal low-pass filters with the following equation:

$$F(f_s, f_t) = \frac{1}{1 + 2\tau_s(1 - cos(2\pi f_s)) + j2\pi\tau_t f_t} \tag{3.1}$$

where f_s and f_t are respectively the spatial and temporal frequencies (expressed in fractions of the sampling frequencies) and τ_s and τ_t are respectively the spatial and temporal constants (expressed in pixels and number of frames respectively). These filters will smooth the transient energy map in space and time, allowing the extraction of stable blobs and eliminating residual noise. Their computation does not demand high resources, since each filter requires only 4 products per pixel whatever the constants τ_s and τ_t are.

In the proposed segmentation stage, 3 filters are applied on the transient energy map (squared retinal magnocellular output) and combined as described in Figure 3.3.

A first filter, F_{local}, is applied for residual noise elimination and smoothing of textured transient areas. Its spatial constant sets the minimum size of the areas to be segmented (we use a value of $\tau_{s,local} = 5$ pixels). A second filter, $F_{neighbor}$, computes the transient energy in the neighborhood of each pixel. Its spatial constant is typically 3 times larger than that of F_{local} ($\tau_{s,neighbor} = 15$ pixels). The difference between these two filters allows the local motion energy from F_{local} to be compared with the energy in its immediate neighborhood, in a center-surround approach. Therefore, a pixel (x, y) is considered as part of a strong transient area (a local maximum L_{max}, part of the segmented blob) using Equation 3.2:

$$L_{max}(x,y) = \begin{cases} 1, & \text{if } F_{local}(x,y) - F_{neighb}(x,y) > \delta_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

where δ_{max} is a threshold. Its exact value is not critical, because most of the noise has been eliminated by the retina, but δ_{max} should remain above 1% of the maximum allowed

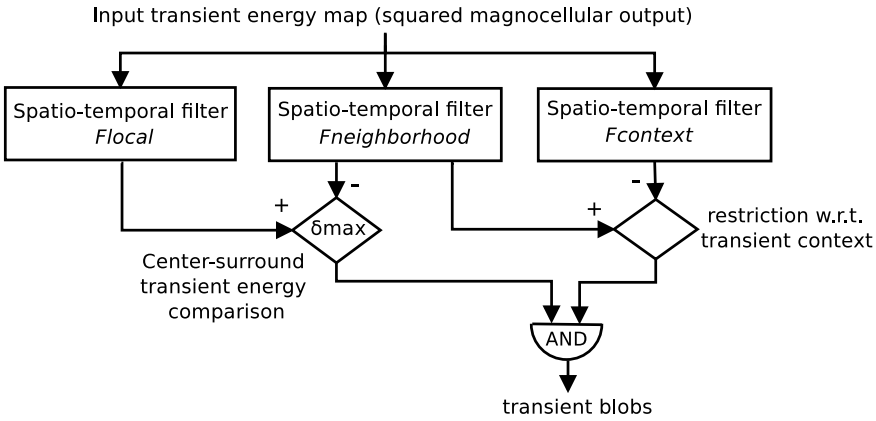


Figure 3.3: Proposed transient area segmentation method: two spatio-temporal low-pass filters allow high local energy areas to be segmentated in a center-surround approach. A third filter finalizes selection by eliminating non significant local transient energies

magnocellular energy, irrespective of the video frame size and frame rate. This way, we ensure that we select pixels with a local energy sufficiently different from the surrounding, and that we are also robust to any remaining noise (we use $\delta_{max} = 1500 > \frac{255^2}{100}$, with 255 being the maximum allowed magnocellular energy).

However, we want to segment points that stand out not only with respect to their immediate neighborhood, but also with respect to the larger local context, such as moving objects on a static background, or objects moving in a different direction than the background (and/or with a different speed). To this end we add a last constraint, with the use of a third filter, $F_{context}$, whose output indicates in which "motion context" local maximums should be identified. The spatial constant of this filter is set experimentally to 15 times the value of F_{local} (75 pixels). Then, a strong transient area is considered only if its neighborhood energy is stronger than the context (i.e. when $F_{neighbor}(x,y) > F_{context}(x,y)$), in addition to the condition from Equation 3.2. Consequently, we can select strong transient areas inside a weaker-amplitude transient context, and also isolated transient areas of different sizes and strenghts (this wouldn't have been possible when simply using an universal threshold).

Note that all these filters use the same temporal constant, $\tau_t = 1$ frame period (0.04s for 25Hz videos). It introduces a temporal smoothing effect which makes the segmented blobs more stable in time.

Figures 3.1g and 3.1h show the result of the segmentation stage. In this example, 5 frames after the start of the visual scene processing (the initialization of the retina), when the retina is still in its transient phase, the presenter and top left logo present highly energetic spatial boundaries that are automatically segmented. Afterwards, when the retina stabilizes, only the presenter is moving her head, thereby generating motion areas of interest, which is evident in the magnocellular output (a very strong response especially for the mouth), from which we extract the interesting blobs. Therefore, our algorithm does not focus on just one type of saliency: during the transient phase of the retina, spatial saliency

(a) Current input frame (b) Blobs in current frame (c) Average blobs over last 6 frames

Figure 3.4: The segmented blobs of low-level saliency are binary masks in each frame. However, averaged over several frames, blob fluctuations can be equivalent to "soft" masking.

dominates, while after the retinal response stabilizes, areas of motion saliency are recovered. This way, we hope to capture local features typical of both spatial semantic concepts *and* of motion-related concepts. Additionally, context information is also included, both during the transient phase, and later on if there is background motion.

Unlike other approaches such as [Vig 2012] which employ fuzzy saliency models, our blob segmentation algorithm results in "hard" masks for each frame: either a local feature is taken into account with a weight equal to 1, or it is completely excluded from the BoW model, with a weight equal to 0. However, because the segmented blobs are not exactly the same from one frame to the next, their moving borders will lead, on average over several frames, to soft masking, as it is illustrated in Figure 3.4.

3.2 Proposed SIFT/SURF retina-enhanced descriptors

We have seen that the retina has some interesting properties: the parvocellular channel reduces noise and enhances spatial details, and since SIFT and SURF describe local gradient vectors, they can be more reliable if extracted on the parvocellular channel; additionally, the magnocellular channel has shown itself as a good basis for detecting potentially interesting areas in the video frames, therefore the transient salient blob detector can be used to focus visual word extraction on potentially more meaningful local features. Following this idea, we propose to augment SIFT/SURF BoW descriptors by employing the human retina model, as it is described in the following.

The BoW descriptors that we create are all based on OpponentSIFT or OpponentSURF (SIFT/SURF vectors extracted from the 3 color channels of an Opponent color space) local features extracted on a dense grid, but we modify the local features extraction step by preprocessing videos with the model of the human retina, as in Figure 3.5.

We employ the retinal parvocellular and magnocellular channels in several ways, constructing two classes of descriptors:

- Keyframe based descriptors, which are similar to the classical approach (often in TRECVid, only keyframes from video shots are analyzed, to reduce computational

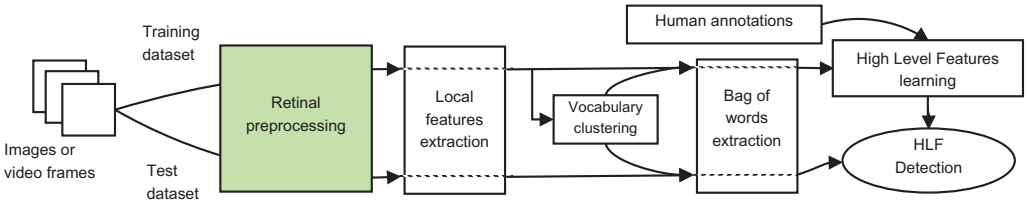


Figure 3.5: Modified BoW extraction toolchain for High-Level-Feature (HLF) (semantic concept) detection: retinal preprocessing is added before the feature extraction step.

cost, at the risk of missing the frame in which the concept appears), except the fact that we collect local features from retinal preprocessed frames. We use one keyframe per shot, from the official keyframe selection of TRECVid.

- Temporal window based descriptors with salient area masking, that accumulate local features from segmented areas of interest (approach from Section 3.1.3.2) inside a temporal window of frames (between 20-40 frames) around the keyframe.

3.2.1 Keyframe based descriptors

Keyframe based descriptors only collect local features from the video shot keyframe, but unlike the classical approach, we pre-process the keyframe with the retina.

From an implementation point of view, in order to avoid the transient response which appears when initializing the retina, we actually start the retinal processing 10-20 frames before the keyframe (after this interval, the response reaches its stable state), but we only collect features at the time of the keyframe. Recall the coarse-to-fine property described in Section 3.1.1: immediately after the initialization of the retina, the parvocellular channel still attenuates mid spatial frequencies (spatial details). However, with the considered retina setup, after waiting 10-20 frames (depending on the temporal constants used), the stable state is already reached and enhanced spatial details can be extracted with a good signal-to-noise ratio.

We propose the following keyframe based descriptors (example for SIFT-based descriptors, but SURF or any other type of local features can be used):

3.2.1.1 SIFT

We collect OpponentSIFT (we call the descriptor just “SIFT” for simplicity) features on a dense grid on the original (unprocessed) keyframe (as shown in Figure 3.6a), and we feed these features into the BoW processing chain. This serves as our reference descriptor. We recall that there is one keyframe per video shot, chosen officially by the organizers of TRECVid SIN.

3.2.1.2 SIFT retina

Instead of collecting the OpponentSIFT features from the original keyframe, we collect them from the parvocellular-processed keyframe (see Figure 3.6b). As stated previously,

we actually start retinal processing 10-20 frames before the keyframe, to give the retina time to reach its stable state, but only collect features at the time of the keyframe.

The idea behind this descriptor is that the parvocellular channel "cleans and enhances" the image, by reducing spatio-temporal noise, reducing transient compression artifacts, boosting local contrast and normalizing colors. Because image degradations are reduced and local contrast (on which SIFT is based) is improved, the SIFT descriptors of local features should be cleaner, resulting in a better BoW description.

3.2.1.3 SIFT multichannel

The OpponentSIFT signature of a local feature from the parvocellular channel only gives spatial appearance information. But we know that the other retinal channel, the magnocellular channel, responds well to contours in motion (and especially contours perpendicular to the motion direction). Therefore, if we would extract the SIFT signature of the same local feature, but from the magnocellular channel, it would encode information about local motion.

We propose to describe a local feature by the concatenation of its OpponentSIFT vector (384 dimensions) from the parvocellular (*spatial* information) channel and its SIFT vector (128 dimensions) from the magnocellular channel (*motion* information), thereby obtaining a *spatio-temporal* description of the feature (with 512 dimensions), as illustrated in Figure 3.6c. In this way, we increase the genericity of SIFT local feature descriptors by incorporating motion information.

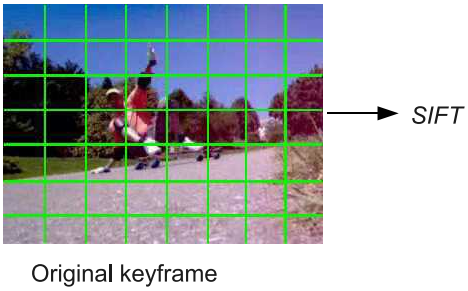
Again, retinal processing is started 10-20 frames before the keyframe, to avoid the retinal transient state.

SIFT multichannel is similar in this respect to the *MoSIFT* descriptor [Chen 2009], in which local features were described as the concatenation of a SIFT vector on the intensity image (spatial description) and a SIFT-like vector on the dense optical flow field (motion description). The magnocellular channel does not give such detailed local motion information as the optical flow field, but it is quicker to compute and it is very easy to integrate in our collection of descriptors.

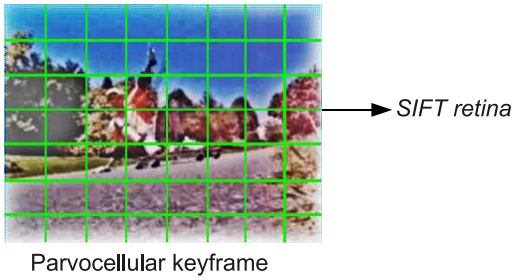
It can also be argued that because the magnocellular channel gives only low-frequency spatial information, SIFT is not the best choice as a local feature descriptor, because it is meant for higher-frequency information. Nevertheless we use SIFT in this set of experiments because it is easier to integrate in our SIFT-based processign chain. Future experiments will address this issue by extracting SIFT at larger scales or by replacing SIFT with Histograms of Oriented Gradients at a spatial scale in accordance with the magnocellular output.

3.2.2 Temporal window based descriptors with salient area masking

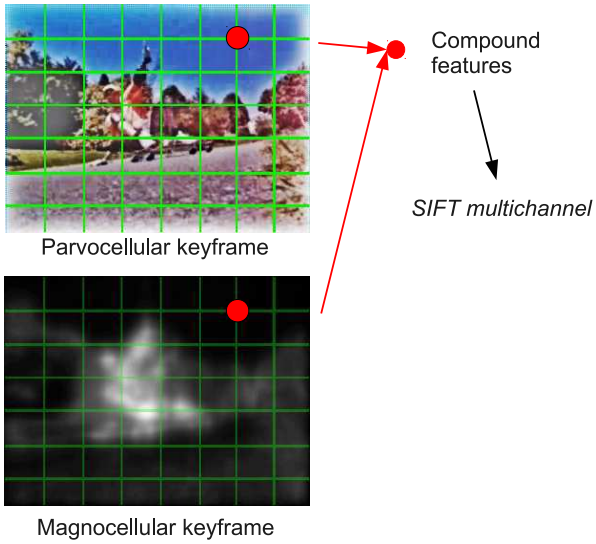
The previous methods only described the video shots at the moments of the keyframes, thus omitting saliency information and, except for the multichannel descriptor, all temporal and motion information. Now, we propose to extract descriptors not only at the moment of the keyframe, but on salient blob areas from an interval of frames (usually between 20-40



(a) SIFT



(b) SIFT retina



(c) SIFT multichannel

Figure 3.6: Proposed keyframe based descriptors: all of them collect local features only at the time of the keyframe, chosen on a dense rectangular grid. *SIFT* collects features from the original keyframe and serves as a baseline. *SIFT retina* collects features from the parvocellular preprocessed keyframe. *SIFT multichannel masking* collects compound parvo-magno features: for a certain position on the dense grid, the OpponentSIFT descriptor from the parvocellular channel is concatenated with the SIFT descriptor from the magnocellular channel to produce the local feature descriptor.

frames) centered on the keyframe, using the area of interest detector from Section 3.1.3.2. We can then extract local features either from the original frames, from the parvocellular channel, or multichannel features. The aim behind focusing the feature extraction process only on salient areas is to obtain descriptors that are less polluted by irrelevant image regions.

This "salient transient blob approach" functions in the following manner: right after we begin processing the temporal window around the keyframe, the transient phase takes place (lasting for less than half of the temporal window duration), during which the magnocellular channel will give strong responses on large spatial structures. Therefore, at this point, we will collect features from *spatially-salient areas*, as illustrated by the mask in Figure 3.1e. After a certain time, the retina stabilizes, and only moving areas will excite the magnocellular channel significantly. Therefore, from this point on, we extract features from *areas of motion saliency*, as in the masks from Figures 3.1e and 3.2d. This way, we integrate in a single descriptor both spatially-interesting and motion-interesting features, constructing a spatio-temporal descriptor. Collected features represent salient spatial information (contextual information), and, if they exist, moving objects features.

The balance between spatially and temporally interesting features is achieved by adjusting the length of time that we take around a keyframe, in relation to the duration of the transient phase determined by the retina parameters. A shorter time interval means more weight for the transient phase, therefore favouring spatial saliency, while longer intervals favour motion saliency. We found experimentally that a window of 20 to 40 frames (depending on the retinal parameters) is a good compromise between the transient state and the stable state of the retina (the spatial and temporal information respectively).

We employ the temporal window around keyframes with transient blob (area of interest) selection to construct the following temporal window based descriptors:

3.2.2.1 SIFT simple masking

This descriptor relies on collecting OpponentSIFT features from the original video frames inside the temporal window around the keyframe, but only from potentially interesting areas, as illustrated in Figure 3.7a.

The expected benefit over the baseline *SIFT* descriptor is that the BoW representation will be based on more representative local features thanks to the transient blob selection. Also, because more frames are taken into account (instead of just the keyframe), this both increases the chances of finding the target concept in the analyzed frames, and feeds more local features into the BoW model, enriching the histogram of visual words.

3.2.2.2 SIFT retina masking

This descriptor is similar to the previous one, except the fact that the parvocellular processed frames are used instead of the original ones, as illustrated in Figure 3.7b. The same benefits as for *SIFT simple masking* are expected, with the following additional properties:

- a reduction in high-frequency noise and compression artifacts, accompanied by local contrast boosting, thanks to the parvocellular preprocessing; during the transient

state of the retina, this effect will be less pronounced;

- the parvocellular channel may increase motion blur on motion-salient blobs, reducing the quality of these features somewhat (visible on the skater's hand in Figure 3.6b and Figure 3.7b).

3.2.2.3 SIFT multichannel masking

In *SIFT simple masking* and *SIFT retina masking*, spatio-*temporal* information is only included in the form of selecting features from interesting areas (often in motion). In *SIFT multichannel masking*, motion information is included explicitly through the addition of the SIFT signature from the magnocellular channel.

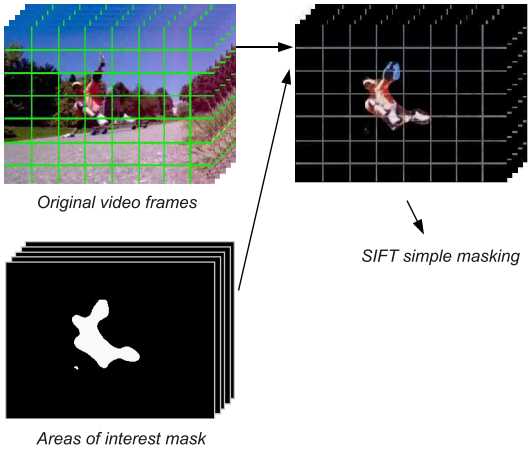
Local features are 512-dimensional vectors, the concatenation of the feature's OpponentSIFT vector from the parvocellular channel and its SIFT vector from the magnocellular channel, similar to *SIFT multichannel*. However, for *SIFT multichannel*, the magnocellular SIFT signature from static areas (such as the sky in Figure 3.6c) is irrelevant. We therefore add the temporal window and salient blob detection for *SIFT multichannel masking*, to focus the analysis on interesting (usually moving) features.

In the methods that we described, we exemplified with (Opponent)SIFT, however these methods can be applied to other local feature descriptors such as SURF, BRIEF, ORB, BRISK, FREAK etc. (see Section 2.3.4.2 for details about local feature descriptors). To prove this, we performed two studies on the impact of retinal preprocessing. The first study uses OpponentSURF features, with results published in [Strat 2012a] and [Strat 2013a] (this study does not include the *multichannel* and *multichannel masking* descriptors, as they were not yet developed at that time). The second study replaces OpponentSURF with OpponentSIFT and adds the two multichannel descriptors, with results published in [Strat 2013b]. Even though the experimental setups of the two studies are different, as well as the local features used (SURF and SIFT), the observations regarding the effects of the retina remain valid for both studies, as it will be shown in the next sections.

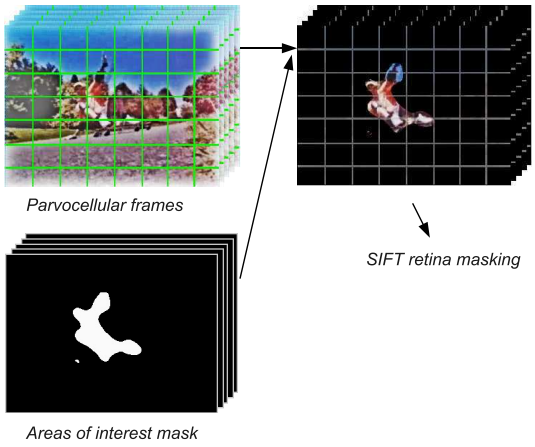
3.3 Experiments

Since our goal is to devise general-purpose descriptors, able to recognize various semantic concepts in various (and uncontrolled) situations, we perform our experiments on the TRECVid Semantic Indexing Task datasets [Over 2011]. These datasets contain a large number of video shots of short length (between a few seconds up to tens of seconds), on which the presence or absence of various semantic concepts (such as *"asian people", "vegetation", "cityscape", "harbor", "ambulance", "airplane flying", "throwing", "cheering"* etc.) has been annotated. Not only the semantic concepts, but also the types of videos are very diverse in these datasets, ranging from amateur videos recorded with a phone camera, to professional news footage. This makes the TRECVid datasets ideal for testing algorithms which aim for a high degree of genericity.

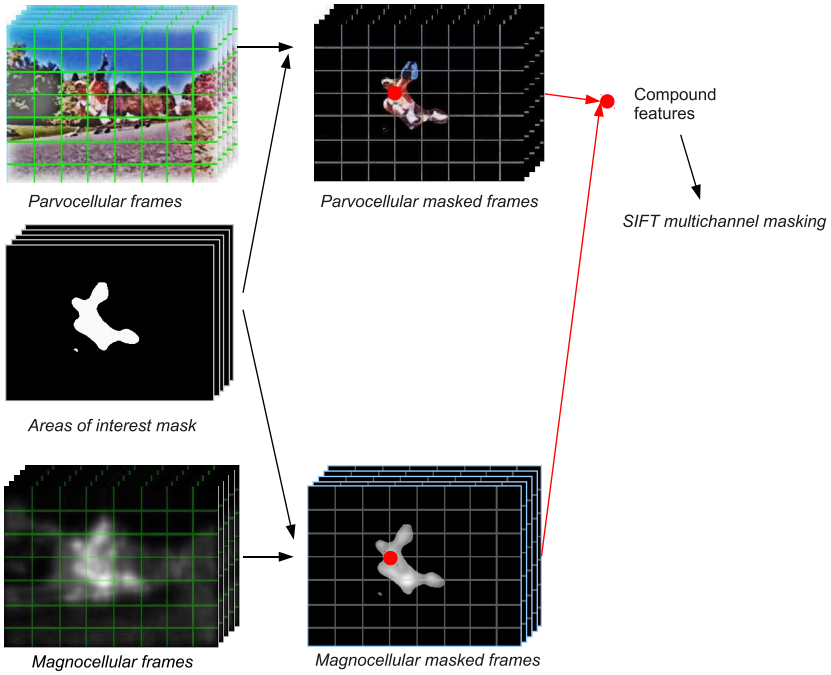
The goal of the TRECVid SIN task is to return, for each of the target semantic concepts, a ranked list of video shots containing the concept. The quality of this ranked list



(a) SIFT simple masking



(b) SIFT retina masking



(c) SIFT multichannel masking

Figure 3.7: Proposed temporal window based descriptors (see text for details).

is evaluated using the official TrecVid measure of performance, the *mean inferred average precision (infAP)* [Yilmaz 2006, Yilmaz 2008], which basically corresponds to the precision averaged for various recall rates.

3.3.1 Preliminary experiments with OpponentSURF

The first study that we performed is concerned with showing the effects of retinal preprocessing on a classical OpponentSURF (denoted *SURF* in the following for simplicity) BoW descriptor extracted on a dense grid. We chose to use SURF-based descriptors in these preliminary experiments because of the lower computational complexity of the SURF feature descriptor (thanks to its use of integral images).

3.3.1.1 Experimental setup

Dataset: We perform this study mainly on the TRECVid 2010 development dataset, containing 130 semantic concepts in $\approx$ 120000 video shots, but we will also show a few results from the 2011 development dataset. For the 2010 edition, we split the development dataset in half: $\approx$ 60000 shots are used for extracting the BoW vocabularies and training the classifiers, while the other half is used for evaluating the performances with the *mean inferred average precision (infAP)*.

Retina and temporal window setup: Regarding the retina parameters, we use the default configuration with mean luminance information cancelling (with $\beta_h = 0$, the parameter of the horizontal cells low-pass filter, see [Benoit 2010] for details regarding the retinal model and all of its parameters).

For the temporal window based descriptors, the length of the temporal window is set to 40 frames, centered on the keyframe of the video shot.

Dense grid setup: The dense grid setup consists of OpponentSURF [Bay 2008] (OpenCV implementation) features extracted from a dense grid with a 9 pixels sampling rate on the video frames. We use a multi-scale grid with 3 scales, with a scaling factor of 1.2 per level.

Vocabulary generation: The feature vocabulary is constructed using the OpenCV implementation of Kmeans clustering, in 3 passes on the training set, using the Kmeans++ initialisation method [Arthur 2007]. As a technical detail, because each concept in the TrecVid database is only present in a very small fraction of the total number of shots, for the vocabulary construction, we select a subset of 1008 video shots from the training database, such that at least 25% of the selected training shots contain at least one positive example of any one of the target concepts. This is in the hope that the vocabulary will be more related to the types of features common with semantic concepts. For keyframe based descriptors, this allows vocabulary extraction on $\approx$ 1.2 million SURF features, while temporal window based descriptors are trained on $\approx$ 6.6 million features. The same subset of video shots is used to generate the vocabularies for each of our proposed BoW descriptors.

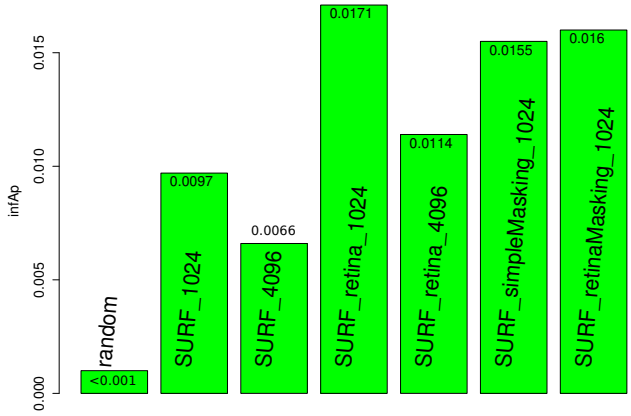


Figure 3.8: Mean inferred average precisions (infAP) [Yilmaz 2008] obtained by the compared methods (plus a random classification), on the TrecVid 2010 dataset

Assigning features to vocabulary words: Fixed-length descriptors in the form of histograms of visual words, corresponding to the methods in Section 3.2, are generated for each video shot, in two versions: either with 1024 or with 4096 visual words (two versions of dictionary size). Local features are assigned to their corresponding vocabulary words using the FLANN library [Muja 2009] (fast *approximate* nearest-neighbour matching) for increased speed.

Supervised classification: Afterwards, the supervised classification stage on the Bag of Words histograms is performed using a K-Nearest Neighbors classifier, as described in [Gorisse 2010].

All methods use the same BoW extraction toolchain shown in Figure 3.5, with the same keyframes (or temporal windows around keyframes), the same sampling rate on the dense grid, the same scales for the multi-scale grid, the same parameters for the retina, the same parameters for the K-means vocabulary generation and the same parameters for the supervised classifier.

3.3.1.2 Global results

General remarks: Figure 3.8 presents the mean inferred average precisions of the proposed BoW descriptors, averaged over the 130 semantic concepts of the TrecVid 2010 dataset part that we used for testing. First of all, the results might appear low, in the range of 0.01-0.023, but they are in fact of the same order of magnitude as other BoW approaches for these datasets, and well above the noise level. For example, [Gorisse 2010] obtain mean inferred average precisions in the range of 0.048-0.054 for descriptors based on dense SIFT. However, our results are not directly comparable with [Gorisse 2010], because SIFT tends to generally give better results than SURF for concept detection [Juan 2009], although at a higher computational cost.

Such low values for the average precisions have several causes. First, the semantic level of the features that we extract is very low, while the one of the concepts to detect is much

higher; this is the so-called *semantic gap* problem, and TRECVid participants deal with it by fusing multiple sources of information, but this is not the purpose of this chapter. The second reason for the low average precision values is related to the very low proportion of true positives contained by the TrecVid datasets: there are many concepts which have just a few tens of positives, sometimes even less, for several tens of thousands of negatives; this causes the average precisions to be inherently low, not to mention that it also poses problems when training supervised classifiers. A third reason is the *approximate* nearest-neighbour matching used to assign local features to vocabulary words, which even though is faster than brute-force matching, can sometimes assign the incorrect vocabulary word.

In any case, our main goal in this preliminary, unoptimized study is to show the *relative* improvement obtained on low-level descriptors when applying a fast, bio-inspired preprocessing method. For this, we generate our own baseline *SURF 1024* and *SURF 4096* keyframe based descriptors, which use exactly the same parameters as the retina-based methods, but do not employ retinal preprocessing.

In the following, we analyze the results in more detail, identifying trends related to which descriptor is better, and in which circumstances.

Keyframe based descriptors: The striking point is that all the methods using the retina outperform the baseline *SURF* descriptors. The parvocellular preprocessed keyframe based descriptor (*SURF retina*) increases global performance by 76% for 1024 visual words and by 73% for 4096 visual words), which constitutes a significant relative increase. This increase can be explained through the image enhancement brought by the parvocellular channel: by filtering out image artifacts ranging from noise and compression effects to under or overexposure problems, spatial details are more accurately extracted, improving performances for concepts related to specific objects or textures.

It can also be observed that a vocabulary size of 1024 is better than 4096, because 4096 fragments the feature space too much and the system becomes too sensitive to small variations of image appearance. Therefore, in the following, we focus our analysis only on the 1024-dimensional versions of our descriptors.

Temporal window based descriptors: Moving on to temporal window based descriptors with salient area masking, for *SURF simple masking 1024*, performances increase by 60% compared to the baseline *SURF 1024*. Therefore, the proposed blob detector from Section 3.1.3.2 also has a large beneficial effect. This proves that the detected blobs bring relevant information even with the low-level saliency we suggested. Therefore, the choice of such a low-level but efficient blob selector makes sense especially in the context of very large video datasets requiring fast processing. In [Usman Niaz 2011], another form of saliency evaluation called “Saliency Moments” is used, but the relative performance increase is not as great in their case (less than 10%) as it is in ours.

For the other temporal-window based descriptor, *SURF retina masking 1024*, the gain compared to *SURF simple masking 1024* is not spectacular, we only increase performance by 3%. Most of the performance boost in this case is given by the blob segmentation step, and in these conditions, the additional parvocellular preprocessing does not improve

results much further. This is explained by the retina's coarse-to-fine property. Its parvocellular channel is designed to enhance stable features but smooth transient signals. Thus, within the detected transient blobs, we have transient signals, which are smoothed by the parvocellular channel, causing a certain loss of detail which prevents a further increase of average precision.

3.3.1.3 Concept per concept results

Keyframe based descriptors: When comparing the most effective keyframe-based descriptor, *SURF retina 1024*, with the baseline (*SURF 1024*), 35 out of the 130 concepts obtain an infAP increase greater than 0.005 thanks to the retinal preprocessing. For the remaining concepts, performance differences are not that significant, while average precisions remain low (less than 0.005). For these remaining concepts, the SURF descriptor, with or without preprocessing, is less adapted. This follows the idea that a single descriptor cannot be efficient for all the concepts, justifying the use of fusion strategies between various kinds of descriptors [Gorisse 2010].

Table 3.1 shows infAP for some of the concepts that best illustrate the performance differences between the two keyframe-based approaches. We notice that whenever *SURF retina 1024* is better than *SURF 1024*, the difference is, on average, 3.3 times greater than in cases when the simpler method is better. This supports the idea that our preprocessing greatly improves results most of the time, and when it doesn't, at least performance loss is limited.

More specifically, *SURF retina 1024* generally reacts better to concepts related to spatial structures or textures and to situations where light changes must not be taken into account, but can disturb the baseline a lot. For example, concepts *"Beach"*, *"Computer or TV screen"* or *"Crowd"* can be acquired in various lighting conditions so that the retina light cleaning effect improves the detection. On the other hand, some concepts do not benefit from the retinal preprocessing, but this can be explained by the model properties. For example, *"Actor"* and *"Highway"* are much better detected without the retina, because they are related to motion (of actors, of cars on the highway). Indeed, the retinal parvocellular channel cancels the mid-spatial frequencies of fast moving objects (it introduces motion blur), and in the case of such concepts, it eliminates an important part of the relevant information.

A noticeable difference between *SURF 1024* and *SURF retina 1024* is the good performance of the latter approach for the *"Nightime"* concept, but its lower efficacy for the *"Daytime outdoor"* concept. This can be explained by the fact that the proposed retina parameters ($\beta_h = 0$) cancel the mean luminance of the input images, thereby eliminating the high mean luminance criterion that can identify *"Daytime outdoor"*. For *"Nightime"*, the retina still eliminates the mean luminance, but it also significantly increases the signal to noise ratio, which would otherwise be low because of physical limitations of image sensors. Moreover, the SURF signatures of halos generated by the retina around light sources (e.g. streetlights, car headlights) can hint towards the *"Nightime"* concept.

Table 3.1: Inferred average precisions obtained by *SURF 1024* and *SURF retina 1024* on the TrecVid 2010 dataset, for some particular concepts.

concept	SURF 1024	SURF retina 1024
Anchorperson	0.0834	**0.2328**
Beach	0.0127	**0.1028**
Cheering	0.0140	**0.0555**
Computer/TV screens	0.0795	**0.1536**
Crowd	0.0008	**0.0189**
Female person	0.0029	**0.0170**
Instrumental musician	0.0081	**0.0283**
Maps	0.0163	**0.0475**
News studio	0.0706	**0.1590**
Nighttime	0.0023	**0.0271**
Reporters	0.0759	**0.1892**
Road	0.0137	**0.0574**
Actor	**0.0134**	0.0066
Bridges	**0.0166**	0.0088
Buildings	**0.0237**	0.0158
Daytime outdoor	**0.0447**	0.0341
Highway	**0.0133**	0.0000
Landscape	**0.0371**	0.0108
Sky	**0.0768**	0.0195
Vegetation	**0.0588**	0.0488

Table 3.2: Inferred average precisions obtained by *SURF retina 1024* and *SURF retina-Masking 1024* on the TrecVid 2010 dataset, for some particular concepts.

SURF type	retina 1024	retina Masking 1024
Beach	**0.1028**	0.0132
Birds	0.0037	**0.0172**
Charts	**0.0734**	0.0157
Maps	**0.0475**	0.0097
Mountain	**0.0110**	0.0007
News studio	**0.1590**	0.0124
Vehicle	0.0067	**0.0166**
Walking	0.0021	**0.0101**
Sports	0.0009	**0.0072**
Athlete	**0.0122**	0.0015
Building	0.0158	**0.0346**
Sky	0.0195	**0.0445**
Snow	0.0341	**0.1109**

Temporal window based descriptors: When comparing keyframe based descriptors with temporal window based descriptors, intuitively *SURF retina 1024* would be expected to give better results than *SURF retina masking 1024* for static concepts. As an example, the former respond best to concepts *"Beach"*, *"Charts"*, *"Maps"*, *"Mountain"* and *''News studio"* (see Table 3.2). *SURF retina masking 1024* can intuitively be expected to do better with concepts related to motion, and examples supporting this idea are the concepts *"Birds"*, *"Sports"* and *"Walking"*. However, there are also exceptions, such as the concept *"Athlete"* (better detected with *SURF retina 1024*), and the concepts *"Building"*, *"Sky"* and *"Snow"* (better detected with *SURF retina masking 1024*).

Some of these exceptions are caused by particularities of the concept within the dataset, and the interaction of these particularities with the functional properties of the descriptor. For example, we noticed that *"Snow"* (a static concept) is detected better with *SURF retina masking 1024* (a transient/motion oriented descriptor). This is because the salient blob detector triggers on the fir trees and rocks often associated with a snowy background. Therefore, *"Snow"* is an example of a concept detected through its association with other concepts, thereby illustrating that contextual relations between concepts should be studied for enhanced detection.

As a general rule, we must not forget that the TRECVid datasets are not standardised. Video content can illustrate various situations and recording can be done with any equipment and by any person, which can lead to exceptions from the intuitively expected behavior of the descriptors. Already there was no single descriptor which was systematically the best for every concept, and when also considering that we cannot even predict by intuition which descriptor will be the best for which concept, *complementary descriptors* therefore become a necessity. Information fusion strategies will then help to cope with the various contexts encountered in practice and combine information from complementary

Table 3.3: Mean inferred average precisions (infAP) [Yilmaz 2008] obtained by the compared methods, on the evaluation set of TrecVid 2011. The first descriptor keeps the old 2010 configuration for the retina, the temporal window and the dense grid, while the others are computed using the new 2011 configuration.

Descriptor	infAP 346 concepts
SURF retina Masking 1024 (cfg. 2010)	0.0110
SURF 1024	0.0123
SURF retina 1024	**0.0162**
SURF simple Masking 1024	0.0116
SURF retina Masking 1024	0.0132

descriptors.

3.3.1.4 Experiments on TRECVid 2011 and effect of parameters

Experimental setup: We performed an additional experiment on the TRECvid 2011 development dataset, which extends the 2010 development dataset by adding another $\approx$ 146000 shots for a total of $\approx$ 266000, and another 216 semantic concepts for a total of 346. In this experiment, we use the whole $\approx$ 120000 shots from 2010 for training, and the newer shots for testing. The goal of the experiment is to check if the tendencies from the 2010 dataset are also found on the newer 2011 dataset.

The dense grid is now single-scale, at the original image resolution. This is meant to reduce computational requirements, compensating for the increase in dataset size.

The retina parameters remain the same, apart from the mean luminance attenuation parameter β_h from the horizontal cells low-pass filter (see the retina model from [Benoit 2010] for details). It is now set to $\beta_h = 0.3$ in order to allow some luminance to be processed by the next stages. The main effect is that halos normally present around high local contrasts are now reduced.

The length of the temporal window around keyframes for salient blob extraction is shortened from 40 frames to 30, also compensating for the increase in database size.

Since the retina parameters were changed, the feature vocabularies are re-extracted for the new configuration, following a similar protocol as for the 2010 experiments.

Results: Table 3.3 reports the inferred average precisions over the whole 346 concepts of TRECVid 2011.

A first thing to notice is that one of the best performing descriptors on the 2010 dataset (*SURF retina masking 1024 config 2010*), kept in its previous configuration, is outperformed by all the descriptors in the new configuration, even by the new *SURF* baseline. Because for the *SURF* baseline, the only thing that changed is the dense grid setup, it means that the main performance difference comes from the new, single-scale dense grid.

In the new configuration, *SURF retina 1024* still gives the best global performance, although the relative increase compared to the baseline *SURF 1024* is of only 32% (compared to 76% for the 2010 dataset). The lower relative increase is due to the new $\beta_h = 0.3$

parameter: because some luminance is transmitted, the color energy (we work in an Opponent color space) is occluded by the luminance energy and loses part of its discriminative power at the retinal parvocellular output. As a consequence, the following OpponentSURF local feature descriptor also loses some discriminative power.

Regarding temporal window based approaches with salient blob selection, *SURF simple masking 1024* fails to give an improvement compared to the baseline, while *SURF retina masking 1024* only gives a 7% improvement. Because the temporal window length has been reduced from 40 frames to 30 frames, the retinal transient state (20 frames long) weighs more than the stable state. But the new $\beta_h = 0.3$ parameter, which allows some luminance information to pass, has the effect that during the transient phase, not just spatially-detailed areas are segmented, but also highlight areas (because they now have a higher energy in the magnocellular channel). However, these areas of high luminance do not bring relevant information. Consequently, the blobs extracted during the (dominant) transient phase select non-relevant features and the final BoW loses some of its discriminative power. Shortening the temporal window therefore demands shortening the transient phase of the retina too, and this can be accomplished by lowering the temporal constants of the photoreceptor low-pass filters, however it will reduce the noise reduction effect of the parvocellular channel.

3.3.1.5 Computational cost

From a computational cost point of view, calculating the retinal outputs adds 35 products per pixel (parvocellular and magnocellular total), while the salient blobs segmentation adds another 12 products per pixel. The tradeoff between the added computational cost and the descriptor enhancement obtained has to be considered from a global application level point of view, especially in the case of fusion-based approaches, where we need to compute several descriptors.

Finally, regarding computational optimizations, all the filtering steps can be easily performed in parallel. Indeed, since revision "5a6114e2" of the OpenCV[1] image processing library (August 2012), the retina implementation has been parallelized. We allowed all the filtering steps to be performed in parallel taking into account multi-core processor architectures. Using the IntelTBB[2] library supported by OpenCV, our experiments showed that the retinal preprocessing runs 3 times faster on a 4 physical core architecture (Intel i7 975XE) and 1.8 times faster on dual-core systems (tested on Intel T2600 and i7 3520M processors).

3.3.1.6 Preliminary conclusions

This preliminary study has given promising results, showing that the performances of (Opponent)SURF-based BoW descriptors can be improved through the use of the human retina model. The descriptors using local features collected from the parvocellular-processed keyframes (*SURF retina*) have shown improvements in both configurations, but the temporal window based descriptor with salient blob selection (*SURF retina masking*) is

[1] `http://www.code.opencv.org`
[2] `http://threadingbuildingblocks.org/`

more sensitive to experimental parameters (retina luminance transmission, temporal constants of retinal filters, length of temporal window around keyframe).

The results could be improved further by searching for an even better configuration of the retina and of the dense grid, however the number of configurations that we can experiment with is limited by computational resources: extracting a new set of feature vocabularies requires approximately 1-2 weeks on an Intel i7 975XE processor, due to the k-means clustering step which is very demanding for a high number of features to cluster (in the order of a few million features) and a high vocabulary size (1024-4096). Afterwards, extracting BoW histograms based on these vocabularies, on a large dataset such as TRECVid SIN, requires in the order of 1000 hours of computation (or more), depending on which year's dataset is used and the retinal and dense grid setup; fortunately, we can parallellize this step and compute BoW histograms on the MUST[3] computing center in 2-3 days.

This preliminary study encourages us to investigate retinal preprocessing solutions further: in the next section, we extend the study to BoW descriptors based on OpponentSIFT features, which generally perform better than their (Opponent)SURF counterparts, and we work with a more optimized configuration of the retina and dense grid. We also examine the performances of the more recent multichannel descriptors.

3.3.2 Experiments with OpponentSIFT

We push further the approach to design augmented, general-purpose spatio-temporal descriptors. We build upon the previous work with OpponentSURF and we show that the retinal preprocessing still improves the BoW video description, even when changing the type of local features (SIFT instead of SURF) and the retina filtering behaviors. Afterwards, we examine the effect of the newer, multichannel descriptors, which were not yet developed at the time when we conducted the SURF experiments.

3.3.2.1 Experimental setup

Dataset: We conduct experiments on the TRECVid SIN 2012 development dataset. It consists in detecting 346 concepts (the same as in 2011) within $\approx$ 400000 video shots. Note that 40% of the shots contain at least one concept. The development dataset is split in two parts, called 2012*x* and 2012*y*. We train our algorithms on 2012*x* and evaluate semantic concept detection results on 2012*y*.

This time, the methods are more optimized than in the previous study with SURF, as it will be described in the following.

Vocabulary generation: We randomly choose 4700 video shots from the training dataset (2012x), such that each shot contains at least one concept. For each shot, we collect local features on a dense grid, after applying our retinal preprocessing. In order not to have too many features to cluster (limit imposed by available memory and available time), we retain only 25% of the local features of a video shot for keyframe based descriptors and

[3] https://lapp.in2p3.fr/spip.php?rubrique80&lang=en

only 8% for temporal window based descriptors. This allows clustering to be performed on $2,5 \cdot 10^6$ features for keyframe descriptors and $5 \cdot 10^6$ features for temporal window descriptors. Visual word clustering is performed using kmeans on 1024 clusters, using 3 passes on the training set. The Kmeans++ initialisation method [Arthur 2007] is used for more efficient clustering, as it was done in the previous section.

Assigning features to vocabulary words: In our previous experiments with SURF, we have used hard assignment of a local feature to its closest matching visual word and the FLANN [Muja 2009] approximate matcher. Now, for our SIFT experiments, we replace FLANN matching with *brute force matching* (which is exact and avoids assignment errors) and hard assignment with *semi-soft assignment*, which was shown to perform better [Strat 2013b]. For semi-soft assignment [Liu 2011], for each feature x to match, the $k = 10$ closest visual words (via an Euclidian distance) are detected. A weight w_i is attributed to the i^{th} matching visual word (v_i) with respect to its Euclidian distance $d_{l_2}(x, v_i)$ according to the following equation:

$$w_i = \frac{e^{-\beta \cdot d_{l_2}(x,v_i)}}{\sum_{j=1}^{k} e^{-\beta \cdot d_{l_2}(x,v_i)}} \tag{3.3}$$

The semi-soft β parameter has been set to 10 following recommendations from [Liu 2011].

Retina configuration: First, we modulate the retina's sensitivity to high frequency temporal changes such as noise and compression artifacts. We have found in [Strat 2013b] that increasing the temporal constant of the the photoreceptor low pass filter to 0.9 frames instead of 0.5 frames gives better results, therefore we use the value of 0.9. The higher the value, the higher the robustness to temporal changes (motion) and to noise, at the cost of omitting some salient details.

Second, we balance the system's ability to describe very contrasted textures. The retina naturally enhances *local* contrasts thanks to its luminance compression stages [Benoit 2010], which is interesting for detail enhancement and description in any lighting conditions. We keep this property, but at the same time, we want very contrasted objects to keep their "contrasted" behavior. This is interesting for concepts like *"lights"*, *"sunlight"*, etc. for which halo effects can generate specific SIFT signatures that can be recognized afterwards. We therefore set $\beta_h = 0.01$ (the parameter from the horizontal cells low-pass filter, see [Benoit 2010] for details) so that mean luminance energy is reduced by -40dB, allowing halo effects to appear.

Local features and dense grid: We extract OpponentSIFT features on a dense grid with a 6 pixels step, at the original image scale. Each SIFT local feature description is computed on 16*16 pixel patches, as this was shown in [Strat 2013b] to work better than smaller, 10*10 patches.

Supervised classification: We use the same KNN algorithm from [Gorisse 2010], as it was done for our previous experiments with SURF.

Table 3.4: Mean inferred average precisions of SIFT BoW descriptors employing retinal preprocessing, over all 346 concepts, with classifier training on TRECVid SIN 2012*x* and evaluation on 2012*y*. The gain is relative to the baseline *SIFT*.

Descriptor	infAP	gain
SIFT	0.0830	baseline
SIFT retina	**0.0904**	9%
SIFT multichannel	0.0878	5.7%
SIFT retina masking	0.0843	2%
SIFT multichannel masking	0.0857	3.2%

3.3.2.2 Global results

As seen in Table 3.4, *SIFT retina* is the overall best descriptor, outperforming the *SIFT* baseline by 9%. Similarly to our previous study from [Strat 2013a] where SURF descriptors were used, this shows that keyframe based retinal preprocessing gives higher-quality local features, thanks to the luminance correction, detail enhancement and spatio-temporal noise filtering of the parvocellular channel. The relative gains (in percentages) are not as high as they were for the SURF experiments, because the baseline is already much higher, but the general ranking tendencies are the same.

Multichannel descriptors: The keyframe-based *SIFT multichannel* descriptor ranks overall second-best, with a gain of 5.7% compared to the baseline. It is not as good as *SIFT retina*, because the added SIFT signature from the magnocellular channel is less informative at the moment of the keyframe: since salient area masking is not used, static features (for which the magnocellular response is zero in the retina's stable state) are also taken into account, negatively impacting the shot's BoW histogram. The *SIFT multichannel* descriptor is also highly correlated in classification results with *SIFT retina*, therefore making it less interesting for further analysis.

However, the other multichannel descriptor (*SIFT multichannel masking*), employing salient feature selection inside a temporal window, performs slightly better than its non-multichannel equivalent (*SIFT retina masking*), showing an increase of 1.6%. This is because when we apply salient blob selection, the contribution from static features (for which the magnocellular SIFT signature has no meaning) is reduced, leading to a higher quality BoW histogram.

3.3.2.3 Concept per concept results

Looking at the global results, we would be tempted to say that *SIFT retina* is the best descriptor. However, we now compare, on a concept-per-concept level, the descriptors *SIFT*, *SIFT retina*, *SIFT retina masking* and *SIFT multichannel masking* (we exclude *SIFT multichannel* as it is less interesting):

- *SIFT* is the best for 48 out of the 346 concepts;

- *SIFT retina* is the best for 50 concepts;
- *SIFT retina masking* is the best for 15 concepts;
- *SIFT multichannel masking* is the best for 41 concepts;
- for the other concepts, the difference in infAP is less than 0.005, therefore not very significant.

This means that even though *SIFT retina* is *on average* the best, it is not always the best, therefore justifying the continued use of the other descriptors as well. The triplet composed of *SIFT*, *SIFT retina* and *SIFT multichannel masking* appears to be especially interesting, since these descriptors are each the best for many concepts.

Table 3.5 gives some examples of concepts for which certain descriptors work better, grouped according to the descriptor that gives the best performance. Keyframe based descriptors (*SIFT* and *SIFT retina*) are intuitively expected to work better for static concepts, while temporal window based descriptors (*SIFT retina masking* and *SIFT multichannel masking*, especially the latter) are expected to better detect concepts related to motion.

This holds true for concepts such as "Beach", "Fields", "Forest", "Muslim", which are better detected with *SIFT* and/or *SIFT retina*, or for "Eaters", "Sports", "Throwing", which are better detected with *SIFT retina masking* and/or *SIFT multichannel masking*. It is also interesting to note how many sports-related concepts such as some in the fourth group in Table 3.5 are better detected by *SIFT multichannel masking*, therefore proving that the added information from the magnocellular channel can improve recognition performance for some motion-related concepts.

There are, however, exceptions from what the intuition would suggest. For example, "Skating" is better detected with *SIFT retina*, "Indian person" with *SIFT retina masking* and "Bridges" and "Mosques" with *SIFT multichannel masking*.

Therefore, the remark from our SURF experiments remains valid: due to the extremely varied context in TRECVid, it is difficult to predict which descriptor will perform the best for a certain concept, and information fusion strategies are needed to exploit complementary information coming from the different descriptors.

3.3.2.4 Exploiting complementarity: simple late fusion

The previous remark motivates us to experiment with a late fusion between the different SIFT-based BoW descriptors, with or without different retinal enhancements, to find out if an additional concept detection improvement can be obtained.

We performed a simple late fusion of all 5 descriptors from Table 3.4, by computing the arithmetic mean of the classification scores obtained at the outputs of the k-NN classifier from each of the 5 descriptors. The gains were impressive: the global mean average precision of the fusion is **0.1220**, a 47% increase compared to basic *SIFT* (0.0830) and 35% compared to the overall best-performing individual descriptor, *SIFT retina* (0.0904). A simpler combination, of only *SIFT*, *SIFT retina* and *SIFT multichannel masking*, provides a close result of 0.1210; this reconfirms the remark from Section 3.3.2.3 that especially

Table 3.5: Results of the proposed retina-enhanced SIFT descriptors for some particular concepts on TRECVid 2012y, both the average precisions and how much better the descriptor is compared to chance. Note: these values are for optimized versions of these descriptors, as done by the IRIM group [Ballas 2012b] and discussed in Section 2.2 (power transformation + PCA); however, the rankings among descriptors remain largely unchanged.

Concept	basic	retina	ret. mask.	multich. mask.	chance
Beach	**0.2882**	0.2606	0.1389	0.1447	0.0423
Beards	**0.1210**	0.0947	0.0656	0.0753	0.0314
Reporters	**0.3159**	0.2898	0.1473	0.1472	0.0326
Teenagers	**0.1295**	0.1062	0.0676	0.0549	0.0198
Clouds	**0.3165**	0.2986	0.2230	0.1989	0.0359
Fields	**0.2630**	0.2034	0.1355	0.1199	0.0377
Golf Player	**0.3318**	0.2303	0.1446	0.1535	0.0011
John Kerry	**0.0732**	0.0592	0.0184	0.0120	0.0016
Processing Plant	**0.1111**	0.0653	0.0591	0.0560	0.0010
Birds	0.1386	**0.1911**	0.0921	0.1142	0.0048
People Marching	0.0407	**0.0623**	0.0297	0.0382	0.0186
Swimming	0.3681	**0.5441**	0.4767	0.4541	0.0062
Waterscape, Waterfront	0.3399	**0.3465**	0.2489	0.2723	0.0860
Baseball	0.2888	**0.3010**	0.1815	0.1885	0.0015
Commentator Or Studio Expert	0.3242	**0.3982**	0.1723	0.1461	0.0095
Forest	0.0979	**0.1145**	0.0925	0.0743	0.0164
Muslims	0.2983	**0.4290**	0.0858	0.0913	0.0046
Skating	0.1348	**0.1525**	0.1170	0.0969	0.0240
Eaters	0.0682	0.0428	**0.1117**	0.0986	0.0028
Motorcycle	0.0305	0.0248	**0.0981**	0.0902	0.0030
Indian Person	0.2505	0.2804	**0.4327**	0.4227	0.0048
Taxi Cab	0.1049	0.0777	**0.1559**	0.1340	0.0004
Basketball	0.1941	0.2353	0.3323	**0.4067**	0.0011
Bridges	0.1280	0.1447	0.1530	**0.1809**	0.0112
Greeting	0.0546	0.0809	0.0172	**0.1210**	0.0070
Indoor Sports Venue	0.2304	0.1802	0.3086	**0.3685**	0.0163
Sports	0.2453	0.2315	0.2576	**0.3061**	0.0425
Throwing	0.2677	0.1984	0.1965	**0.3569**	0.0037
Mosques	0.0022	0.0625	0.0008	**0.0833**	0.0005

this triplet of descriptors is interesting. The complementarity between these three descriptors was also evidenced by a Wilcoxon paired-difference test applied on their classification scores.

3.4 Conclusions

In these studies, we presented derivations of SIFT/SURF local image descriptors that rely on a human retina model preprocessing. In the context of visual semantic concept detection with BoW approaches, applied on the realistic and difficult case of the TRECVid challenge, such descriptors provide more accurate and also *complementary* information.

On the one hand, the detection of concepts from video keyframes can be significantly enhanced by preprocessing such keyframes with low-level human vision filtering (the parvocellular retina channel). The involved spatio-temporal properties help the BoW to better describe the static visual scene by reducing sensitivity to noise and to luminance changes, which was confirmed for both SURF and SIFT local features.

On the other hand, we have shown that taking local features from spatio-temporal salient blobs within the video sequence also enhances performances, helping the BoW to describe the areas that provide more relevant information. Even though we did not use a true, high-level saliency model, the performance increase can be very high compared to the baseline (with a good retinal setup), proving that our simple segmentation method is a good compromise between the quality of results and the computational cost.

Multichannel descriptors also provide an interesting lead, as they integrate both local appearance information and local motion information, especially in combination with salient blobs on temporal windows. *SIFT multichannel masking* is even more interesting when considering its high degree of complementarity with *SIFT retina*.

The retinal preprocessing approach is flexible and generic, and it can easily be extended to other local feature descriptors. The next steps will consist in further optimizing descriptor parameters, trying other multiscale configurations, experimenting with other state of the art descriptors such as FREAK [Alahi 2012] to further test the extendability of the method, and also testing more sophisticated fusion methods. We expect descriptors that have a sensitivity to luminance changes, incorrect colors, image noise, compression artifacts or other image defects, to be particularly helped by the parvocellular processing, while Bag of Words approaches have been shown to also benefit from the selection of salient spatio-temporal information. Additionally, we will also explore more elaborate late fusion methods in Chapter 5, in order to exploit the complementarity between different descriptors better than with a simple arithmetic mean.

Chapter 4

Trajectory-based BoW descriptors

Contents

In the previous chapter, we have described the spatial appearance of video shots through the use of the *SIFT* and *SIFT retina* BoW descriptors, and we have also started delving into the spatio-*temporal* representation. First, we have focused the BoW local feature selection on areas of low-level saliency, often associated with motion, through the use of descriptors employing masking (*SIFT simple masking*, *SIFT retina masking* and *SIFT multichannel masking*), which, even though they do not include motion explicitly, they at least give a higher importance to areas in motion. Second, we have enriched the description of local features by describing each point on the dense grid not only by its OpponentSIFT spatial descriptor, but also by a SIFT signature of the magnocellular channel, which is representative of motion; this way, descriptors *SIFT multichannel* and *SIFT multichannel masking* use spatio-temporal descriptions of local features, thereby including temporal information explicitly in the form of contours perpendicular to the motion direction. Therefore, the set

of SIFT BoW descriptors employing the human retina model constitutes a generic tool for a spatio-temporal description of video content.

However, when it comes to extracting a more motion-oriented description, the descriptors above have their limitations: the non-multichannel descriptors still represent just spatial information (although with the possibility to focus on moving areas), while the multichannel descriptors, through the SIFT signatures extracted on the magnocellular channel, only describe *local* and *momentary* motion: *local* because it is just the motion in the vicinity of a local feature, and *momentary* because only the motion direction at the instant of the local feature is encoded. The local aspect in both space and time can be advantageous in cluttered scenes where objects occlude one another often, but it cannot represent the *evolution* of motion across frames.

To counter this problem, we propose to complement the spatial and spatio-temporal description from the previous chapter by adding an even more temporally-oriented set of descriptors, in the form of trajectories of points tracked across frames. Point trajectories, as opposed to our multichannel descriptors or to other approaches such as MoSIFT [Chen 2009] or spatio-temporal interest points [Laptev 2003], describe motion as it evolves with time, and for this reason, trajectories can be potentially more discriminative for action recognition tasks than the other methods. They have been shown in works such as [Ballas 2011, Wang 2011] to match or outperform spatio-temporal interest points.

The trajectory descriptors that we propose in this work are inspired from the state of the art [Ballas 2011, Wang 2011]; they are based on tracking a set of points across frames in order to construct trajectories, afterwards describing these trajectories in various manners, and in the end, feeding the trajectory descriptions onto a Bag-of-Words model. The exact functioning is detailed in the next section.

4.1 Functioning

Our goal is to construct a BoW model based on trajectories of tracked points. To this end, the approach that we use is based on the following steps, also illustrated in Figure 4.1:

- *Choose a set of points to track*: obviously, if we want to track points across frames, we first have to choose these points. Ideally, these points should cover the frame in a dense enough manner and should be distinctive features, so they can be tracked more precisely. The algorithm for choosing points should not demand high computational resources.

- *Track each point across frames*, until tracking is lost, the trajectory becomes too long or motion stops: this way, we construct *trajectories* of tracked points, which are the elements that we feed into the Bag of Words model.

- During tracking, get an *estimate of the camera-induced motion* of the trajectory and store it. This will allow, if desired, to distinguish between an object's real motion (which is generally more meaningful for action recognition) and the camera motion (which is in most cases meaningless).

- *Add new tracked points from time to time* to replace those whose tracking has ended: in videos longer than a few dozen frames, tracking can be lost for various reasons and new trajectories need to be created to continue describing motion. Also, even if there are enough trajectories active at a certain time, we may wish to add new ones, whose characteristics will be different because they start later; this enriches the BoW representation.

- At the end of the video shot, *filter out trajectories that are too short or that do not have enough motion*, because these usually come from static background points, therefore they are meaningless. Also, retained trajectories can be *post-processed* to make their descriptions more robust to various unwanted phenomena.

- *Compute descriptions* for each retained, post-processed trajectory: these provide a more compact description of the trajectory (compared to the tracked position in each frame), which can be robust to certain variations, depending on the description. Making an analogy with SIFT BoW descriptors, a trajectory's description is the equivalent of the SIFT signature of a local feature. A trajectory can also be described with a concatenation of two or more signatures, such as a histogram of motion directions and a histogram of acceleration directions (analogous with the *SIFT multichannel* descriptor, whose local feature signature was a concatenation of the OpponentSIFT vector from the parvocellular channel and the SIFT vector from the magnocellular channel).

- For each type of trajectory descriptor (or combination), *construct a BoW descriptor* using the classical BoW framework: k-means clustering, describing each video shot using a BoW histogram, followed by supervised classification. Pursuing the analogy with SIFT BoW descriptors, the "local features" are now trajectories, and we feed trajectories into the BoW model (one BoW model per type of trajectory description).

We describe these steps in more detail in the following subsections.

4.1.1 Choice of points to track

There are two possible strategies to choose points to track: either from a dense grid, as it is done in [Wang 2011], or by detecting interest points.

Choosing points from a dense grid has the advantage of giving a very large number of trajectories, which is useful in BoW approaches; however, this requires computing dense optical flow, which is computationally expensive. Additionally, because points are not necessarily chosen on specific, distinctive features, the drifting problem (by which a trajectory gradually drifts away from the real position as tracking progresses across frames, because of small errors in the optical flow computation), can be more severe than if very distinctive features were tracked. This is due to the equation of gradient-based optical flow [Fleet 2005]:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \tag{4.1}$$

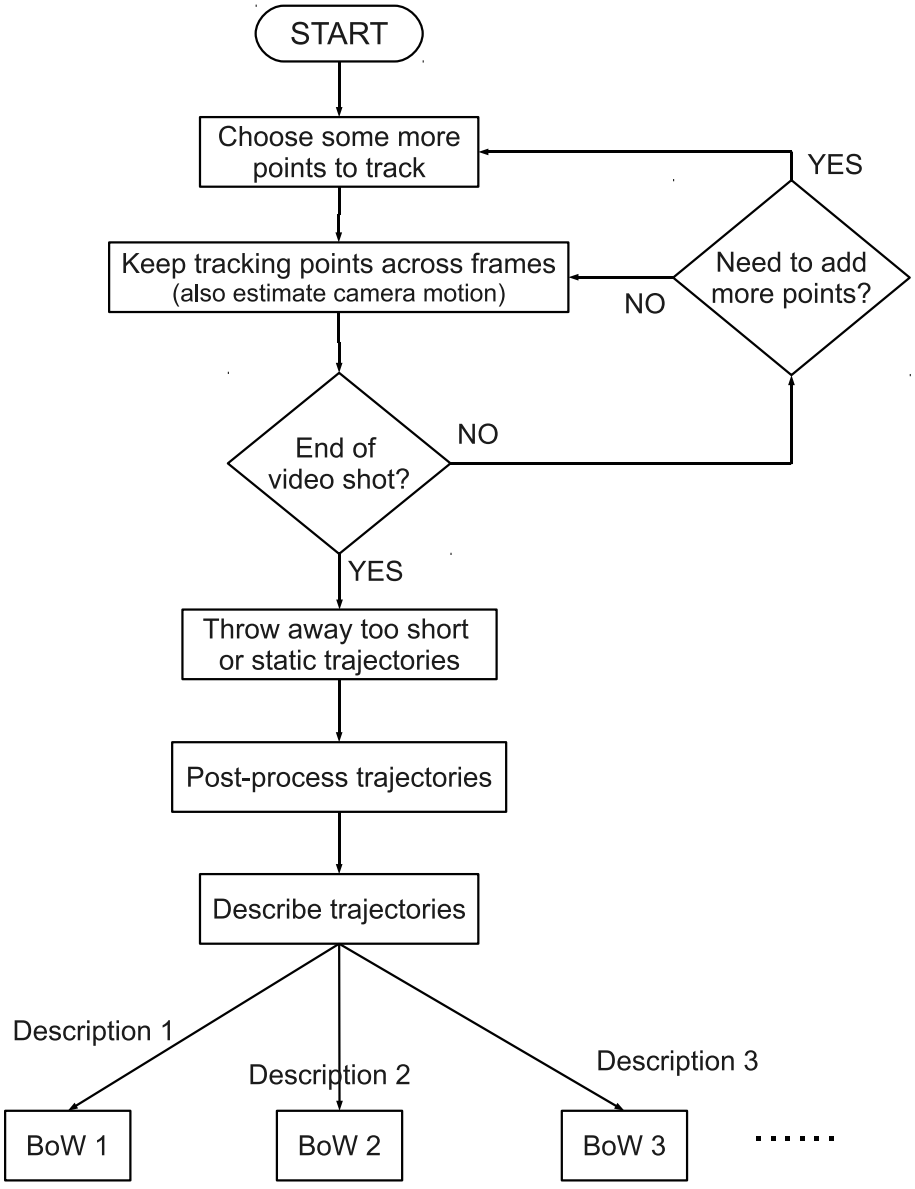


Figure 4.1: Processign chain for constructing BoW descriptors with trajectories as features (see text for details).

where I is the image intensity and V_x and V_y are the horizontal and vertical speeds. In places where $\frac{\partial I}{\partial x}$ or $\frac{\partial I}{\partial y}$ are close to zero, the corresponding speed components are ill-defined.

The drifting problem can be reduced by limiting the maximum length of trajectories, as it is done for example in [Wang 2011], where the length of trajectories is limited to 15 frames, during which substantial drifting does not have time to occur.

Alternatively, an interest point detector can select distinctive features to track, as in [Matikainen 2009], where the Good Features To Track (GFTT) detector [Shi 1994b] has proven itself as a good choice for this application. The advantage is that optical flow needs to be computed only for a sparse set of features, which is computationally less expensive than dense optical flow [Matikainen 2009]. Also, because the tracked points stand out very well, the optical flow is more accurately computed and the drifting problem is reduced. The GFTT detector is in fact an extension of the Harris corner detector [Harris 1988], which ensures that the detected points have high enough values for $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ so that the speeds can be estimated more precisely.

We opted to use this latter approach in our experiments, of tracking discrete points given by an interest point detector, because of the large size of the TRECVid dataset which forces us to compute the less computationally-expensive discrete optical flow. As an interest point detector, we choose GFTT, for the reasons stated in the previous paragraph.

4.1.2 Tracking strategy

Now that we have chosen the points to track, we need a method to track these points across frames. We considered three possible strategies for tracking:

1. *A classical KLT (Kanade-Lucas-Tomasi) tracker* [Birchfield 2007]: we compute the optical flow at each tracked point's location using the Pyramidal Lucas-Kanade algorithm from OpenCV [Bouguet 2000]. The positions in the next frame given by the optical flow become the new positions of tracked points, and the process is repeated. This approach has the advantage that it is very fast, however as time progresses, tracked points can drift farther and farther away from their real positions due to the accumulation of small tracking errors.

2. *By matching SIFT/SURF descriptors* (or other local spatial descriptors) of tracked points from one frame to the next: we detect keypoints in the next frame, compute their descriptors and match them with points in the previous frame according to the similarity of descriptors. Matches that are too different are disregarded (traking is ended for those trajectories), as well as ambiguous matches (for which the first and the second best matching keypoints are not dissimilar enough). Compared to the KLT tracker, this approach reduces the drifting problem, because the exact position of a keypoint is redetected in the next frame; it is also immune to the *aperture problem* that optical flow approaches experience. However, this method requires much greater computational resources, for computing (and matching) SIFT/SURF descriptors. There is also a risk that not exactly the same keypoints will be detected in the next frame, leading to loss of tracking, or that more than one good match is found, leading to ambiguity and again loss of tracking.

3. A *hybrid approach* between the previous two: optical flow is used to estimate the approximate position of the keypoint in the next frame. Keypoints are redetected in the next frame and their descriptors computed. For a keypoint in the previous frame, its match is searched only among the new keypoints in the vicinity of the location predicted by optical flow, and the match is chosen based on descriptor similarity. Compared to the previous approach, this reduces the risk for ambiguous matches, because matches are searched in a smaller area instead of the entire frame. However, the computational cost is even greater, because it requires computing both optical flow and SIFT/SURF local feature descriptors.

For all methods, if a match was found at a too large distance, it is considered an error and that trajectory is ended.

The first approach, the *KLT tracker*, was by far the fastest (faster than real-time on videos of the KTH dataset). Tracking was quite precise, although after $\approx 40 - 60$ frames (depending on video quality, frame rate and type and speed of motion) the trajectories started to drift away from the good positions. The amount of drifting was not enormous (in the order of 2-3% of the frame width, depending on the video resolution and motion), and because the Pyramidal Lucas-Kanade algorithm worked at multiple scales, it was still capable of recovering the motion of the tracked point. Nevertheless, we decided to limit the length of trajectories to $\approx 40-60$ frames maximum to work with more reliable trajectories.

The second approach, based on *matching SIFT/SURF descriptors* (or their Opponent color space versions), was found to be unusable in our tests: it was much slower, and especially on textured surfaces, many ambiguous matches were found and therefore discarded. Because of this, the method had difficulties constructing trajectories longer than 5-10 frames, because tracking was lost too easily.

The third, *hybrid approach*, had a similar computation time as the second approach. However, because optical flow restricted the search area for potential matches, it was able to construct some long trajectories, of lengths up to ≈ 30 frames And thanks to the matching of keypoint descriptors, the drifting problem is greatly reduced, because now the match falls exactly on the interest point's position. Unfortunately, because from one frame to the next, the interest point detector did not always return the same keypoints, this method, too, lost tracking quite quickly, as certain keypoints no longer "existed" in the next frame: only 10-20 trajectories longer than 15 frames were active at any time, compared to hundreds from the KLT tracker.

For the second and third approaches, we experimented with several interest point detectors (SIFT, SURF, GFTT, FAST, ORB) and several interest point descriptors (SIFT, SURF, BRIEF and their Opponent versions) but found the same unsatisfying results.

Therefore, *we chose to use in our experiments the KLT tracker*: GFTT keypoints detected in a frame constitute the starting points of trajectories, and the rest of the trajectories is constructed just by computing optical flow from one frame to the next. When a trajectory reaches a maximum predetermined length (of 2-3 seconds, corresponding to 50-75 frames for 25fps videos), or when it "jumps" too much from one frame to the next (more than 10% of the frame width), or when it reaches the frame border (the object is exiting the scene), or when there isn't enough motion (less than 1.5% of the frame width in the last 0.5

seconds), this trajectory is ended (parameters related to time are expressed in seconds, because seconds are more relevant for the duration of an action, as the frame rates may vary). 0.5 seconds is enough to not interrupt an action with just a momentary stopping point, but at the same time stop trajectories whose motion has ended. These parameters were chosen empirically so that tracking would function correctly, however the exact values are not critical. Examples of trajectories are given in Figure 4.2.

4.1.3 Camera motion estimation

In many video datasets, especially those in which amateurs also contribute videos (such as TRECVid), camera motion is an ever-present problem, and it can mask the interesting motion of persons or objects in the scene. We therefore propose to estimate camera-induced motion at each tracked point and in each frame, and store the camera-induced motion component of the tracked point in its trajectory. Based on this, we will later construct trajectory descriptors both from the "apparent" motion (the total motion, object/person + camera), and from the "real" motion (only of the object/person).

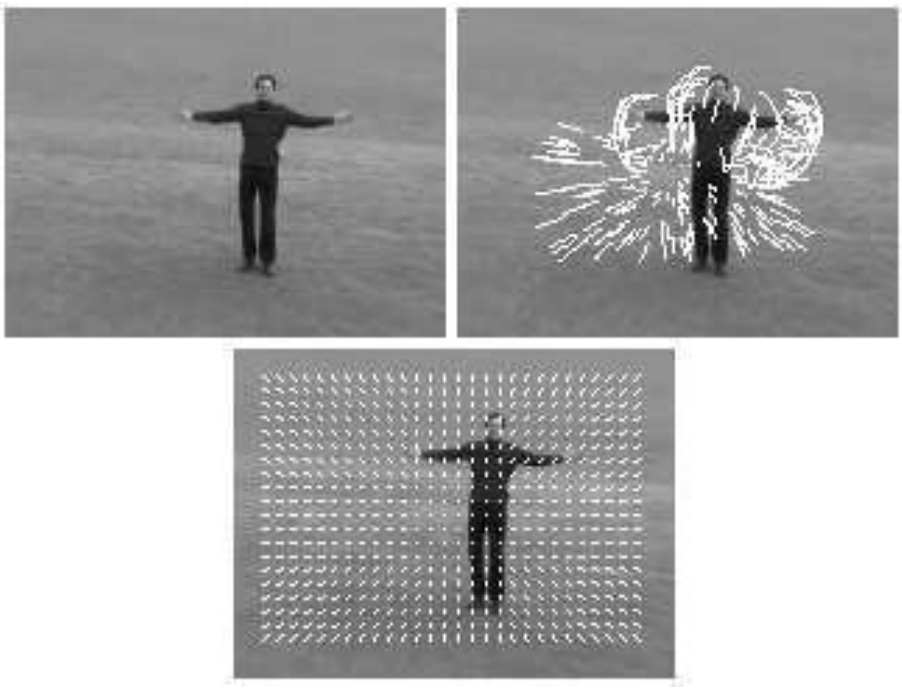
The problem of camera motion estimation is not new and there are many previous publications addressing it. Some authors estimate camera motion directly in the MPEG domain [Ewerth 2004, Wang 1999], while others compute it on the original video [Zhang 1999]. In [Nistér 2005], RANSAC is used to determine ego-motion in videos. In [Ikizler-Cinbis 2010], Harris corners are detected and matched between consecutive frames in order to determine the homography that transforms one frame into the other; RANSAC is again used so as not to be influenced by outlier matches situated on moving objects.

Other ideas to determine camera motion exist, such as the median of optical flow from the grid points, or, as it is done in [Jiang 2012], clustering of optical flow vectors to determine dominant motion. These methods work well for translation motions, but do not function for rotation and/or zooming, because in the latter case, camera-induced motion varies greatly with position.

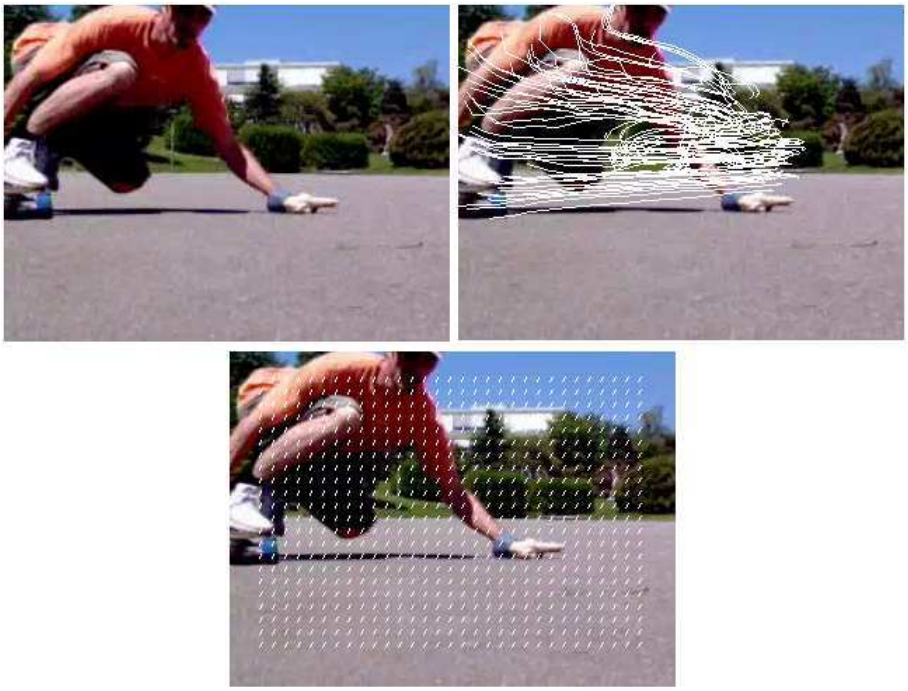
In our experiments, we opt for a simple camera motion estimation algorithm, that also works for zooming and rotation motions thanks to the use of *homographies* for modeling image deformations. The algorithm is described in the following:

First, we choose points on a rectangular grid (the grid step is $1/40^{th}$ of the frame width), but not too close to the border (at least $1/15^{th}$ of the frame width away). We compute the optical flow of these points with the same algorithm as for the keypoints. As long as the background occupies a large enough part of the frame, the motion of most of these points will be only camera-induced. The grid setup is chosen such that it will give a large enough number of points ($\approx$ 1000), while avoiding points close to the border, for which motion cannot be correctly calculated.

Suppose for a moment that there are no moving objects, there is only camera motion. The camera can pan, tilt, zoom and rotate, and all of these, including combinations of them, constitute *homographies* (perspective transformations between a source plane and a



(a) KTH handwaving action



(b) TRECVid video

Figure 4.2: Examples of trajectories on the KTH and TRECVid datasets. The second image in each set depicts the current active trajectories from their starting time until the current frame, while the third image shows the estimated camera motion component between consecutive frames. For the KTH video, the camera is zooming in, with zooming illustrated by the trajectories of some background points and correctly detected by the camera motion estimation method. For the TRECVid skater video, the camera shake is correctly detected, because the skater does not occupy a large enough part of the frame.

destination plane), described by the following equation:

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = H \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{4.2}$$

where (x_i, y_i) are the coordinates of a point in the source plane, and (x'_i, y'_i) are the coordinates of the corresponding point in the destination plane.

The homography matrix H that transforms one frame into another can be estimated from a set of point correspondences (as the ones given by optical flow), and this constitutes the camera-induced frame transformation (the camera-induced motion).

In reality, the points that fall on moving objects (not on the background) will have different motions and they will be outliers with respect to the transformation. We use RANSAC (Random Sample Consensus) [Fischler 1981] to deal with these outliers and recover the true image transformation. RANSAC is a state-of-the-art method for determining perspective transformations between images, and it has also been used in the context of camera motion estimation [Nistér 2005]. We use the implementation from OpenCV to find the homography using RANSAC.

After determining the homography using the points from the grid, this transformation is applied to the tracked trajectory points, thereby obtaining the camera-induced motion component for each of these points, as illustrated in Figure 4.2.

The method that we use is able to correctly estimate camera motion as long as the background occupies a large part of the image (more than 50%, the more, the better), but fails when this condition is not met. Unfortunately, it is very difficult to determine automatically what is the background in such an unconstrained scenario as TRECVid, in order to use only points from that area for camera motion estimation, but we deal with this by considering two versions of trajectory descriptors, both with and without camera motion compensation.

4.1.4 Replenishing the set of tracked points

As tracking progresses, some trajectories end and they need to be replaced. If this is not done, then starting from the beginning of the video, after the maximum length of trajectories is reached (between 2-3 seconds, depending on the setup used), there are no more trajectories left to track; other trajectories are lost because their tracking fails for various reasons (exiting the frame, errors in optical flow computation, occlusion etc.). The rate of loss due to tracking problems is difficult to quantify, as it depends greatly on the content of the video: it can be zero when the tracked points stand out well, the video does not have severe compression artifacts and the motion is smooth; or all tracked points can be lost from one frame to the next when there is an abrupt transition.

Additionally, in order to insure a temporally dense representation of the video, new trajectories should be started at regular intervals (in the order of 0.5 seconds, intuitively corresponding to the length of a short action fragment). These trajectories can spatially overlap those that started earlier, but because the starting moment is not the same, the motion content in each trajectory is different, ensuring a richer representation of the video.

Following these ideas, new trajectories are added to the tracker (new GFTT features are detected) when the number of current active trajectories is very low (below 50), which usually occurs when most of the trajectories are lost, usually due to a change of scenes in the video (a "cut"). Also, to maintain a generally high number of active trajectories at all times, new trajectories are added if no trajectories were added in the last 3 frames and if the current number of active trajectories is below a threshold (between 500 and 3000), the last condition for limiting computational demand. With parameters in these intervals, at any given time, the frame is covered densely enough by tracked points.

Minimum 500 current active trajectories is a good compromise between computation speed and richness of representation, achieving more or less real-time processing on a standard PC (Intel Core i7 running on a single thread), depending on the video resolution and frame rate (videos from the KTH dataset (160x120 @ 25 fps) run 2-3 times faster than real-time). The minimum number of current active trajectories can be increased to improve concept recognition accuracy (the more features/trajectories for the BoW model, the better), at the cost of reduced computation speed.

4.1.5 Trajectory selection and trimming

After the analysis of a video shot has ended, a large set of trajectories has been accumulated, from all over the video shot. An important part of these trajectories carries little useful information: some of them are too short, others have too little motion, while there are also trajectories whose entire motion is only camera-induced.

We choose to throw away trajectories shorter than 0.5 seconds, because these are all trajectories that were ended on purpose by the tracker due to the fact that they experienced insufficient motion during the last (and in this case only) 0.5 seconds. If we were to include these static trajectories into the BoW model, they would degrade performances for action recognition because they would clutter the BoW model with static, irrelevant information. The ratio between the initial number of trajectories and those after this selection step varies greatly from one video to another, but it is roughly equal to the average fraction of surface area occupied by moving elements in the video.

Additionally, trajectories can have "uninteresting" extremities that do not contain motion, up to 0.5 seconds in length (determined by the minimum recent motion condition of having at least 1.5% of the frame width motion in the last 0.5 seconds to continue tracking). We found that trimming away these static ends, if they exist, improves global recognition performances in TRECVid. The trimming algorithm is the following: we start from each extremity (start and end moments) of the trajectory, and if the motion between consecutive frames is less than 10% of the maximum along the trajectory, we cut away that part; we stop trimming an extremity when we find a displacement between consecutive frames above the threshold. However, we keep at the extremities three displacements (frames) below the threshold, on the assumption that they may encode acceleration from a stand-still. The process is illustrated in Figure 4.3.

We have found that trimming away the static extremities of trajectories improves performances by around 5-20%, depending on the trajectory description used, because robustness to static, non-informative extremities is obtained. On the TRECVid 2012 development

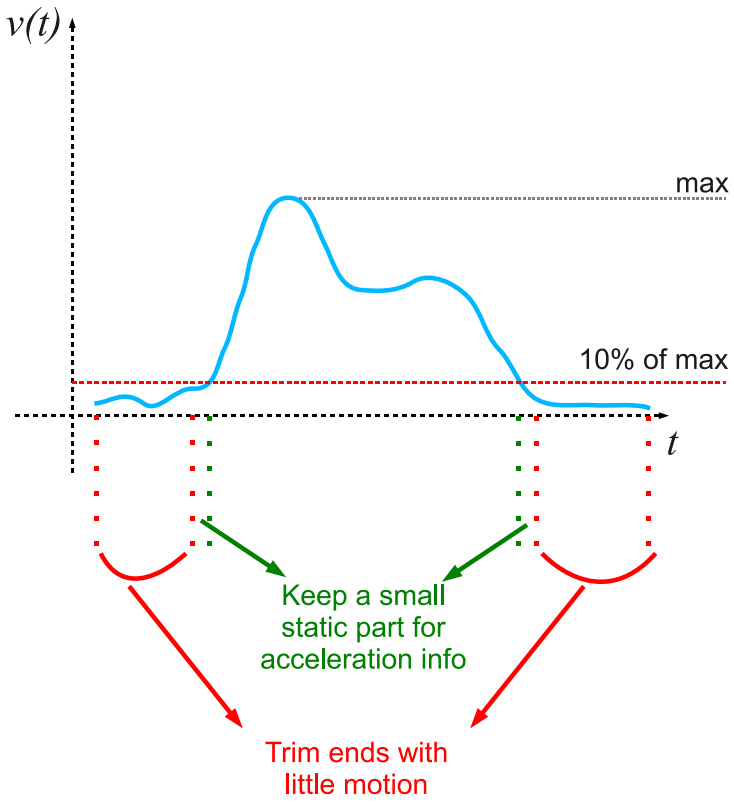


Figure 4.3: Trajectory trimming principle: the speed along a trajectory is traced in blue. The ends of trajectories before substantial motion starts (or after motion ends) are trimmed away: only a small fraction of the ends is kept, to maintain the possibility of encoding acceleration from a standstill (or to encode stopping).

dataset, the normalized vectors of displacements, in their camera motion compensated versions, benefit greatly from this step, with improvements of around 20%. Decreases in performance were only seen for the histograms of motion and acceleration directions with zero-bins, in their camera motion compensated versions (decreases of around 15%).

To sum up, the main parameters of the tracker are the following:

- *maximum length of a trajectory*: empirically set to 2 seconds (number of frames determined according to frame rate); very long trajectories lose sense in simple representations and also suffer from drifting;
- *minimum length of a trajectory*: empirically set to 0.5 seconds; very short trajectories do not provide enough information on the evolution of motion, or may have simply been ended due to lack of motion;
- *maximum allowed displacement between consecutive frames*: empirically set to 10% of the frame width; it should be set according to the usual speed encountered in the video dataset; higher speeds impose higher values for this threshold;
- *minimum allowed recent motion*: empirically set to 1.5% of the frame width; if within each consecutive 0.5 seconds (the same as the minimum length of a trajectory), the total motion is less than the threshold, tracking is ended; the value of the threshold should be set also according to the usual speed encountered in the database;
- *maximum number of active tracked points*: 500 still allows close to real-time performance on a standard PC (2012); increasing the value to 3000 improves concept detection rates due to the better-populated BoW histogram, but also proportionally increases computation time and memory usage;

4.1.6 Trajectory descriptors

At this point, we have a set of selected and post-processed trajectories accumulated along the video shot. The next step is to describe these trajectories, which we do by computing the following descriptors:

1. The *keypoint descriptor of the trajectory starting point*: when we detect keypoints to start new trajectories, we also compute their associated spatial descriptor. This allows us to encode the spatial appearance of the feature we track. Unlike [Wang 2011], where HOG descriptors are averaged along the trajectory, we only describe the start. We chose this approach on the assumption that because of drifting and rotation/zooming of the tracked point, the *average* spatial descriptor loses sense, especially for long trajectories as ours. We use BRIEF as the keypoint descriptor, because it is very compact (32-dimensional), which reduces memory requirements when storing many trajectories, and it is also fast to compute.
2. A *histogram of motion directions* along the trajectory; this is only for the tracked point, not its neighborhood (the HOF descriptor from [Wang 2011] described the

motion in the neighborhood of the trajectory). We use 8 bins for direction (up, down, left, right and the diagonals), and the magnitude of the point's displacement between frames n and $n + 1$ gives its weight in the histogram, as shown in Figure 4.4a. The histogram is then L_1 normalized. This resembles partly what was done in [Ballas 2011] where not only direction, but also magnitude was quantized, resulting in a 25-dimensional histogram (8 bins for orientation with 3 bins for magnitude, plus an extra bin for zero-magnitude). We chose the simpler histogram in our work (without bins for magnitude, we simply accumulate the total motion in each direction) because it does not require setting additional parameters (the magnitude thresholds) and because it is more robust to small variations (small changes in magnitude do not lead to changing bins in the histogram).

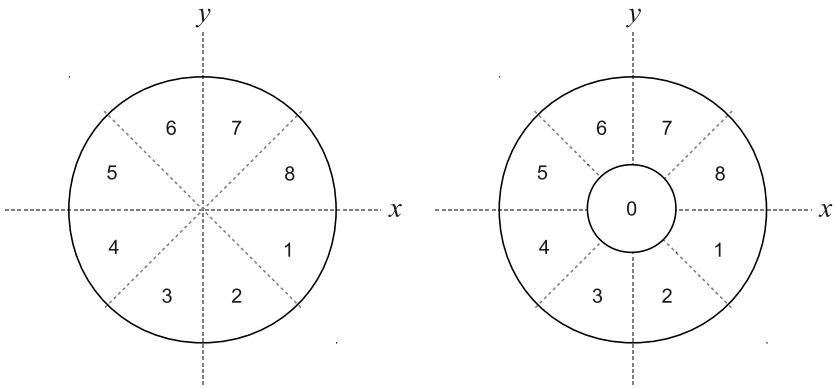
3. A *histogram of motion directions with zero bin* along the trajectory: 8 bins for direction as in the previous histogram, and an additional zero-bin. If the displacement magnitude between two frames is lower than 20% of the maximum displacement along the trajectory, the zero bin is incremented; otherwise, the bin corresponding to the motion direction is incremented, as in Figure 4.4b; note that this time, the value of the magnitude is not added to the bin as it was done for the previous histogram, instead the bin is simply incremented. The histogram is then L_1 normalized. Because the histogram from the previous point did not have bins for magnitude as the approach of [Ballas 2011], we partly compensate with this second histogram, whose goal is to encode whether or not there are any parts of the trajectory that experience little motion (the zero bin). Otherwise, concerning the dominant motion directions, the previous histogram gives more information.

4. *Histogram of acceleration directions* along the trajectory: similar idea as the histogram of motion directions, but working with acceleration information.

5. *Histogram of acceleration directions with zero bin*: similar idea as for motion directions, but working with acceleration information.

6. *Normalized vectors of displacements* in x and y directions: the displacements (motions) in horizontal and vertical directions are resampled to only 8 samples (as if the trajectory only advanced 8 frames) to give a coarse representation of the trajectory. They are then normalized with respect to the total 2D displacement magnitude along the trajectory. A second version of these vectors is generated with 16 samples, offering a medium-resolution representation of the trajectory. Unlike histograms, vectors of displacements are more discriminative, because the order in which motions are performed matters; they are also not robust to temporal shifts of the entire trajectory, but it is to be seen in the experimental phase which is better: more discriminative power or more robustness. Vectors of displacements were used before by [Wang 2011], however they fixed the lengths of trajectories to 15 frames. We, on the other hand, allow variation in the trajectory length, but perform resampling to achieve a fixed description length.

7. *Normalized vectors of accelerations* in x and y directions: they are deduced from



(a) Histogram without zero bin (b) Histogram with zero bin

Figure 4.4: Histograms of motion (or acceleration) directions along a trajectory. The histogram without a zero-bin weights the speed (or acceleration) between two consecutive frames by its magnitude. The histogram with a zero bin does not use weighting, it simply counts the number of occurrences of each bin.

the normalized displacement vectors by temporal derivative and renormalized with respect to the total 2D acceleration magnitude along the trajectory. Two resolution versions are obtained, one with 7 samples and one with 15 samples. One can argue that the acceleration information is already encoded in the displacement vectors, but this is a way to make it stand out, in case it is relevant. For example, the acceleration vectors are invariant to constant-speed camera motion.

All of the motion descriptors are computed in two versions, with and without taking into account the camera motion compensation, as stated previously in Section 4.1.3.

Concerning the keypoint descriptor, we extract it from the parvocellular-preprocessed frames, because we have seen previously that parvocellular preprocessing generally improves the quality of spatial descriptors by making them more robust to image degradations. The parvocellular channel does introduce a certain degree of motion blur, but since the BRIEF descriptor is more compact and encodes less information than the SIFT descriptor, the degradation due to motion blur is less important. In any case, spatial appearance is handled better by the more specialized descriptors from Chapter 3, which are also computed in our complete TRECVid processing chain.

If the choice of using or not the retina for the keypoint descriptor is not critical, keypoint detection and optical flow tracking on the other hand are performed on the original video frames, because the retina introduces motion blur and reduces the quality of tracking.

We have shown the 7 types of trajectory representations that we use to describe each trajectory of the video shot. One representation, the BRIEF descriptor of the keypoint, describes in fact spatial appearance, while the other 6 representations are focused strictly on motion. Of the 6 focused on motion, the normalized vectors of displacements and accelerations are computed at two resolutions, therefore we have in fact 8 motion representations. We can generate these 8 motion representations in versions with or without camera motion

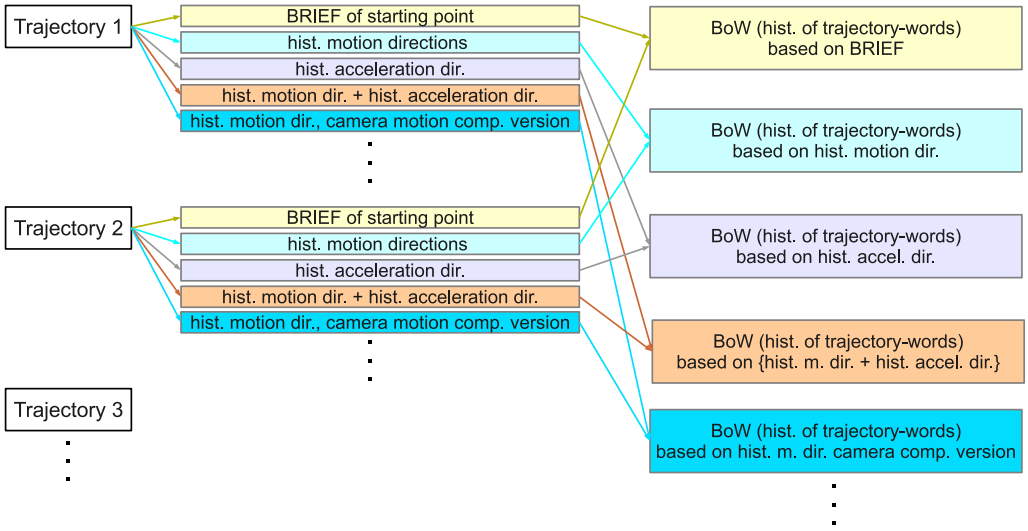


Figure 4.5: Each different type of trajectory representation generates its own BoW representation. Therefore, each video shot will be represented by a set of BoW histograms (of trajectory-words).

compensation, therefore increasing the total number of *elementary* trajectory representations to 17 (1 for spatial appearance and 16 for motion). We also propose a few *combined* representations, in which a trajectory is represented by the *concatenation* of two or more elementary representations among those 17. These combinations are stated in Table 4.2

4.1.7 Integration into the BoW framework

After computing trajectory descriptors, the next step is to aggregate the different trajectories into a model of the video shot. We have chosen the BoW model, because of the simplicity of this representation and the fact that it requires very little prior information and it makes no assumptions about the spatial and temporal structure of events. It would have been very difficult to use a more specialized model, such as the Actom Sequence Model of [Gaidon 2011], because in TRECVid we do not have any annotation information about the time span of events or about their spatial locations.

We can represent a trajectory through any one of the elementary descriptors seen previously, or through concatenations of some of these descriptors. Each of these different representations will generate a different BoW model, as shown in Figure 4.5, and we can perform supervised classification independently on each of these BoW representations.

We use the classical BoW framework from Figure 2.3, with the following details:

- Kmeans clustering is used to generate visual words, with 3 passes performed on the training set, using the Kmeans++ initialisation method [Arthur 2007].

- The Euclidian distance is used to compare two trajectory descriptors (employed at Kmeans clustering and at visual word assignment).

- We use hard assignment to assign a trajectory descriptor to a visual word, because we observed on TRECVid that semi-soft assignment for trajectories is not so appropriate, due to the generally low number of visual words required for trajectory descriptors (see Table 4.2 for concrete examples).
- Supervised classification is done using the KNN tool from IRIM [Delezoide 2011], for reasons of high computation speed (the nearest neighbours need to be found only once for all 346 concepts of TRECVid). Each type of BoW representation undergoes supervised classification independently (we will use information from different representations at a later stage, via late fusion of classification scores). [Ballas 2012b]

4.2 Preliminary experiments on the KTH dataset

The KTH dataset contains 6 actions, each performed by 25 persons and in 4 different situations:

1. outdoor, camera held steady (only very slight camera motion);
2. outdoor; camera zoomed in and out repeatedly for boxing, handclapping and handwaving; camera steady but motion direction changed for jogging, running, walking;
3. outdoor, camera held steady, but the person performing the action has different clothes;
4. indoor, camera held steady;

KTH is considered to be a simple action recognition dataset (as we have seen in Chapter 2, state of the art performance is close to 100%). Our final goal is to include temporal information for concept detection on the TRECVid dataset, but KTH is a good starting point to check the validity of our trajectory descriptors.

4.2.1 Experimental setup

We use a similar experimental setup as [Laptev 2003]: we use the first 8 persons for training (trajectory vocabulary generation and training of supervised classifiers) and the last 9 persons for evaluation. In [Laptev 2003], the rest of the 8 persons were used for optimizing the parameters of the method, but since our final goal is TRECVid, we do not perform any optimization (we do not use these 8 persons) and simply test the default configuration. Most of the experiments used all 4 situations for classifier training and/or vocabulary generation, while we also performed a few tests using only the first situation. We even performed some tests using vocabularies extracted on the TRECVid dataset, to see if the vocabularies from an extremely diverse dataset have any sense in a simple context.

Concerning trajectory parameters, we used trajectories of at least 0.5 seconds and maximum 3 seconds in duration, and motion was considered to have ended if in the last 0.5 seconds, the tracked point moved less than 1.5% of the frame width. Additionally, a trajectory was taken into consideration for the BoW model if at least 10% of its duration

experienced high-enough motion (at least 20% of the maximum velocity along the respective trajectory), with the aim of rejecting less interesting trajectories. Trimming of static ends of trajectories was not applied on the KTH dataset, because the improvement was not yet developed at this stage; however, because KTH contains periodic actions, this should not impact the results by much.

For classification, we used a simple KNN classifier (implementation from the Weka software [Hall 2009]) with K=3 neighbours, which even though it is not the best, gives results quickly and provides information about which trajectory descriptors are better.

4.2.2 Results

Table 4.1 shows classification precisions P for the various trajectory descriptors, with vocabularies extracted on the first 8 persons of the KTH dataset and all 4 situations. The best performances were obtained by the normalized vectors of displacements with 8 and 16 samples, followed by the histograms of motion directions with a zero-bin, and then by the normalized vectors of accelerations. Descriptors based on acceleration information are generally inferior to their equivalents based on speed, with the worst performances given by the histograms of acceleration directions; this is somewhat to be expected, as acceleration is more sensitive to noise than speed (displacement).

Also, the versions employing camera motion compensation do not perform as well as the versions without. One reason is that sometimes the camera motion estimation is erroneous, therefore altering the camera-motion compensated representations. A second possible reason can be due to the dataset itself: the type of camera motion might actually be linked to the type of action (for example, situation 2 means zooming for the boxing, handclapping and handwaving actions, and movement along the diagonal of the frame instead of the horizontal for the jogging, running and walking actions).

We also noted that due to the higher possible variability, the normalized vectors of displacements or accelerations require larger vocabulary sizes to give good results, compared to the histograms of motions or accelerations.

Usually, the highest confusion is between the boxing and handclapping actions (both exhibit horizontal left-right hand movement), and between the jogging and running actions (some of the videos in these two classes are hard to distinguish even for a human examiner).

We also experimented with representing a trajectory by the concatenation of two or more trajectory descriptors, but for the combinations that we tried, we found no improvement compared to the best of the individual descriptors. For example, representing a trajectory by the concatenation of the histograms of motion and acceleration directions, with and without zero-bins, in their versions without camera motion compensation (in total, a concatenation of 4 histograms), with 64 vocabulary words, gives a precision of only 72% (the best of the components has 77,78%). Representing a trajectory by the concatenation of the normalized vectors of displacements and accelerations, all without camera motion compensation, using 192 vocabulary words, gives 77% precision (the best component has 81,94%). This lack of performance increase can be due to the combined representations not being the best choice for this dataset, and/or the vocabulary size not being ideal for the combination.

Table 4.1: Action recognition precision on the KTH dataset (evaluated on the last 9 persons and all 4 situations): vocabulary extraction and KNN training on first 8 persons and all 4 actions, evaluation on last 9 persons and all situations. The vocabulary size refers to the number of visual words used in Kmeans clustering (and the size of the histogram of visual words). Classification precision *P* is shown for trajectory descriptors in two versions: not taking into account camera motion, and subtracting the estimated camera motion from the total motion.

trajectory descriptor	**vocab. size**	**P (%)**	**P (%) with camera comp.**
hist. motion dir.	32	67.13	62.04
	64	71.30	67.59
	128	69.44	70.83
hist. motion dir. with 0-bin	32	**79.17**	70.37
	64	**77.78**	73.15
	128	**79.63**	**75.93**
hist. accel. dir.	32	62.96	51.39
	64	61.57	51.85
	128	63.43	54.63
hist. accel. dir. with 0-bin	32	59.26	53.24
	64	62.04	55.56
	128	61.57	54.17
displace. vect. 8 samples	96	**75.46**	**75.00**
	192	**81.94**	**76.39**
	384	**81.84**	75.00
displace. vect. 16 samples	96	68.98	72.69
	192	**76.39**	**75.46**
	384	**80.09**	**76.39**
accel. vect. 7 samples	96	75.93	62.96
	192	70.83	67.13
	384	68.52	61.57
accel. vect. 15 samples	96	59.72	57.87
	192	66.67	60.65
	384	64.81	57.87
For reference:	**-**	**P (%)**	**-**
STIP [Laptev 2008]	-	91.8	-
tracklets [Wang 2011]	-	94.2	-

A very small improvement can be obtained through early fusion, by representing a video as a concatenation *of different BoW histograms*, each coming from a different trajectory representation. An interesting combination that we found is the early fusion of BoW descriptors based on the histogram of motion directions (64 visual words), the histogram of motion directions with a zero bin (64 words), the histogram of acceleration directions with a zero bin (64 words), the normalized displacement vectors of 8 and 16 samples (each 192 visual words) and the normalized vector of accelerations with 7 samples (192 words), all without camera motion compensation; compared to the best of these individually (at 81,94%), we increased precision to 83,80%.

We did not experiment with late fusion on the KTH dataset, because the KNN classifier simply gave, for each BoW representation, the action class, not a classification *score*. A voting strategy would have had to be used, where each representation voted for an action, but the problem remained of how to choose the weights of each representation and what happens when two or more classes get equal votes. In any case, we could not have extended this strategy to the TRECVid dataset which is our main goal, because in TRECVid we deal with classification scores, not discriminating between different actions.

As for using a *vocabulary extracted on the TRECVid dataset* to describe the videos of the KTH dataset, we tested a few of the trajectory descriptors (and their concatenations). For example, representing a trajectory by the concatenation of the histograms of motion and acceleration directions, with and without zero-bins, in their versions without camera motion compensation (in total, a concatenation of 4 histograms), with a dictionary of 64 vocabulary words extracted from TRECVid, gives a precision of 77%. Representing a trajectory by the concatenation of the normalized vectors of displacements and accelerations, all without camera motion compensation, using 192 vocabulary words from TRECVid, gives 71% precision. These results are close to what we obtained with the KTH vocabularies, and the same was true for all descriptors that we tested. This means that even though the TRECVid dataset is extremely diverse and uncontrolled, the vocabularies make sense and can be used to describe simple actions such as those of the KTH dataset, which is encouraging.

A final note on vocabularies is that the quality of the vocabulary can have a non-negligible impact on the recognition performance. Depending on how the clusters are initialized in the Kmeans clustering algorithm, this can lead to more or less different vocabulary words, which impact the BoW description, which in turn impact the correct recognition rate. In our tests, we found recognition rate variations of up to 13% (for the normalized vector of accelerations with 15 samples, without camera motion compensation, on 96 vocabulary words from KTH) between two runs using the same parameters, due to different random initializations of the Kmeans clusters (but for most trajectory descriptors, the difference is around 5%).

4.2.3 Conclusions

With these experiments on the KTH dataset, we have shown that the trajectory descriptors function properly and are ready for the extension to a larger dataset. Our maximum performance is lower than the state of the art on this dataset, but we used a very generic

and unoptimized classifier, and in any case, our goal was not to be the best on KTH, but to have generic tools allowing us to include motion information in TRECVid. We did not attempt any parameter tuning to improve results in KTH, because these parameters would most certainly not be the same for TRECVid.

In order to obtain very high performances on KTH, very discriminative descriptors are needed (to distinguish, for example, between running and jogging), but in an extremely uncontrolled and diverse context such as TRECVid, a descriptor that is too discriminative also risks being too sensitive to perturbations, irrelevant information and noise. We will see if our trajectories are general-purpose enough for TRECVid in Section 4.3.

4.3 Experiments on TRECVid

We performed our tests on the TRECVid 2012 development dataset, which was split into two parts, both of $\approx$ 200000 video shots: one part called 2012x for vocabulary extraction and classifier training, and the other part called 2012y for evaluation. Most of the 346 semantic concepts are not necessarily directly related to motion, but there are some concepts for which motion is expected to be an important information, such as: Athlete, Basketball, Bicycling, Dancing, Handshaking, Running, Throwing, Exiting a car, Person dropping an object, Violent action etc. We evaluate performances using the official TRECVid measure, the inferred average precision [Yilmaz 2006, Yilmaz 2008].

Of course, we do not expect trajectory BoW descriptors to outperform BoW descriptors of the SIFT family, because most of the concepts are better described by spatial descriptors, but we do hope to add complementary information that can help boost the average precision.

Additional problems in the way of trajectory descriptors are the following:

- Some video shots are extremely short (less than 1 second, even less than 0,5 seconds), in which case we cannot extract trajectories and we have to return an empty Bag of Words (a BoW histogram full of zeros).

- Most of the TRECVid videos have been reencoded to have a standard resolution and frame rate: 320x240 pixels at 25 frames per second. This, of course, increases compression artifacts, but for trajectory descriptors, the big issue is that if a video originally had less than 25 frames per second, frames are duplicated to bring up the frame rate to 25. This creates unnatural motion patterns of moving between two frames, standing still between the next few frames, moving again for a frame, standing still again a few frames etc. This has a negative impact on most of the trajectory descriptors (except the histograms of motion directions without zero bins), and unfortunately, at the time of writing, we have not yet addressed this issue. In the future, we plan to implement a frame duplication detector to try and eliminate this effect.

4.3.1 Experimental setup

For vocabulary generation, we selected only the shots from 2012x that contained at least one semantic concept, in the hope that the extracted vocabulary would better represent the target concepts. From the selected shots, we kept only 12% (with a uniform sampling step on the shot list) in order to limit the number of analyzed videos and reduce vocabulary extraction time, and we extracted trajectories on these shots. From the trajectories generated for each shot, we kept only 10%, in order to further limit the data amount to cluster so that the file can be loaded in the computer's memory. In the end, clustering was performed on 6,8 million trajectories.

In this experiment, we allowed up to 3000 currently tracked points, which is a very high value and has high computational demands. New trajectories were added to tracking (new GFTT points detected) if the total number of currently active trajectories was below 3000 and if no trajectories were added in the last 3 frames. Usually, for videos of 320x240 pixels and 30 frames per second (as in TRECVid), we use only up to 500 active trajectories, for which computational cost is reasonable, but using 3000 improved results by around 10%, for some descriptors by up to 20%.

We used trajectory durations of minimum 0,5 seconds and maximum 2 seconds. Tracking for a point was considered to be good if the displacement between two consecutive frames was not greater than 8% of the frame width, and if the tracked point did not get closer than 3% of the frame width to the border of the frame. Motion was considered to have stopped (and the trajectory ended) if during the last 0,5 seconds, the point moved less than 1,5% of the frame width. The condition for taking a trajectory into account for the BoW model was quite relaxed: either its total motion, or its "real" (after subtracting camera motion) total motion, must be greater than 1.5% of the frame width.

Each selected trajectory is then further processed by trimming away extremities if they contain too little motion, as described in Section 4.1.5, because we have found that this improves average precision for most descriptors.

Concerning trajectory representations, we concatenated at the end of each elementary or combined representation (from Section 4.1.6) the length of the trajectory in seconds, because we found that this increases performances by around 2-3%, for some descriptors by up to 10-15%. The trajectory length is divided by 2, so as not to have a too high weight compared to the other components (the division by 2 is also a normalization with respect to the maximum length of 2 seconds).

4.3.2 Differential descriptors

Using the BoW model with trajectories from the entire video shot means that if the action of interest only occupies a small time interval in the shot, the trajectory-words that represent this action will be in insignificant numbers compared to the rest: in the histogram of trajectory-words, the desired patterns will be "drowned" by trajectories coming from uninteresting elements. and/or uninteresting moments. Unfortunately, we do not have annotations, not even approximate, for the spatial and temporal localization of actions inside TRECVid video shots, which means that we cannot even train a concept detector on "clean"

data.

We propose a workaround for this problem, based on the following idea: suppose that a video shot is composed of motions that are executed throughout the entire duration of the shot, and of motions that are executed only when an interesting action takes place. If these motions (types of trajectories) are quantized onto the dictionary of visual words (trajectory-words), then that would mean that for the entire duration of the shot, some words will be present always (and in about the same amounts as time progresses), and other words will only be present when the interesting action takes place (hopefully, the interesting motion is characterized by different trajectory-words). If the words corresponding to the isolated event could be given more weight in the BoW histogram, then the isolated event would be easier to detect.

To this end, we use the following algorithm:

1. We examine which types of trajectory-words, and in what quantities, start at each frame of the video shot. This can be represented as an image, as in Figure 4.6d.

2. An intermediate step consisting of a MAX order-filter (dilation morphological operator) along the time axis: a sliding window of 20 frames is used to replace the number of visual words of type n in frame i with the maximum number of visual words of type n in the temporal window of 20 frames centered on frame i. This step "bridges small gaps" along time, as seen in Figure 4.6e.

3. The derivative along the vertical (the temporal axis) of the previous "word-image" is computed, and all the "pixels" that are negative are set to zero. This makes the "word-image" respond to *appearances of new trajectory-words*, or to increases in quantity of existing trajectory-words, and makes the "word-image" much less sensitive to trajectory-words that are present in constant amounts over time; the decreases in quantities of trajectory words are ignored by setting to zero the negative pixels. This is exactly the behaviour that we wanted, as we see that in Figure 4.6f, vertical dotted lines are greatly reduced compared to Figure 4.6d. We only choose the positive part of the temporal derivative because we choose to focus on *appearing* trajectory-words, associated to the *onset of new motion patterns*, not disappearing ones. The dilation step was introduced to handle the fact that trajectories are only introduced once in a few frames, which would have harmed the temporal derivative. The fact that the dilated "word-image" is "brighter" than the original does not have any impact on the temporal derivative, because the derivative only regards differences, not absolute levels.

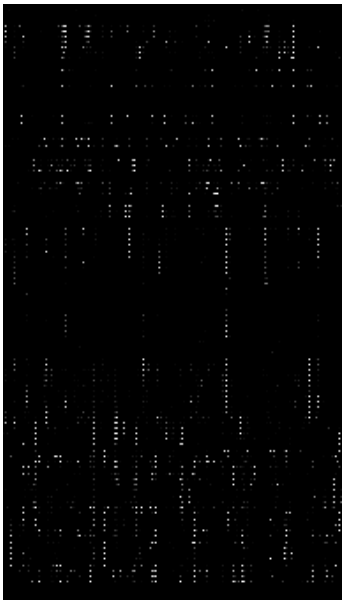
4. The "word-image" from the previous step is summed along the vertical (the temporal axis) to obtain a Bag of Words in which trajectory-words corresponding to isolated events have a much higher weight than normal. The BoW is then L_1 normalized to turn it into a histogram. We call a BoW histogram obtained in this manner a *differential BoW descriptor*.

The length of 20 frames for the sliding window in step 2 above is not critical, it just needs to be wide enough to fill in the gaps between the moments when new trajectories are added.

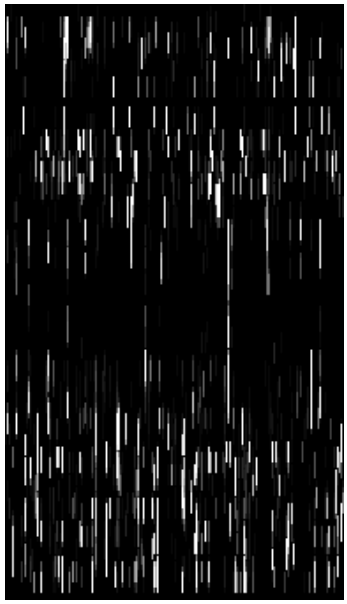
(a) Example frame 1

(b) Example frame 2

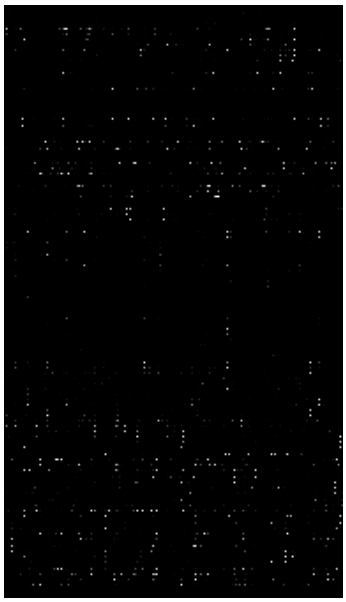
(c) Example frame 3



(d) Original "word-image"



(e) After dilation along time



(f) After positive time derivative

Figure 4.6: Above: example frames from a TRECVid video of a dog running. Below: trajectory-word images depicting which trajectory-words start in each frame of the video (the horizontal axis corresponds to the vocabulary word, the vertical axis to the frame index); the trajectory descriptor used is a concatenation of histograms of motion and acceleration directions and of displacement and acceleration vectors, with and without camera motion compensation, clustered onto 192 visual words. The temporal structure (along the vertical) of the "word-images"can vaguely be associated to the three phases of the video: the dog chasing the car, the dog running back and the appearance of the people. See text for explanation about "word-image" processign steps. You can see that in the final image (after the positive derivative), vertical "lines" (actually dotted lines in the initial image) corresponding to persisting motions are greatly attenuated.

Regarding the parameters of the feature tracker, differential BoW descriptors are an additional reason to sacrifice some computation speed in order to have a high maximum number of trajectories active at any given time (3000 instead of 500), because the temporal derivative reduces the number of visual words that make it into the differential BoW histogram. In general, BoW histograms perform better when there are many local features, and an increase in the number of trajectory-words to start with (before differential BoW) is needed to compensate for the reduction due to the derivative. As a bonus, the standard (non-differential) BoW descriptors will also benefit from the increased number of features and will give better recognition results.

4.3.3 Results

4.3.3.1 Global results

Table 4.2 shows the global (averaged over all 346 concepts) performances of our trajectory descriptors evaluated on TRECVid 2012y. We would first like to remind the reader that in TRECVid, average precisions far from 0.8-0.9 are not out of the ordinary, given the difficulty of the task and the scarcity of true positives for most concepts. We recall that on exactly the same dataset, the retina SIFT-based descriptors from Chapter 3 obtained average precisions around 0.08-0.09, while our purely motion-based descriptors attain 0.03-0.04 (almost 0.05 in combinations); it is interesting that our purely motion-based descriptors attain performances that are of the same order of magnitude as SIFT-based BoW descriptors, given the fact that most concepts are apparently not directly related to motion.

Examining the global results in Table 4.2, we can make the following remarks:

- The camera motion compensation benefits the normalized vectors of displacements significantly (e.g. +10% for (5) with 384 vocab. words), while penalizing most of the histogram descriptors; the histograms with zero-bins are especially penalized (e.g. -22% for (2) with 128 vocab. words). However, the combination C1 of all the individual motion descriptors shows an increase in performance when using camera motion compensation (+5% for 192 vocab. words).

- The differential descriptors generally give a slight decrease in performance (e.g. -2% for (1) without c.c., 128 vocab. words), with the exception of the histogram of motion directions with a zero-bin and the histogram of acceleration directions with a zero-bin (the latter has +8% for 256 vocab. words). Combination C2 also shows a non-negligible increase in performance (+4% for 192 vocab. words) in the differential BoW version.

- The BRIEF descriptor of the first keypoint of each trajectory gives the highest performance (0.0588 avg. precision), however this is mainly a spatial descriptor. It could be considered spatio-temporal in the sense that only spatial descriptors of points in motion are considered, somewhat similar to the SIFT/SURF descriptors employing transient blob masking from Chapter 3.

- Putting BRIEF aside, the highest global performance for individual, purely motion-based trajectory descriptors is attained by the normalized vectors of accelerations

Table 4.2: Global results for semantic concept detection on TRECVid 2012y (training on 2012x), expressed in inferred average precisions, averaged over all 346 concepts. All trajectory representations (or combinations) have the trajectory length concatenated at the end. "c.c." refers to camera motion compensation, "diff." refers to differential BoW descriptors. Combinations refer to describing a trajectory by a concatenation of elementary descriptors. Results in bold show when camera compensation and/or differential BoW significantly improve performance.

trajectory descriptor	vocab. K	AP	AP c.c.	AP diff.	AP c.c. diff.
BRIEF of start point	256	**0.0588**	-	0.0489	-
	512	0.0564	-	0.0473	-
hist. motion dir.	64	0.0367	0.0371	0.0360	0.0364
(1)	128	0.0385	0.0384	0.0377	0.0378
	256	0.0391	0.0386	0.0385	0.0373
hist. motion dir. with 0-bin	64	0.0346	0.0281	0.0341	**0.0321**
(2)	128	0.0366	0.0285	0.0368	**0.0338**
	256	0.0367	0.0282	0.0379	**0.0340**
hist. accel. dir.	64	0.0396	0.0358	0.0378	0.0351
(3)	128	0.0403	0.0375	0.0391	0.0371
	256	0.0408	0.0386	0.0392	0.0377
hist. accel. dir. with 0-bin	64	0.0281	0.0242	**0.0303**	**0.0283**
(4)	128	0.0304	0.0247	**0.0328**	**0.0300**
	256	0.0311	0.0254	**0.0336**	**0.0311**
displace. vect. 8 samples	192	0.0379	**0.0408**	0.0370	0.0413
(5)	384	0.0385	**0.0425**	0.0382	0.0421
	768	0.0389	**0.0420**	0.0386	0.0411
displace. vect. 16 samples	192	0.0374	**0.0413**	0.0366	0.0411
(6)	384	0.0386	**0.0419**	0.0386	0.0420
	768	0.0381	**0.0429**	0.0379	0.0418
accel. vect. 7 samples	192	0.0403	0.0396	0.0387	0.0372
(7)	384	0.0413	0.0412	0.0392	0.0376
	768	0.0412	0.0403	0.0390	0.0380
accel. vect. 15 samples	192	0.0410	0.0421	0.0398	0.0388
(8)	384	0.0428	0.0431	0.0413	0.0411
	768	0.0444	0.0436	0.0430	0.0418
combinations:	vocab. K	AP	AP c.c.	AP diff.	AP c.c. diff.
C1 = 1+2+3+4+5+6+7+8	192	0.0423	**0.0443**	0.0416	0.0439
	384	0.0438	**0.0451**	0.0436	0.0440
C2 = C1 non c.c. +	192	0.0445	(same)	**0.0463**	(same)
+ C1 with c.c.	384	0.0472	(same)	**0.0483**	(same)
C3 = BRIEF + (1 non c.c.)	1024	0.0551	-	0.0453	-
	2048	0.0514	-	0.0420	-
C4 = BRIEF + (1 non c.c.) +	1024	0.0541	(same)	0.0451	(same)
+ (1 with c.c.)	2048	0.0517	(same)	0.0423	(same)

with 15 samples (0.0444), followed by the displacement vectors (with camera motion compensation, 0.0429). Combined trajectory representations based purely on motion, such as C1 (0.0451) and C2 (0.0483), can obtain an even better result.

- When describing a trajectory by the concatenation of the BRIEF descriptor and a motion descriptor (C3 and C4), no improvement is obtained, on the contrary, performances decrease (-7% for C3). However, this is likely due to the number of vocabulary words being set too high, since performances seem to degrade when going from 1024 to 2048 visual words for C3 and C4, and also when going from 256 to 512 for the single BRIEF descriptor. Also, no normalization of the BRIEF descriptor (32 values of 1 or 0) compared to the histograms of motion directions (vectors whose sum is 1) is performed, which could also have a negative impact. These issues will be addressed in the future by experimenting with smaller vocabulary sizes in these cases.

4.3.3.2 Results for particular concepts

For the trajectory descriptors in Table 4.3, we have performed a concept-per-concept analysis of the results. Out of the 346 concepts, 129 concepts had a result better than chance with all of these trajectory descriptors. This may not seem much, but we must remember that the TRECVid dataset is not dedicated to action recognition, and most of the concepts do not have a direct link with motion. Out of these 129 concepts, 30 of them were better detected by one of the trajectory descriptors than by the *SIFT retina* descriptor from the previous chapter. This means that not only the trajectory descriptors are informative even with all the disturbances of the TRECVid dataset (uncontrolled contexts, any types of camera motion, frame duplication at video re-encoding etc.), but for some concepts, trajectories are even more informative than descriptors from the SIFT BoW family.

Table 4.3 shows some results which we consider interesting. For some of the concepts related to motion, such as Athlete, Car racing, Eaters, Indoor sports venue, Fight - physical, Football, the trajectory descriptors outperform the *SIFT retina* descriptors, as it was expected, and in some cases, such as for Eaters and Football, the difference is remarkable.

For some concepts that apparently are not directly related to motion, such as First lady, Bridges, Chair, Snow and Female reporter, the *SIFT retina* descriptor performs better, although the trajectory descriptors also give good results (e.g. First lady has 0.1543 with *t3* and 0.1559 with *SIFT retina*).

For Pickup truck, Police, Gun, Rifles, Court and Press conference, the trajectory descriptors actually perform better, even though intuitively there is no strong relation to motion. The explanation could be that the movement of vehicles might constitute a hint for the presence of Bridges, Van and Pickup truck, while the motion of someone sitting down or standing up might indicate a Chair. Gun, Machine gun, Rifles and Armed person might respond to trajectories because of motions associated to combat, while First lady and Press conference could be detected thanks to waving motions or pointing at someone in the audience. A Female reporter is probably detected not because of the genre, but because of typical motions of a reporter in the news, while Snow might be detected through activities

Table 4.3: Results for some particular concepts on TRECVid 2012y, both the average precisions and how much better the descriptor is compared to chance. Values in bold indicate notable good performances for that concept (discussed in text). Descriptors:
t1 = hist. motion dir., vocab. 256, non c.c., non diff.
t1 diff. = t1 in differential BoW version
t2 = displace. vect. 8 samples, vocab. 384, with c.c., non diff.
t2 diff. = t2 in differential BoW version
t3 = C1 from Table 4.2, vocab. 384, with c.c., non diff.
t3 diff. = t3 in differential BoW version
SIFT r. = *SIFT retina* from previous chapter
chance = what classifying shots randomly would give

Concept	t1	t1 diff.	t2	t2 diff.	t3	t3 diff.	SIFT r.	chance
Athlete	0.1084	0.0903	0.1423	0.1273	**0.1430**	0.1405	0.1367	0.0357
Car racing	0.0785	0.0844	0.1008	0.0903	**0.1118**	0.1090	0.0771	0.0006
Eaters	0.1044	**0.1048**	0.0633	0.0607	0.0623	0.0588	0.0428	0.0028
Indoor sports venue	0.0925	0.0616	**0.2103**	0.2098	0.1632	0.2092	0.1802	0.0163
Fight - physical	0.0087	0.0147	0.0661	**0.0707**	0.0228	0.0247	0.0194	0.0046
Football	0.0982	**0.1014**	0.0715	0.0635	0.0638	0.0519	0.0519	0.0021
Pickup truck	0.1340	**0.1422**	0.1255	0.1281	0.1302	0.1220	0.1286	0.0019
Police	0.0290	0.0262	0.0280	**0.0305**	0.0268	0.0174	0.0002	0.0039
Gun	0.0610	**0.0746**	0.0519	0.0321	0.0401	0.0659	0.0539	0.0340
Rifles	0.0621	0.0603	**0.0952**	0.0488	0.0700	0.0739	0.0500	0.0107
Court	0.0588	0.0588	0.0588	0.0010	**0.0588**	0.0042	0.0064	0.0004
Natural disaster	0.0396	**0.0398**	0.0277	0.0230	0.0311	0.0253	0.0320	0.0075
Press conference	0.0127	0.0147	**0.0851**	0.0721	0.0846	0.0594	0.0144	0.0109
First lady	0.1409	0.0866	0.1369	0.0976	**0.1543**	0.1097	**0.1559**	0.0008
Bridges	0.0918	**0.0943**	0.0898	0.0853	0.0843	0.0865	**0.1447**	0.0112
Chair	0.1052	0.0960	0.1285	0.1275	0.1186	**0.1352**	**0.1468**	0.0460
Snow	0.0859	0.0828	0.0879	0.0875	**0.1069**	0.0956	**0.2308**	0.0292
Female reporter	0.0581	0.0406	0.1357	0.1380	0.1571	**0.1623**	**0.1976**	0.0076
Van	**0.0928**	0.0946	0.0713	0.0646	0.0880	0.0789	**0.1391**	0.0026
Running	0.1224	0.1205	**0.1257**	0.1218	0.1156	0.1072	**0.1509**	0.0064
Soccer player	0.2310	0.2346	0.2303	0.2274	0.2453	**0.2503**	**0.3096**	0.0020
Throwing	0.1195	0.1185	0.1276	0.1075	**0.1404**	0.1318	**0.1984**	0.0037
Skating	0.0384	0.0379	0.1133	0.1213	**0.1488**	0.1470	**0.1525**	0.0240
Swimming	0.0125	0.0155	0.0263	0.0505	0.0540	0.0637	**0.5441**	0.0062

usually related to snow, such as skiing.

In the end, there are also concepts that are intuitively highly related to motion, but in practice are better detected by *SIFT retina*. In the case of Running, Soccer player, Throwing and Skating, trajectories still perform comparatively well, owing to the motion content. However, for Swimming, *SIFT retina* gives an astonishing performance, while the best of the trajectory descriptors are around 10 times worse. This could be due the motion patterns of waves generated by a person swimming, which start to resemble chaotic camera motion, and the difficulty of correctly tracking the motion of body parts under the agitated water surface; however, such wave patterns can be correctly described by SIFT signatures, justifying the good result of *SIFT retina*.

4.3.3.3 Complementarity of descriptors

By examining Table 4.3, we can see that no single descriptor is the best for all concepts. Some concepts are better detected with a certain trajectory descriptor, while other concepts are better detected by other descriptors. We can also see that even though globally, differential BoW descriptors are not as good as their regular counterparts (see Table 4.2), for some concepts they do outperform the normal versions.

But descriptor complementarity can go beyond simply choosing the best descriptor for a particular concept. As we have shown in [Strat 2012b] (and detailed in the next chapter), a *late fusion of classification scores* coming from various descriptors can also boost performance when the descriptors are complementary. This late fusion can achieve significantly better results than simply taking the best descriptor for each of the semantic concepts, thereby proving that the descriptors being fused contain *complementary* information.

In the case of our trajectory descriptors, we test complementarity by performing a set of simple late fusions within different sets of descriptors, each set designed to highlight the complementarity between certain types of descriptors. Here, the late fusion that we use is just the arithmetic mean of classification scores coming from different descriptors, because we just want to illustrate that there is a gain when performing fusion. However, in the next chapter we will experiment with more complex fusion methods, in order to optimize results and exploit complementarity to the maximum. In this section, we will also present a result of such an optimized fusion applied on our trajectory descriptors, but the method will be presented in detail in the next chapter.

We conduct the following simple late fusions by arithmetic mean of classification scores:

- *Fusion basic*: descriptors (1) to (8) from Table 4.2, each with its middle vocabulary size (128 for (1-4), 384 for (5-8)), all without camera motion compensation and in their normal (non-differential) versions; this will test if the different types of how to represent a trajectory are complementary;

- *Fusion c.c.*: the same descriptors as in *Fusion basic*, but in their camera motion compensated versions; the goal is again to test complementarity between different types of trajectory representations, but this time in the camera-compensated version;

Table 4.4: Global results of simple late fusions of classification scores (see text for details), along with two references: the motion descriptor among 1-8 (with or without c.c., in normal or diff. form) that achieved the best overall result (0.0431), and what would be obtained if we were to take, for each concept, the best descriptor from the previously mentioned set for each concept (0.0565).

Descriptor or fusion	AP
accel. vect. 15 samples, 384, c.c.	0.0431
Best descriptor for each concept	0.0565
Fusion basic	0.0599
Fusion c.c.	0.0579
Fusion diff.	0.0587
Fusion ccDiff.	0.0563
Fusion basic + c.c.	0.0659
Fusion basic + diff.	0.0612
Fusion c.c. + ccDiff.	0.0590
Fusion diff. + ccDiff.	0.0648
Fusion all	0.0670

- *Fusion diff.*: the same descriptors as in *Fusion basic*, but in their differential BoW versions; the goal is also to test complementarity between different types of trajectory representations, but this time in differential BoW version;
- *Fusion ccDiff.*: the same descriptors as in *Fusion basic*, but with camera motion compensation and in the differential BoW versions; similar goal;
- *Fusion basic + c.c.*: the arithmetic mean of *Fusion basic* and *Fusion c.c.*; the goal is to see if the descriptors with camera motion compensation are complementary to the ones without camera compensation;
- *Fusion basic + diff.*: the arithmetic mean of *Fusion basic* and *Fusion diff.*; the goal is to see if differential BoW descriptors are complementary to regular BoW descriptors;
- *Fusion c.c. + ccDiff.*: the arithmetic mean of *Fusion c.c.* and *Fusion ccDiff.*; again, the goal is to see if differential BoW descriptors are complementary to regular BoW descriptors, but this time with camera motion compensation;
- *Fusion diff. + ccDiff.*: the arithmetic mean of *Fusion diff.* and *Fusion ccDiff.*; the goal is to check complementarity between descriptors with and without camera motion compensation, but this time on differential BoW descriptors;
- *Fusion all*: the arithmetic mean of *Fusion basic, Fusion c.c., Fusion diff.* and *Fusion ccDiff.*; the goal is to see if taking everything into account gives an additional boost;

By examining the results in Table 4.4, we can see that all types of descriptors are complementary in some degree, because each of the fusions gives better results than any of the individual components being fused.

- *Fusion basic* is 40% better than the best of its input components (the normalized vector of accelerations with 15 samples (6) with k=384), and similar significant increases are obtained also for *Fusion c.c.*, *Fusion diff.* and *Fusion ccDiff.* compared to each one's inputs, proving that the 8 trajectory descriptors form a complementary set in any one of their versions.

- When combining regular descriptors with camera-compensated versions, as in *Fusion basic + c.c.*, the gain is 10% compared to *Fusion basic*. For differential descriptors with and without camera compensation, as in *Fusion diff. + ccDiff.*, the gain is also 10% compared to *Fusion diff.*. This shows that the complementarity between non-camera compensated and camera compensated descriptors is less important than between descriptors that are different altogether, although the gain is still significant and reproductible between *Fusion basic + c.c.* and *Fusion diff. + ccDiff.*, which means that it is still useful to employ both versions with and without camera motion compensation.

- When combining normal BoW descriptors with differential BoW descriptors, *Fusion basic + diff.* obtained a gain of 2% compared to *Fusion basic*, while *Fusion c.c. + ccDiff.* also gained 2% compared to *Fusion c.c.*. This means that complementarity between normal and differential BoW is not as great as for previous cases. However, we must not forget that this is a very simple fusion, through arithmetic mean, which is not optimized in any way and does not fuse descriptor scores taking into account which descriptor is more appropriate for a particular concept. Also, we have seen in Table 4.2 that for some descriptors, differential BoW descriptors are better globally than normal BoW, and Table 4.3 has reconfirmed this observation for particular concepts. This justifies the continued use of differential BoW descriptors, because for some types of trajectory descriptors and for some concepts, they perform better than normal BoW, and a more adaptive fusion will be able to exploit this information.

- Fusing all of the (1)-(8) descriptors, with and without camera motion compensation and with or without differential BoW, as in *Fusion all*, gives the highest performance of 0.0670, which is 12% better than the best of the input "smaller" fusions (*Fusion basic*), and 55% than the best of the input trajectory descriptors (the normalized acceleration vector). This shows that the more descriptors we add, the better, as each additional amount of complementarity, even if small, will still improve the results.

As a final note on these simple fusions, we can see than even if we would have taken, for each individual semantic concept, the trajectory descriptor among (1)-(8) (in any version) that performed the best for that concept, the result (0.0565) would still have been inferior even to the late fusions that did not take into account all descriptors. The fusion of all descriptors *Fusion all* performed 18% better than the best descriptor individually for each concept, confirming that even a simple late fusion, through arithmetic mean of classification scores, can improve results.

In the next chapter, we explore a *more complex late fusion method* that fuses classification scores coming from different descriptors based on how well they perform for a

particular concept and on how correlated different descriptors are for a particular concept. We will see that this fusion method achieves even better results.

4.3.4 Conclusions

We have devised a set of trajectory descriptors that respond to many TRECVid concepts, even to concepts that are intuitively not very much related to motion. Of course, being motion descriptors, they cannot be expected to give results as good as other types of descriptors, such as those from the SIFT BoW family, but the results are nevertheless interesting. For some concepts, the results are in fact very good, as we have seen in Section 4.3.3.2. The results could be further improved by employing a mechanism that detects duplicated frames due to video file re-encoding, in order to avoid the unusual motion patterns that this can cause, but this will be the subject of a future study.

We have shown that camera motion compensation can improve general results for some descriptors, notably the normalized vectors of displacements, while differential BoW can boost performances for camera-compensated histograms of velocities and accelerations with zero-bins. On a concept-per-concept level, the ranking of descriptors and descriptor versions varies from one concept to another and is difficult to predict by intuition, however a late fusion step (as will be seen in Chapter 5) can help to always maximize performance.

Our different types of descriptors and their versions form a complementary set, each responding better to some concepts and/or in some particular situations. The complementarity is the greatest between trajectory descriptors of different types, and less between different versions of similar descriptors, although both types of complementarity can be exploited to improve results through late fusion approaches.

Exploiting complementarity at its maximum is what motivates us to develop optimized late fusion approaches, not only between trajectory descriptors, but in a much broader set of diverse video shot descriptors, as it will be presented in the next chapter.

4.4 Global conclusion on trajectories

We have shown that trajectories have proven useful on multiple databases, from highly-specialized action recognition in highly controlled contexts such as KTH, to the completely uncontrolled TRECVid dataset, where trajectories have even sometimes detected concepts not necessarily directly related to motion. This proves that our trajectories are a generic tool, applicable to videos of diverse types and diverse contexts. Coupled with the SIFT BoW-based descriptors employing the retinal model, this gives us a rich spatio-temporal description of videos, with some descriptors being highly spatially-oriented (*SIFT* and *SIFT retina*), other descriptors with a mixed spatio-temporal behaviour (*SIFT multichannel masking*) and trajectory descriptors completing the package as highly motion-oriented descriptors.

However, the descriptors that we have treated so far are not the only way to describe videos: for example, there can be other SIFT BoW descriptors for spatial appearance, color histograms for color composition, audio descriptors etc. In the context of our participation at TRECVid as a member of the IRIM group, we have had access to many more descriptors

than the ones treated in this thesis. We exploit such a broad description of videos, to achieve the highest genericity possible (our goal) for semantic concept detection, by fusing diverse multimodal descriptors, as it will be discussed in the next chapter.

CHAPTER 5

Late fusion of classification scores

Contents

5.1 Introduction

As we have seen in Section 2.2, a basic framework for semantic indexing on a multimedia dataset consists in extracting content descriptors from the samples (e.g. images or video shots), then training supervised classifiers on each of these descriptors. In the case of videos, content descriptors can be, for example, color histograms, Bags-of-Words of local features, BoW of trajectories, audio descriptors etc. and supervised classifiers can be K-Nearest Neighbours, Support Vector Machines etc. This produces, for each available descriptor and for each associated classification method, a set of classification scores that describe the "likeliness" of each sample to contain a given target concept. When possible, such scores can be calibrated as probabilities for the samples to contain the target concept.

We call an *expert* any method able to produce a set of likeliness scores for multimedia samples to contain a given target concept. Such scores can then be used to produce a ranked list of the samples the most likely to contain this concept. A combination of a content descriptor and a supervised classification method constitute an *elementary expert*. These steps are represented by the *"Descriptor computation"* and *"Supervised classification"*

blocks in Figure 2.1 (this figure illustrates the entire processing chain that we use in our experiments).

As several content descriptors and several supervised classification methods can be considered, many elementary experts can be built. So far, information coming from different elementary experts is not jointly exploited, as experts are treated independently. However, different types of elementary experts, each based on different aspects of the multimedia samples (such as colors, textures, contour orientations, motion or sounds etc.), give *complementary* information.

Several aspects of complementarity can be discussed. The first is *inter-concept complementarity*, which means that a certain expert (based on a certain type of content descriptor) can give very good results for a particular semantic concept, yet perform poorly for another concept. For example, on the TRECVid SIN video dataset, the concept *"Football"* is better detected by experts using trajectory descriptors than by those using SIFT Bag-of-Words descriptors, or vice-versa, the concept *"Bridges"* is better detected with SIFT Bag-of-Words than with trajectories. As a general rule, there is no single expert which is systematically the best for all target concepts.

The second aspect of complementarity is *intra-concept complementarity*, which means that even if two (or more) experts have modest performances for a particular concept, their combination can produce a *higher level expert* that often performs better than any of its input elementary experts. This is especially true when one of the elementary experts detects the concept better in some situations (corresponding to some of the multimedia samples where the concept is present), while the other expert works better in the rest of the situations (the rest of the samples where the concept is present), which means that there is *complementarity at the context level*.

Because of these observations, for the sake of universality and in order to exploit complementary information, many systems rely on the combination of a large set of experts (up to 100+), each based on different descriptors or descriptor versions, and using various supervised classification algorithms.

As seen in Section 2.2, two broad classes of fusion methods distinguish themselves: *early fusions* combine descriptors before the supervised classification step, while *late fusions* combine the outputs of supervised classifiers (classification scores, *experts*). In the context of the TRECVid Semantic Indexing (SIN) task and as part of our participation with the IRIM group, we opt for the use of late fusion approaches (in a concept-per-concept manner), because an early fusion would mean training supervised classifiers on very high-dimensional descriptors, which is not trivial. Late fusions are easier to apply, because they fuse simple classification scores, not complex multidimensional descriptors, and in the case of TRECVid SIN, it was shown in [Ballas 2012b] that late fusions also give better results.

As inputs for the late fusion, we have a battery of (50+) *experts*, which are classification scores for each of the multidimensional descriptors (and their versions), on each video shot and each concept.

5.2 Choice of late fusion strategy

The goal of the late fusions presented in this chapter is to exploit complementarity between experts as well as possible, for boosting the concept detection performance as far as possible. Therefore, when looking for an effective combination of experts, several interrogations arise: should we use all experts in the fusion process, or just the best ones? Does combining two experts always yield better results than the two of them taken separately? Should we weigh them differently in case one is much better than the other? Tackling a similar problem, *Ng and Kantor* [Ng 2000] proposed a method to predict the effectiveness of their fusion approach and concluded:

> Schemes with dissimilar outputs but comparable performance are more likely to give rise to effective naive data fusion.

where the *similarity* between two experts' *outputs* can be measured as the Spearman rank correlation coefficient – and *naive data fusion* should be understood as fusion by sum of (normalized) classification scores.

We have seen in [Strat 2012b] that the difference between experts mostly comes from the type of descriptors they rely on, and partly from the type of classifiers trained on top of these descriptors. Experts relying on similar descriptors generate similar outputs and therefore strongly agree with each other.

In the context of our work on the TRECVid SIN task, the remark of *Ng and Kantor* should be interpreted in the followign manner (as proven by a preliminary experiment described in [Strat 2012b]): when fusing two experts for a particular concept, each expert based on different descriptors, the maximum performance gain is obtained when the average precisions for the two experts are similar, but the first expert detects half of the true positives of the concept, while the other expert detects the other half (because of, for example, each descriptor working better in a certain context or with a certain type of videos). Therefore, when two experts give approximately the same average precisions, a simple fusion will give the maximum performance boost when the complementarity is also maximum (when the two experts' scores are not correlated, yet they give similar average precisions because each works better in different conditions).

Based on this idea, the fusion approach that we propose has three stages, as illustrated in Figure 5.1:

- First, experts are grouped based on similarity into clusters of similar experts. This grouping can either be done manually, using external knowledge about the internal workings of each expert (e.g. grouping all experts that use color descriptors), or automatically, based on a similarity measure of the experts' scores, as we attempt to do in this chapter.

- Then, intra-cluster fusions are performed, in which the experts from each cluster are fused. This balances the quantity of experts of each type, avoiding the case when numerous similar experts dominate the others (because some groups may be very numerous, while other groups may only have a few or even a single expert), and also helps to reduce classification "noise" within the group.

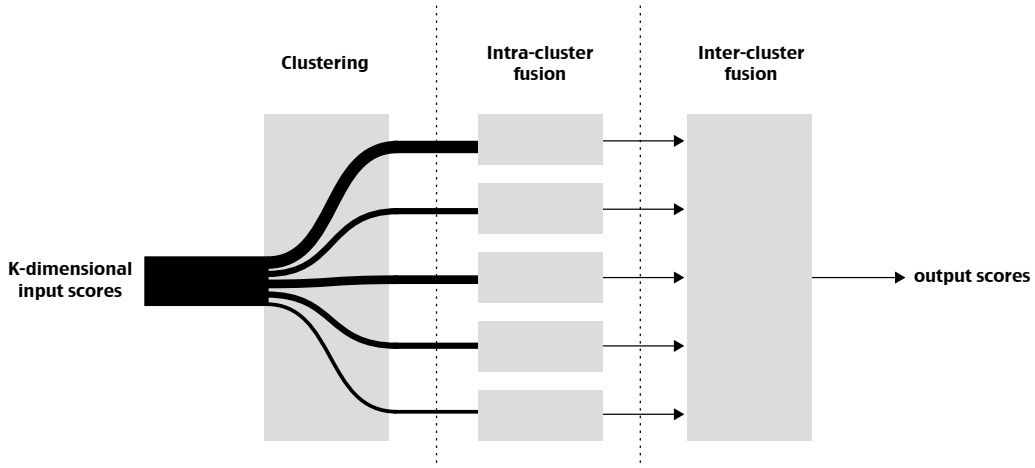


Figure 5.1: Basic principle of our fusion approach and of the other two methods from our collaborators with IRIM (described in [Strat 2012b]: K input experts are available, which are clustered based on similarity into several groups, followed by an intra-cluster fusion and an inter-cluster fusion. Figure from [Strat 2012b].

- Last, an inter-cluster fusion is performed, in which the different clusters (which are complementary because they contain experts of different types) are fused together. This gives the main performance boost due to complementarity, based on the remark of *Ng and Kantor* [Ng 2000].

Our fusion approach combines information coming from different sources (experts) in a way close to the optimum, so that the gain from complementarity is maximized. Additionally, the approach is completely automatical, meaning that it determines by itself how to group experts and what weight to give to each of them. Our method is then compared with a manual hierarchical late fusion done by our partners in IRIM [Strat 2012b], based on the same idea as in Figure 5.1.

5.3 Proposed late fusion approach

The late fusion that we propose, in its original version from [Strat 2012b] is based on grouping and fusing experts progressively based on similarity, until a minimum similarity threshold is reached; it clusters experts into groups and performs intra-group fusion at the same time. Because of this functioning, we call this fusion method *agglomerative clustering*. After this step, inter-group fusion is performed to obtain the fused result.

Compared to what was done in [Strat 2012b], we extend this agglomerative clustering approach by also performing, in parallel, four additional fusions: two versions of AdaBoost fusions inspired from [Cai 2007, Wu 2003, Tang 2008], one weighted arithmetic mean of experts, and the best expert for each concept. At the end, the results of the five fusions are combined by choosing, for each semantic concept, the fusion method among the five that gave the best result for that concept on the training set, as illustrated in Figure 5.2.

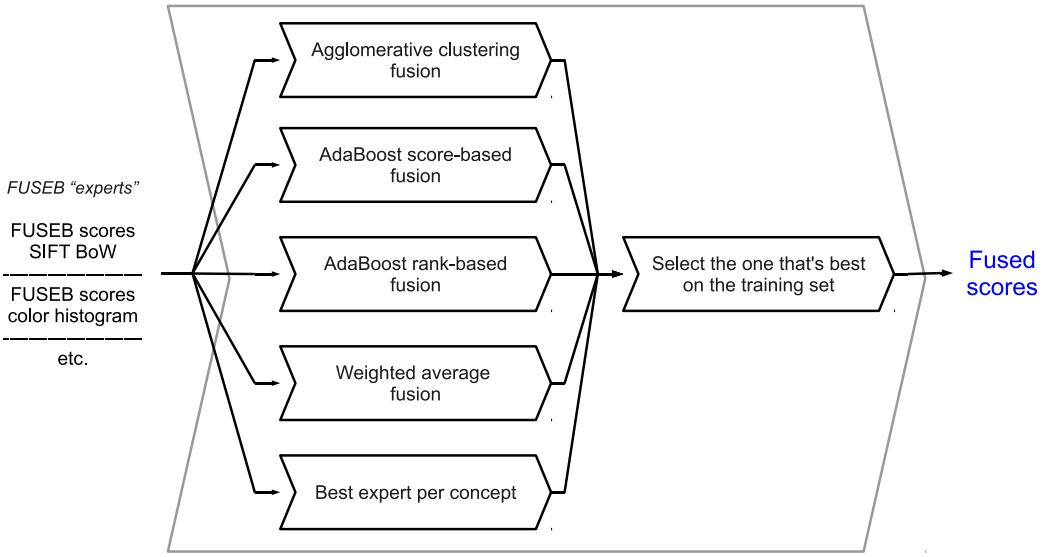


Figure 5.2: Proposed fusion approach, on a concept-per-concept level, corresponding to block *"Late fusions"* from Figure 2.1): five fusions are applied in parallel on the input experts (*FUSEB* experts from Figure 2.1), and the fusion that worked best for a particular concept on the training dataset is selected for that concept on the test dataset.

We will first present the original approach, utilizing only agglomerative clustering, and then we will detail the other fusions with which we compare and also extend the agglomerative clustering.

5.3.1 Agglomerative clustering of experts

The agglomerative clustering fusion method treats each semantic concept independently, and *for each concept*, applies the following steps:

1. *Relevance of experts estimation*: The relevance of each of the input elementary experts is estimated on the training set, for the concept in question. The relevance is measured as the average precision of the expert normalized with respect to chance (the result of randomly choosing samples). An expert with a relevance of 1 means that it performs just as poorly as chance.

2. *Selection of experts*: Experts with a relevance less than 1 are thrown away, because they are irrelevant to the concept in question. Experts with a relevance 8 times smaller than that of the best are also thrown away, in order not to "pollute" the best expert with others that are much worse. This second selection is not critical, neither is its threshold, but using it tends to reduce performance degradation from fusion for the (very few) concepts that have an extremely good best expert.

3. *Iterative fusion*: Some of the retained experts are highly correlated, so we look for the pair of experts *with the maximum correlation* and fuse it into a single expert

(through arithmetic mean). The correlation between the resulting expert and the remaining ones is updated, and the process is repeated. The iterative fusion stops when a sufficiently correlated pair of experts can no longer be found. The iterative fusion corresponds to the first 2 steps in Figure 5.1, as it groups and fuses similar experts at the same time (progressively, as pairs of highly-correlated experts are found).

4. *Selection of resulting fused experts*: The experts resulting after the iterative fusion are again selected according to similar criteria as in step 2. This step is not critical, as generally, the experts resulting after the iterative fusion respect the conditions anyway.

5. *Weighted arithmetic mean*: The iterative fusion does not give a large gain, because it only groups and fuses *similar* experts. The main performance boost comes now, when we fuse *different* groups via a weighted mean of experts. The weights are given by the average precisions (for the current concept on the training dataset) of the experts from the previous step. A single expert is obtained, the result of our agglomerative clustering fusion approach. This weighted arithmetic mean corresponds to the last step in Figure 5.1.

The correlation measure used in the iterative fusion step is the Pearson product-moment correlation coefficient ρ of the raw classification scores. $\rho \in [-1; 1]$, with values in the range of 0.6-1 corresponding to high correlation. In order to fuse a pair of experts, not only does the correlation coefficient for the classification scores of *all* samples need to be at least 0.75 (the two experts give similar information on a global scale), but also the correlation coefficient for the scores of *only the positive* samples must be at least 0.65 (to ensure that the two experts tend to detect more or less the same true positives of the semantic concept being analyzed). The constraint related to positives was added again with regards to the remark of *Ng and Kantor*, as at this stage, we want to group similar (not very complementary) experts; also, without this constraint, because of the imbalance between positives and negatives, the scores for negatives would have dominated the correlation measure. The exact values of 0.75 and 0.65 are not critical, but we obtained good results using this configuration.

The goal of iterative fusion is to balance the contribution of each family of experts, as we will see in Section 5.4.2.1 that some families are very numerous, while other families are small. This method is automatic and avoids needing to specify the families manually, making it practical for often-changing expert sets and for automatically grouping experts of similar types but from different contributors. The groups formed by the iterative fusion correspond in a large degree to the expectations based on descriptor type.

As a side note, because during the iterative fusion, several experts can be added successively to a group, and because at each iteration, an arithmetic mean of the experts in a pair is done, it would mean that the last expert added in a group would always have a larger weight than classifers added previously. We compensate for this by keeping track of how many experts were already used to form an intermediate expert (at a certain time during the iterative fusion), and adjust weights accordingly so that at the end of the iterative fusion, all inputs that went into a resulting expert have the same weights.

Alternatively, instead of using an arithmetic mean with equal weights for the late fusion step, we can assign weights for the input experts based on their average precisions for the current concept. However, the performance difference of this setup compared to the simpler one was minimal (because similar experts have similar performances anyway), therefore we retained the simpler setup. In this case, non-uniform weights will only be used in the final, weighted arighmetic mean step, when combining complementary experts.

In addition to the agglomerative clustering fusion, we also experiment with other fusion approaches and with combining the results from these different fusion approaches, as described in the following.

5.3.2 AdaBoost score-based fusion

AdaBoost [Freund 1997], short for "adaptive boosting", is an algorithm that constructs a strong expert through a weighted average of a large number of weak experts. AdaBoost functions properly when each of the weak experts is at least slightly better than chance, and when the different involved experts are complementary (they each correctly classify different parts of the dataset). This is very much the case of TRECVid, where we have a large battery of experts, most of them not having spectacular individual performance (but better than chance), organized into complementary families.

Unlike agglomerative clustering, AdaBoost does not first group experts into families and then obtain complementarity between families; instead, AdaBoost tries to exploit complementarity directly by choosing, at each step, the most complementary expert.

The AdaBoost algorithm that we use is inspired from the original one in [Freund 1997] with adaptations for TRECVid. It is very similar to that of [Wu 2003], however they applied it in a different context of TRECVid. It is also very similar to that used by [Tang 2008] in the 2008 edition of TRECVid, but they did not use it on such a large battery of experts as we do in our experiments.

For a particular concept, given the training set $(x_1, y_1), \ldots, (x_m, y_m)$ where x_i are the multimedia samples, and $y_i \in \{0, 1\}$ is the groundtruth of the sample x_i (y_i is 0 if x_i does not contain the concept, 1 if it does), the algorithm that we use to train the fusion is the following:

1. We initialize a set of weights D_1 where $D_1(i)$ is the weight of sample x_i:

$$D_1(i) = \begin{cases} \frac{0.5}{nPos}, & \text{if } y_i = 1 \text{ (a positive sample)} \\ \frac{0.5}{nNeg}, & \text{if } y_i = 0 \text{ (a negative sample)} \end{cases} \tag{5.1}$$

   where $nPos$ and $nNeg$ are the number of positive and negative samples respectively in the training set.

2. At iteration t ($t = 1, \ldots T$), we choose the input expert h_t that minimizes the weighted classification error $\varepsilon_t = \sum_{i=1}^{m} D_t(i) I(y_i \neq h_t(x_i))$. I is called the indicator function, and it gives the cost associated to the classification result of a sample being different than the groundtruth. In our case, $I(y_i \neq h_t(x_i)) = |y_i - h_t(x_i)|$, the absolute value of the difference between the classification score (between 0 and 1) and the groundtruth (0 *or* 1).

3. Compute the weight updating factor $\alpha_t = ln\frac{1-\varepsilon_t}{\varepsilon_t}$;

4. Update the weights of the samples according to:

$$D_{t+1}(i) = D_t(i)exp(\alpha_t I(y_i \neq h_t(x_i))) \tag{5.2}$$

   and normalize the weights for positive samples and for negative samples separately, so that $\sum_{i,y_i=1} = 0.5$ and $\sum_{i,y_i=0} = 0.5$ (always keep the total weight of positives and the total weight of negatives equal).

5. Repeat steps 2-4 until all input experts have been considered (each expert is only considered once).

6. At the end, the *strong expert* $H(x)$ will be a weighted sum of the weak experts chosen at each iteration t:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{5.3}$$

The functioning principle of the iterative AdaBoost algorithm is the following: at first, the chosen weak expert is the one with the lowest total classification error, all multimedia samples being considered equal in importance. This expert will not classify all samples correctly, it will make errors for some. But if for the next iteration, we increase the weights of samples that were classified incorrectly by the weak expert h_t (see Equation 5.2), therefore increasing these samples' importance, the next iteration will select a *complementary* weak expert h_{t+1}, that focuses on the samples incorrectly classified at the previous iteration. Because at each step, AdaBoost selects the expert that correctly classifies the multimedia samples for which the previous expert failed, it achieves *intra-concept complementarity at the context level*.

As for the weights of weak experts (inputs) in the final strong (fused) expert, weak experts that achieved low weighted errors at the iteration when they were chosen are given a larger weight in the final expert, while weak experts with larger errors are given a lower weight (it is assumed (and generally true) that the error ε_t is lower than 0.5).

For datasets with severe class imbalance (as is the case of the TRECVid SIN video dataset, in which, for many concepts, there are only a few tens of positives and hundreds of thousands of negatives), we have added the additional constraint that the total weight of positives and the total weight of negatives should have fixed values on 0.5 each, at every iteration, as in [Wu 2003], so that the classification result for true positives would still matter in the fusion.

Also for the case of TRECVid, we performed a similar expert preselection as for the agglomerative clustering fusion: we rejected experts with relevances less than 1 or less than 8 times that of the best expert for that concept, for similar reasons as in the case of the agglomerative clustering.

When training is complete, the resulting strong expert will be a weighted arithmetic mean of input experts, and this strong expert is applied on the evaluation dataset.

5.3.3 AdaBoost rank-based fusion

When quering a dataset for a particular concept, we receive a ranked list of multimedia samples, in descending order of their likelihood to contain the concept. Ideally, in this ranked list, all the true positives should be concentrated towards the beginning, and all the negatives should follow until the end of the list. The previous AdaBoost method was made to improve the classification scores, which would indirectly improve the ranked list. We now try to optimize directly the ranks of the true positives, by altering the indicator function (the cost function when a classification error appears).

We therefore propose the following indicator function: for a positive sample, the associated cost is equal to the number of negatives that are in front of it in the ranked list, divided by the total number of negatives; for a negative sample, the cost is zero (we don't care about the negatives, we just want the positives in front):

$$I(y_i \neq h_t(x_i)) = \begin{cases} \frac{negPreceeding}{nNeg}, & \text{if } y_i = 1 \text{ (a positive sample)} \\ 0, & \text{if } y_i = 0 \text{ (a negative sample)} \end{cases} \tag{5.4}$$

where *negPreceeding* is the number of negatives preceeding the positive sample in question in the ranked list, according to the weak expert h_t, and *nNeg* is the total number of negatives.

Of course, because the negatives do not matter any more as long as the positives are in front, the weight of the negative samples becomes meaningless, and we only work with the weight of positives. All of the other aspects concerning AdaBoost remain the same as for the previous method.

As with the agglomerative clustering fusion and the adaboost fusion based on scores, we perform similar expert selections before starting the actual fusion.

5.3.4 Weighted average of experts

As a reference for comparing the performances of the fusion methods presented so far, we consider a simple weighted average of the input experts, with weights given by the average precisions of experts on the training set, for the concept in question (the weights can vary from one concept to another, depending on how the experts react to the concepts). We can say that in the end, the other methods are also weighted means of experts, but with more elaborate ways of choosing the weights. We wish to compare the more elaborate methods with this simple baseline.

As with the other fusion methods presented so far, we perform similar expert selections before starting the actual fusion.

5.3.5 Best expert per concept

We add a second reference for evaluating the performance of our fusion methods, namely the best expert per concept. This method consists in simply choosing, for each semantic concept individually, the expert that gives the best average precision on the training set. This is our most basic reference when examining other methods, as the goal of fusions is to obtain gains compared to simply considering the best expert for the concept of interest.

5.3.6 Combining fusions

After applying all of the previous approaches in parallel, we now dispose of a battery of five fused experts: agglomerative clustering, score-based AdaBoost, rank-based AdaBoost, weighted average and best expert per concept. Our preliminary experiments have shown that for some concepts, some (or all) of the fusion methods degrade performance on the training set when compared to simply choosing that concept's best expert. To prevent this, we propose that for each concept, we see which of the fusion methods (including the best expert per concept) performs best on the training set, and *choose* that fusion method as the final result for that concept.

5.3.7 Improvements: higher-level fusions

So far, we have treated each concept independently, disregarding any relationship that may exist between concepts. However, the video shots from TRECVid result from the temporal segmentation of longer videos, therefore there may also exist temporal relations between shots.

We now propose to integrate these additional semantic and temporal relations, by considering two additional types of information:

- *temporal context information*, which we address using a *temporal re-scoring of shots*;
- *semantic context information*, which we address using *conceptual feedback*.

After the late fusion step, we dispose, for each concept, of the classification scores on all video shots. Because a concept that is present in a shot of a video also tends to be present in the neighboring shots of the same video due to temporal correlation, a *temporal re-scoring* of shots can be performed in order to take advantage of the temporal context (block ‘‘*Temporal re-scoring*” in Figure 2.1). The approach is described in [Safadi 2011] and was shown to give an increase in average precision.

After temporal re-scoring, we apply *conceptual feedback* on the classification scores with the algorithm from [Hamadi 2013]. This exploits the semantic relations between concepts by constructing a new descriptor with 346 dimensions (for the 346 semantic concepts of TRECVid 2011-2013), the i^{th} dimension of this descriptor being the classification score of the shot with the i^{th} concept. Supervised classification is applied on this descriptor as if it were a normal descriptor, and the resulting classification scores are re-fused with the previous results (block *“Conceptual feedback”* in Figure 2.1).

5.4 Experiments

We apply our fusion approach in two sets of experiments:

- First, we test the gains from this optimized fusion on the SIFT-based BoW descriptors employing retinal preprocessing, on the trajectory-based BoW descriptors and

on the combination of these two sets, in order to evaluate the gains from complementarity within and between these two groups of experts.

- Second, we apply the same fusion approach on an even larger and more diverse set of experts contributed by the entire IRIM group, providing an even richer description of the video shots and increasing performances even further.

5.4.1 Fusion of retina and trajectory experts

We have seen in Sections 3.3.2.4 and 4.3.3.3 that even simple late fusions such as arithmetic means of classification scores from different experts can improve concept detection on the TRECVid dataset.

We recall that in the case of retina-enhanced SIFT descriptors, on the TRECVid 2012y dataset, the arithmetic mean of experts *SIFT*, *SIFT retina*, *SIFT multichannel*, *SIFT retina masking* and *SIFT multichannel masking* gave a mean infAP of 0.1220, a 35% increase compared to the overall best-performing individual descriptor, *SIFT retina* (0.0904).

For trajectory BoW descriptors, also on the TRECVid 2012y dataset, the arithmetic mean of all the experts from Table 4.2 (but with a single version for the vocabulary sizes) gave a mean infAP of 0.0670, which is 55% better than the best of the input trajectory descriptors (see Section 4.3.3.3 for details).

We now apply the more complex late fusion method described in Section 5.3 to sets of retina-enhanced SIFT BoW experts and/or trajectory BoW experts. We train the late fusion on classification scores from the TRECVid 2012x dataset, and evaluate the fusion on classification scores from the 2012y dataset. All of the input experts use the same KNN supervised classifier from [Ballas 2012b].

5.4.1.1 Fusion of retina-enhanced SIFT BoW experts

We use *SIFT*, *SIFT retina*, *SIFT retina masking* and *SIFT multichannel masking* as fusion inputs. The descriptors are in an optimized form, having been subjected to a power transformation and Principal Component Analysis prior to supervised classification, as in [Ballas 2012b].

The results of fusing these 4 experts are shown in Table 5.1, column *"Ret."*. The *Agglomerative clustering fusion* performed the best, with a mean infAP of 0.1368, which is 12% better than the simple arithmetic mean with equal weights. However, the *Adaboost score-based fusion*, the *Weighted average fusion* and the *Selected best fusion* also give very close results. Only the *Adaboost rank-based fusion* has inferior performances, because ranks are very sensitive to small score variations, therefore the rank measure is unstable and the good fusion weights cannot be correctly determined. In any case, all fusion methods outperform simply taking the *best expert per concept* (a 33% increase for *Agglomerative clustering*).

We can also conclude that in the case of only 4 input experts, the more complex methods (*Agglomerative clustering*, *Adaboost score-based fusion* and *Selected best fusion*) give very close performances to the *Weighted average fusion*, which we recall to be a concept-per-concept weighted average of experts, with weights given by the average precisions of

Table 5.1: Mean (over all concepts) inferred average precisions of fusion approaches, for different sets of inputs: retina-enhanced SIFT descriptors (column *Ret.*), trajectory descriptors with normal (non-differential) BoW (*traj. norm.*), trajectory descriptors with differential BoW (*traj. diff.*), all trajectory descriptors (*traj. all*) and the full set of retina and trajectory descriptors (*full set*).

	Ret.	traj. norm.	traj. diff.	traj. all	full set
Adaboost score-based fusion	0.1366	**0.0805**	**0.0783**	**0.0828**	0.1264
Adaboost rank-based fusion	0.1147	0.0614	0.0578	0.0623	0.1249
Agglomerative clustering fusion	**0.1368**	0.0769	0.0746	0.0776	0.1274
Weighted average fusion	0.1363	0.0771	0.0748	0.0775	0.1013
Best expert per concept	0.1030	0.0583	0.0540	0.0582	0.1033
Selected best from 5 above	0.1346	0.0799	0.0776	0.0824	**0.1358**

experts for that concept. Therefore, for just a few input experts, choosing weights according to performance is enough to give an increase in infAP compared to an arithmetic mean with uniform weights.

5.4.1.2 Fusion of trajectory BoW experts

In this experiment, we fuse large sets of trajectory BoW experts. Descriptor optimizations in the form of power transformation and Principal Component Analysis prior to supervised classification, as in [Ballas 2012b], are not performed in this case, due to the too large number of descriptors to optimize.

We take all the experts from Table 4.2, including experts based on combined trajectory descriptions (and a few more combined representations not listed in this table, but with lower performances than those listed). We take all available vocabulary size versions and all versions of descriptors: with or without camera motion compensation, in classical or in differential BoW form. We perform three sets of fusions: one with non-differential BoW descriptors, one with differential BoW and the last one with all trajectory experts as inputs. In total, there are 144 experts, 72 for non-differential BoW and 72 for differential BoW.

Table 5.1 shows the mean inferred average precisions obtained from these fusions (columns *traj. norm.*, *traj. diff.* and *traj. all*). In all three cases, the *Adaboost score-based fusion* performs best, but the *Selected best fusion* is not far behind (because for most concepts, the *Adaboost score-based fusion* is selected anyway).

Although not perfectly comparable with the results from Table 4.4, which illustrate the results of arithmetic mean fusions with uniform weights, but which only use non-combined trajectory representations and only one version of vocabulary size per descriptor (see Table 4.2 for details), we can still see the performance boost given by the *Adaboost score-based fusion* (0.0828 infAP for *traj. all*) compared to the arithmetic mean (0.0670 for *"Fusion all"* from Table 4.4).

When comparing *Adaboost score-based fusion* with the simpler *Weighted average fusion*, this time on the same input experts for a fair comparison, we still have an increase of 7% (for *traj. all*) in favor of the first method, showing that the more complex *Adaboost*

score-based fusion makes sense when the number of input experts is very high, as it is for our trajectories.

As in the previous fusion of retina-enhanced experts, the *Adaboost rank-based fusion* does not perform so well due to the sensitivity of shot ranks to small score variations.

The *Agglomerative clustering* and *Weighted average* give close results, similar to the retina fusions, because the *Agglomerative clustering* resembles in behaviour to the *Weighted average* when the input experts are correlated (and they are, because they are all based on trajectories): the expert groups formed by *Agglomerative clustering* are very similar (but fewer in number) to their members, and the groups are fused through a weighted mean with weights given by the performance of each group.

As with the retina fusions, all methods manage to outperform simply choosing the *Best expert per concept*: a 30% for the *Adaboost score-based fusion* applied on *traj. all*.

As for normal BoW versus differential BoW, we see that the performances of differential BoW fusions are slightly lower (-3% for *AdaBoost score-based fusion*), but when fusing normal BoW experts with differential BoW experts *"traj. all"*, we do have a small performance boost of 3% compared to just fusing normal BoW. In practice, this small boost should be considered from an application framework point of view, as obtaining this boost requires doubling the amount of trajectory experts, which is not feasible for every system.

5.4.1.3 Fusion of retina *and* trajectory BoW experts

Overall, retina experts (and fusions) perform better than trajectories, but we wish to see whether or not the addition of trajectories can bring an additional performance increase compared to just using the retina-enhanced experts. To this end, we use as inputs for our fusions all 4 retina-enhanced experts and all 144 trajectory experts seen previously, for a total of 148 experts.

The results of fusions applied on this set are shown in Table 5.1, column *"full set"*. We see that for most fusions, the results are inferior to their correspondents from the *"Ret."* column. This time, the *Selected best fusion* is the best performer for this set of input experts, managing to improve performances compared to the other 4 fusions and the best expert per concept; however, this result is still inferior to the best one from the *"Ret."* column, which means that our fusion methods have not managed to improve performances by adding trajectories to the set of retina experts.

This lack of improvement is explained by the large imbalance between the set of retina experts (only 4) and the set of trajectory experts (144), which causes the contribution of trajectories to outweigh that of retina-enhanced experts.

Based on this remark, we modify our fusion by manually introducing an *intermediate hierarchical level*, based on our knowledge of the internal workings of each expert. We first fuse the set of retina experts using our previously-described approaches, and independently the set of trajectory experts usign the same methods. Afterwards, we perform an arithmetic mean (with equal weights) of the retina fusion and the trajectory fusion, thereby avoiding the imbalance problem between the two sets. We obtain the following results:

- *Selected best fusion* of *retina* + *Selected best fusion* of *trajectories*: **0.1427** infAP,

which constitutes an improvement of 6% compared to the retina fusion. Compared to the best of the input elementary experts per concept, the total increase is 38%;

- *AdaBoost score-based fusion* of *retina + AdaBoost score-based fusion* of *trajectories*: **0.1445** infAP, again an improvement of almost 6% compared to the retina fusion;

Therefore, including our knowledge of the internal workings of experts has helped us to better fuse information from the *complementary* retina-enhanced SIFT BoW experts and trajectory BoW experts. The result could be further improved by optimizing the weights when combining the retina fusion and of the trajectory fusion, but in order to avoid overfitting, this would require splitting the dataset even more, complicating our experimental setup.

5.4.1.4 Preliminary conclusion

Our proposed information fusion strategies can significantly improve semantic concept detection by taking advantage of complementary information coming from different experts. Fusing complementary retina experts gives good results, and fusing trajectory experts also gives a significant improvement. Fusing retina *and* trajectory experts gives an additional gain, but fusing normal BoW with differential BoW gives only a small gain at the cost of doubling the amount of data.

We have also seen that including human knowledge about experts to construct a hierarchical fusion framework can further improve results, as it will be confirmed in Section 5.4.2 by the *manual hiearchical fusion* of [Ballas 2012b] on an even more diverse set of descriptors. However, manually specifying expert groups is cumbersome for large sets of experts, and our automatic fusion approaches will be shown to still give good results on a diverse, large set of experts.

5.4.2 Fusion of diverse IRIM experts

We have seen the performance gains obtained when fusing SIFT-based BoW experts using retinal preprocessing and trajectory-based BoW experts, now it is time to apply our fusion method on an even more diverse set of experts. We perform this experiment on the TRECVid SIN 2013 dataset, for which the IRIM partners have provided a large and diverse set of descriptors and descriptor versions on which supervised classifiers were trained, resulting in a large battery of elementary experts. The TRECVid SIN dataset is split in two parts, the first one (dev or 2013d), for training the fusion parameters, and the second one (test or 2013t) on which we evaluate performances using the official TRECVid measure, the *mean inferred average precision* [Yilmaz 2006, Yilmaz 2008].

5.4.2.1 Input data for fusions: elementary experts

Recalling the processing chain from Figure 2.1, the first step for semantic indexing is to extract descriptors from the video shots. For its participation in the TRECVid challenge, the laboratories that form the IRIM group have all shared their descriptors, creating a very

rich and multimodal representation of the video shots. The IRIM partners have contributed many descriptors and descriptor versions, and a full listing of them is beyond the scope of this work. Instead, we will just list some of the main descriptors, without going into details.

Color descriptors: A large family of color descriptors was submitted by ETIS, with color represented in the Lab color space, with an optional spatial division of the keyframe [Gosselin 2008]. A color histogram in the RGB color space was also submitted by LIG.

Contour and texture descriptors treating the keyframe globally: ETIS also contributed quaternionic wavelets, which are a texture descriptor, also with an optional spatial division of the keyframe [Gosselin 2008]. A normalized Gabor transform of the keyframe was contributed by LIG, as well as an early fusion of their RGB color histogram and this normalized Gabor transform.

Descriptors constructed from local spatial features: There were many descriptors employing a BoW model of various local features. BoW of Opponent SIFT features were contributed by LIG in versions with keypoints either from a Harris-Laplace corner detector, or from a dense grid [van de Sande 2010]. From the same family, CEALIST contributed BoW of dense SIFT with spatial pyramids [Shabou 2012, Ballas 2012a].

We contributed BoW of dense SIFT employing retinal preprocessing [Strat 2012a, Strat 2013a, Strat 2013b]: *SIFT*, *SIFT retina*, *SIFT retina masking* and a version of *SIFT multichannel masking*.

BoW descriptors based on Local Binary Patterns were contributed by LIRIS [Zhu 2013], and texture local edge patterns enhanced by color histograms [Zhu 2013] were contributed by CEALIST. Multi-level histograms of multi-scale LBP with spatial pyramids were contributed by LSIS [Paris 2010].

Vectors of locally-aggregated tensors (VLAT) [Negrel 2012], which also deal with local SIFT features clustered on a visual vocabulary, but use a pooling mechanism different than BoW to generate image signatures, were submitted by ETIS.

Saliency moments, a descriptor that exploits the shape and contours of salient regions [Redi 2011b], was submitted by EURECOM.

Spatio-temporal descriptors: BoW of space-time interest points, described with histograms of oriented gradients or with histograms of optical flow, as in [Laptev 2005], were submitted by LIG.

EURECOM submitted spatio-temporal edge histograms, based on temporal statistics of the (2D) MPEG-7 edge histogram.

Descriptors based on tracking and describing faces in successive frames (face tracks) were submitted by LABRI.

Some of our SIFT-based BoW descriptors employing the retinal model are spatio-temporal, namely *SIFT retina masking* and the version of *SIFT multichannel masking*.

Trajectory descriptors: We submitted 5 of the best-performing trajectory BoW descriptors, using the following descriptions for trajectories (in non-differential BoW version):

- the BRIEF descriptor of each trajectory's starting point; k-means clustering on 256 vocabulary words;
- the BRIEF descriptor concatenated with a histogram of displacement without a zero-bin (without null-speed); 1024 vocabulary words;
- the BRIEF descriptor concatenated with a histogram of displacement without a zero-bin and with histogram of displacement without a zero-bin but in camera motion compensated version; 1024 vocabulary words;
- concatenation of (1)-(8) from Table 4.1, with and without camera motion compensation; 384 vocabulary words;
- concatenation of the histogram representations ((1)-(4) from Table 4.1), with and without camera motion compensation; 256 vocabulary words;

Audio descriptors: Audio descriptors in the form of a BoW of Mel-frequency cepstral coefficients (MFCC) were contributed by LIRIS.

Highly-semantic descriptors: Detection scores of various semantic concepts from the ILSVC and ImageNet datasets [Deng 2009] (and with detectors trained on ImageNet) were assembled to form descriptors by XEROX [Sánchez 2013]. Individually, these gave the best-performing experts.

From the same family of highly-semantic descriptors, LIF contributed a descriptor based on detection scores for a set of 15 mid-level concepts called "percepts" [Ayache 2007].

As we can see from the list above, we have a very rich and diverse description of the video shots, therefore encouraging fusion approaches.

Before supervised classification, most of the descriptors went through an optimisation consisting in applying a power transformation to normalize the values of the descriptor dimensions, followed by Principal Component Analysis (PCA) to make each descriptor more compact, and at the same time, more robust [Safadi 2013], corresponding to the *"Descriptor optimization"* block in Figure 2.1.

The next step was to train and apply supervised classification algorithms (classifiers) on each of the (optimized) descriptors (*"Supervised classification"* in Figure 2.1). A classifier gives, for each concept and for each video shot, the estimated "likeliness" of the shot to contain the concept (a classification score between 0 and 1).

Two classifiers were applied to each video shot descriptor. The first one is based on a K-Nearest Neighbours search [1]. The second one, called MSVM, applies a multiple learner approach based on Support Vector Machines [Safadi 2010]. MSVM generally performs better than KNN, but it is more computationally expensive [Ballas 2012b].

[1] `http://mrim.imag.fr/georges.quenot/freesoft/knnlsb/index.html`

KNN and MSVM classifiers applied to a given descriptor constitute two different elementary experts. These can be combined (or fused) into a first level non-elementary expert. The combination can be done in a number of ways. For this first level, we use a weighted mean of classification scores, the weights between KNN and MSVM being their infAP performance estimated by cross-validation within the training (dev) set. The corresponding expert is called *FUSEB*; it is most often better than either KNN or MSVM. We later *use the FUSEB experts as elementary ones* for the next steps in our proposed late fusion approaches.

The most numerous family of FUSEB experts is that of ETIS color histograms in the Lab color space (12 experts), while their quaternionic wavelets family numbered 9 experts. We ourselves contributed in total 11 SIFT-based BoW experts, some with and some without retinal preprocessing, and 5 experts using trajectories. 6 OpponentSIFT BoW experts from LIG were also used, as well as two more dense SIFT experts from CEALIST. There were 5 experts based on percepts, while the experts corresponding to the remaining descriptors from the previous list were less numerous (only one or two).

5.4.2.2 Results on TRECVid 2013

All of the compared fusion methods are tested using the same input elementary experts, the FUSEB experts for the descriptors listed in Section 5.4.2.1. The experts' supervised classifiers are trained on 2013d and applied on 2013t. The fusions are also trained on experts from 2013d, and fusion results are evaluated on 2013t. In the case of parameter optimizations for experts or fusions, they are done in cross-validation on 2013d.

We report mean infAP averaged over a subset of 38 concepts out of the total 346, the same concepts that are used for evaluating official TRECVid SIN 2013 submissions.

For comparison, we also include results from a *manually-optimized hierarchical late fusion* [Strat 2012b] of the same experts contributed by the LIG laboratory. In this approach, expert groups are chosen manually, but in a hierarchical manner, on more levels than the agglomerative clustering. The multi-level hierarchy starts by fusing different variants of the same descriptor (e.g. BoW of the same local descriptor but with different dictionary sizes). Afterwards, it fuses the experts corresponding to different image spatial decompositions (pyramid) if available. Finally, the last level concerns descriptors of different types within the same modality (e.g. color, texture, interest points, percepts or faces) and descriptors from different modalities (audio and visual).

Global results: Table 5.2 (column *"basic"*) shows the mean infAP obtained by the proposed fusion methods. The *manual hierarchical fusion* performs the best, thanks to the carefully-optimized weights of experts, the additional score normalization steps between fusion stages and the manual grouping of experts that ensures more homogeneous properties within a group.

Among the automatic methods, the *Adaboost score-based fusion* performs the best, with performances not far behind the manually-optimized hierarchical fusion. The *Adaboost rank-based fusion* performs less good, because the rank of a shot can vary greatly with small variations in the classification score, which makes the method more sensitive to

Table 5.2: Mean (over all concepts) inferred average precisions of fusion approaches: basic (without any post-processing), +RS (with temporal re-scoring, *temporal context* integration), +RS+CF (with RS followed by conceptual feedback, *semantic context* integration), +RS+CF+RS (+RS+CF followed by a second RS).

	basic	+RS	+RS+CF	+RS+CF+RS
Manual hierarchical fusion	0.2576	0.2695	0.2758	0.2848
Adaboost score-based fusion	0.2500	0.2630	-	-
Adaboost rank-based fusion	0.2346	0.2534	-	-
Agglomerative clustering fusion	0.2383	0.2516	-	-
Weighted average fusion	0.2264	0.2409	-	-
Best expert per concept	0.2162	0.2367	-	-
Selected best from 5 above	0.2495	0.2631	-	-

classification noise. The *agglomerative clustering fusion* is relatively close in global results to the *Adaboost rank-based fusion*. Among the fusion methods, the *weighted average fusion* is the least good, showing that a greater performance boost can be obtained with more careful expert weight choosing strategies; for example, the *Adaboost score-based fusion* performs 10% better than the weighted average.

In any case, it can be seen that whatever the fusion method, the global result is always better than what would have been obtained if we would have taken, for each concept, its best expert on the training dataset (*Best expert per concept*). The *manual hierarchical fusion* is 19% better, the *Adaboost score-based fusion* is 16% better and the even the *weighted average* has a 5% improvement, proving that late fusion schemes, even naive ones, generally improve concept detection performances.

The *selected best fusion* selects, for each concept, the fusion approach (among *Adaboost score-based fusion, Adaboost rank-based fusion, agglomerative clustering, weighted average* and the *best expert for that concept*) that performed the best on the training set. The *Adaboost score-based fusion* was by far chosen the most often, for 230 out of the 346 concepts, which is in agreement with it having the highest mean infAP. The *Adaboost rank-based fusion* was chosen for 60 concepts, the *agglomerative clustering* for 14 concepts and the *weighted average* for only 8 concepts. For the rest of the 34 concepts, the *best expert* was chosen, because the fusions were found to degrade performances on the training dataset. Considering this, it was to be expected that the mean infAP of the *selected best fusion* would be close but slightly above that of the *Adaboost score-based fusion*. However, no global gain is observed for the *selected best fusion*, because the choices made on the training set are not always the best also for the test dataset, due to variations between the two datasets.

Concept-per-concept results: Moving on to a concept-per-concept analysis, Table 5.3 shows the infAP gains for the 38 semantic concepts used in the official TRECVid 2013 evaluation, when comparing the best of the automatic methods (the *Adaboost score-based fusion*) with the baseline *best expert per concept*. For the majority of concepts, the fusion

gives a significant performance boost (such as for *Airplane, Bus, Hand, Running, Throwing*). For some concepts, the boost is not too high, especially for concepts that already have large infAP to start with (such as *Beach, Government leader, Instrumental musician, Skating*); this happens when the other experts do not bring any pertinent and complementary information compared to the best expert. There are only 6 concepts that experience performance degradations from the fusion, namely *Animal, Computers, Explosion or fire, Female face closeup, Girl and Kitchen*.

As a preliminary conclusion, we can say that fusing a large battery of complementary experts yields a significant performance increase. It is now time to examine the gains of higher-level fusions, at the temporal and semantic context levels.

5.4.2.3 Results for higher-level fusions

Table 5.2, column *"RS"* shows the mean infAP after applying the temporal re-scoring algorithm made by our partners in IRIM [Safadi 2011], briefly described in Section 5.3.7. The best-performing method, the *manual hierarchical fusion* also by our partners, has gain of 4,6%, while our methods also experience gains in the range of 5-10%. This shows that the temporal context can also bring useful information, resulting in a performance increase for all methods.

After temporal re-scoring, we apply the conceptual feedback step from our partners inside IRIM [Hamadi 2013], briefly described in Section 5.3.7 (*+RS+CF* in Table 5.2). Because of the significant computational cost, we limit this experiment to the best-performing method, the *manual hierarchical fusion*, for which an additional gain of 2,3% is obtained compared to the previous result. Adding a second temporal re-scoring step after the conceptual feedback (*+RS+CF+RS*) increases results by another 3,3%. In the end, the successive temporal re-scoring and conceptual feedback steps give an increase of 10,5% compared to the basic approach.

5.5 Conclusion

In this chapter, we proposed several methods of combining dozens of input experts into better ones, and applied these methods in the context of the *TRECVid Semantic Indexing* task.

On the TRECVid 2012 SIN task, we have shown that such fusion methods can better exploit the complementarity between SIFT-based BoW descriptors utilizing retinal preprocessing and trajectory-based BoW descriptors, leading to a performance improvement greater than that of a simple arithmetic fusion with uniform weights, as it was done in Chapters 3 and 4.

On the more diverse set of experts from the TRECVid 2013 dataset, we have shown that all of the fusion methods globally outperform taking the best expert for each concept, and that more elaborate fusions can perform better than a naive weighted arithmetic mean. Our automatic late fusion approach based on AdaBoost performs almost as good as a manually-optimized hierarchical fusion, without having a large computational cost. We have also

Table 5.3: Comparison of inferred average precisions for the *best expert per concept* and the *AdaBoost score-based fusion*, for particular concepts.

concept	best expert	AdaBoost sc.	rel. gain (%)
Airplane	0.0573	0.0923	61
Anchorperson	0.4850	0.5988	23
Animal	0.0659	0.0078	-88
Beach	0.4658	0.4722	1
Boat or ship	0.2907	0.3083	6
Boy	0.0291	0.0316	9
Bridges	0.0372	0.0393	6
Bus	0.0273	0.0598	119
Chair	0.1621	0.2394	48
Computers	0.2647	0.1919	-28
Dancing	0.2990	0.4019	34
Explosion or fire	0.1780	0.1617	-9
Female face closeup	0.3741	0.3550	-5
Flowers	0.1752	0.1895	8
Girl	0.0462	0.0360	-22
Government leader	0.4387	0.4546	4
Hand	0.1532	0.2847	86
Instrumental musician	0.5141	0.5782	12
Kitchen	0.1072	0.0952	-11
Motorcycle	0.1778	0.2369	33
News studio	0.7213	0.8223	14
Old people	0.3719	0.4096	10
People marching	0.0388	0.0470	21
Running	0.0863	0.1405	63
Singing	0.1096	0.1459	33
Sitting down	0.0003	0.0023	667
Telephones	0.0063	0.0133	111
Throwing	0.1121	0.2506	124
Baby	0.1317	0.2234	70
Door opening	0.0369	0.0410	11
Fields	0.0753	0.1375	83
Flags	0.2607	0.2819	8
Forest	0.0911	0.1150	26
George Bush	0.6092	0.6624	9
Military airplane	0.0172	0.0381	122
Quadruped	0.0807	0.1133	40
Skating	0.4956	0.5328	8
Studio with anchorperson	0.6228	0.6871	10

shown that additional fusions, at the temporal and semantic context levels, can give an additional performance boost.

Even though we experimented on the TRECVid SIN video dataset, these late fusion approaches are generic and can be extended to other multimedia collections as well.

Chapter 6

Conclusions and perspectives

Contents

Our work explored the topic of automatic semantic indexing of highly-diverse video datasets. We have taken a state of the art semantic indexing framework (Figure 2.1) which we have enriched with spatio-temporal descriptions and with information fusion methods.

Our experiments have shown that the proposed retinal preprocessing approaches lead to a set of better-performing, complementary spatio-temporal descriptors, which are a good compromise between computational demands and semantic indexing performance. The spatio-temporal diversity of the descriptor set was then pushed even further towards motion description thanks to the inclusion of Bags of Words of trajectories of tracked points, which have also proven themselves as valid methods not only on the motion-oriented KTH dataset, but also on the extremely diverse TRECVid SIN dataset.

In the end, the availability of such a set of diverse spatio-temporal descriptors, along with other various and complementary descriptors contributed by the IRIM partners, has motivated us to develop automatic late fusion methods. These late fusion methods have allowed us to benefit from the joint information brought by the various descriptors and to significantly improve the overall semantic concept detection performance.

6.1 A retrospective of contributions

6.1.1 Retina-enhanced SIFT BoW descriptors

We have shown that the two retinal outputs, the parvocellular and magnocellular channel, can help us enhance classical SIFT/SURF Bag of Words descriptors. The parvocellular channel's "cleaning" effect, which reduces noise and compression artifacts, normalizes colors and enhances local contrast, can lead to more accurate local feature signatures, which in turn give a higher-quality Bag of Words histogram (*SIFT retina*). The magnocellular channel on the other hand can be used as base for a low-cost detector of areas of interest, thereby guiding feature selection only to such potentially more relevant areas, again improving concept detection results (*SIFT retina masking*).

While the approaches above remain mostly oriented towards spatial appearance, with only the possibility to orient *spatial* feature collection on moving areas, we also employ the magnocellular channel in a second way. This time, we truly integrate motion information in the form of SIFT signatures collected on the magnocellular channel. Because the magnocellular channel responds to contours perpendicular to the motion direction, this gives us information about the local motion around a feature point. Concatenating a local OpponentSIFT spatial appearance signature from the parvocellular channel with the SIFT signature from the same location on the magnocelular channel thus gives spatio-temporal *multichannel* feature descriptions. Coupled with area of interest masking (*SIFT multichannel masking*), this type of descriptor becomes very interesting, as it reacts especially well to concepts related to motion in TRECVid (concepts often related to sports activities).

Even if the parvocellular preprocessing of keyframes leads to the best global result, it still makes sense to keep the other descriptors, because together they form a *complementary set*. Complementarity is maximum expecially between the baseline SIFT BoW keyframe descriptor (*SIFT*), the parvocellular preprocessed keyframe BoW descriptor (*SIFT retina*) and the multichannel descriptor employing area of interest masking (*SIFT multichannel masking*).

The two keyframe-based descriptors are complementary because *SIFT retina* works better when there is no significant motion (the parvocellular channel introduces motion blur) and the retina can nicely clean the image, while *SIFT* works better when there is motion, because it is much less affected by motion blur. The keyframe-based descriptors on the other hand are complementary with *SIFT multichannel masking* because the former focus only on spatial information, while the latter is spatio-temporal.

We have then shown that we can exploit such complementarity through a late fusion of classification scores coming from these complementary descriptors, thereby significantly increasing performance.

6.1.2 Trajectory BoW descriptors

Regarding motion representation, we have proposed a large battery of trajectory-based BoW descriptors, based on various trajectory descriptions. Our preliminary tests on the KTH dataset showed that even though we do not obtain performances as high as the state of the art, we are still well above the chance level, thereby validating the functioning of out method. We must also not forget the fact that we did not perform any parameter optimisation whatsoever, following the idea that optimising for the very restricted KTH dataset is not our goal anyway, since we are interested in semantic indexing of generic databases.

On the very difficult TRECVid dataset, despite again making only limited optimisations (due to the high computational cost of running each experiment) and despite not being in an action recognition context (most of the concepts are only distantly, if at all related to motion), the trajectories performed better than chance for 129 out of the 346 concepts.

As with the retina-enhanced SIFT BoW descriptors, we have shown that our trajectory descriptors form a complementary set, on which information fusion can be applied to obtain a significant average precision gain.

6.1.3 Late fusion of experts

Different descriptors, especially if they are from different families, generate complementary experts. Remaining in the same semantic indexing framework, this motivates us to contribute (and compare) several late fusion strategies, by which classification results from several experts are combined, improving concept recognition.

We have shown that in this context, a version of AdaBoost adapted to the data imbalance problem (too many negative samples compared to positive ones) of TRECVid generally leads to the highest performance increase, without having a high computational cost. This tendency has been confirmed when fusing retina experts, trajectory experts, retina and trajectory experts together, as well as even more experts contributed by other members of the IRIM group. The advantage of our approaches is that they are completely automatical, not requiring the user to specify expert groups or weights manually.

Our fusion methods have been shown to perform almost as well as a manually-optimized hierarchical fusion when there is no severe imbalance between expert families. When such severe imbalance exists, as was the case when fusing the 4 retina experts with the battery of 144 trajectory experts, family imbalance tends to shift the mean infAP towards those of the dominant family. However, we have shown that manually introducing a hierarchical level that forces the fusion algorithm to first fuse family members inside families, and only afterwards to fuse different families, brings the fusion back on track so that a performance boost is obtained.

6.2 Perspectives for future research

The retina-enhanced SIFT descriptors that we tested on the TRECVid 2012 dataset employed local features only at the original image scale, since the preliminary (but unoptimized) experiments with SURF performed better in a single-scale configuration. However, we think that information at multiple scales can bring added value to the descriptors, if properly configured. We are in the process of extending our descriptors to collect features at multiple scales and the first results are encouraging, but we have yet to optimize the parameters of the dense grid, of the retina and of the temporal window.

A second direction of study for retinal preprocessing is to experiment with other types of local features such as FREAK [Alahi 2012], in order to verify if the similar behaviours as for SURF and SIFT are observed. FREAK would be especially interesting to try, since it is also a bio-inspired model, and its interactions with our bio-inspired retina might lead to even better results.

We also know that the retinal outputs are affected by camera motion. The parvocellular channel suffers from motion blur, thereby degrading the quality of local spatial feature descriptors. The magnocellular channel on the other hand will give strong responses on all contours perpendicular to the motion direction. For the multichannel descriptors, this means that the local features from the magnocellular channel no longer reflect the real local motion, instead being polluted by camera motion. As for salient blob selection, camera

motion resulting in a moving background means that a lot of background features will be selected, which are generally not so meaningful.

This motivates us to pursue a version of the retina model that is robust to camera motion. We plan to do this by feeding camera motion estimation data (from our trajectory tools) into the retina at each frame, changing the functioning of retinal filters so that they combine other pixels than normal, in such a way as to eliminate the effects of camera motion both in the parvocellular and in the magnocellular channel. Only the real movement of the objects in the scene will still cause motion blur or magnocellular responses, but not camera motion. Based on a dense optical flow map, we could even go a step further and construct a retina model in which the parvocellular channel is immune to all motion, both foreground and background.

This motion compensation is in fact similar to what the human visual system can achieve with eye movements, whereby we as humans are capable of stabilising the image in our eyes by compensating for head motion, and we are also capable of following the motion of an object of interest. This helps the retina to work in its most favourable context, enhancing details on objects of interest and blurring only uninteresting items (which we do not follow with our eyes).

In the long run, we also plan to further extend our biologically-inspired approach by including higher levels of processing, such as those occurring in the primary cortex V1, that could serve for object or face recognition and tracking [Benoit 2010]. It would be very interesting if we could also model even higher cortical areas, which would in fact mean to simulate an important part of the visual brain. However the functioning of these higher areas is not yet well understood [Hérault 2010].

Regarding trajectory BoW descriptors, there is still room to optimize their configuration as well, such as the tracker parameters or the bins used for histograms of motion and acceleration directions. Other trajectory descriptions such as Fourier transforms or wavelet representations of motion vectors should also be experimented with. Fourier transforms can describe a trajectory in a manner that is robust to the exact starting or ending moment, while wavelet analysis can give better multiresolution information than just resampling the motion vectors to either 8 or 16 samples.

The temporal granularity of trajectories is also a matter which should be further investigated. Currently, we use trajectories of durations varying between 0.5 and 2 seconds, including the length of a trajectory explicitly in the trajectory descriptions (which are otherwise invariant to trajectory length and duration). Alternatively, trajectories can be grouped into separate Bags of Words corresponding to different duration intervals, such as a BoW for short movements and one for long movements. Even long trajectories can be cut into smaller pieces and used to populate the short-duration BoW. These BoW will hopefully be complementary and after a fusion step, will hopefully give better performance than a single, mixed-duration BoW.

Further advances can be made in the interaction between the retinal preprocessing and the extraction of trajectories. Not only can the estimation of camera motion from trajectories help improve the retinal response, but vice-versa, the segmented transient blobs described in Chapter 3 can potentially help isolate more interesting trajectories. This is

especially true if the segmented blobs are robust to camera motion, meaning that we can select trajectories only from interesting moving foreground objects.

The Differential Bag of Words descriptors, for highlighting less common visual words corresponding to short-duration actions, also have room for improvement. It would be interesting to experiment with the tiny activity analysis algorithm from [Zhang 2013] to select areas where small actions occur, and to eventually combine it with our Differential BoW to emphasize the moments when less common trajectory-words appear.

Later on, it would also be interesting to experiment with more complex models than the Bag of Words for trajectories, such as the Actom Sequence Model of [Gaidon 2011], which is capable of encoding the temporal succession of types of trajectories. An added benefit that can be obtained with the Actom Sequence Model would be the possibility to localize actions in longer videos, not just to say whether or not a pre-segmented video shot contains an action or not.

All of these experiments should also be carried out on other more action oriented datasets, such as Hollywood 2 [Marszalek 2009], in order to get a better idea of the performance in a wider range of applications and contexts.

Finally, our late fusion methods can be easily applied to other multimedia (not just video) databases, because they are independent of the type of the inner workings of experts. For example, it would be interesting to try these information fusion methods on an image dataset such as ImageCLEF[1].

As a conclusion, the topics explored within this thesis open many directions, from spatial and temporal description of video content to the fusion of those descriptors, in order to improve the semantic level of automatic indexing systems.

The proposed approaches have interesting properties that present a great generalisation potential, and after optimisations in the short run, they can be generalized and enriched in the long run.

From an application point of view, this work can be useful in many areas, such as automatic semantic annotation of videos uploaded to on-line databases, indexing of archived video transmissions from television networks, video-on-demand applications in which a user searches for videos that are visually-similar to a query sample, or even video surveillance applications, in which suspicious objects, persons or events could be detected.

[1] http://www.imageclef.org/

Chapitre 7

Résumé

Contents

7.1 Introduction

7.1.1 L'explosion multimédia

Durant ces dix dernières années, notre société a connu des avancées importantes dans les domaines de l'électronique et des technologies numériques. Ces avancées sont, comme toujours, accompagnées d'une baisse constante du prix des appareils électroniques. En 2013, quasiment tous les téléphones portables sont dotés d'un capteur optique capable d'enregistrer des photos ou des vidéos d'une qualité toujours croissante. Aussi, les appareils photo ou caméras vidéo numériques dédiés sont accessibles à presque tous, dans les pays développées.

Cette augmentation du nombre de dispositifs capables d'acquérir du contenu multimédia à qualité croissante est aussi corrélée à l'augmentation de l'espace de stockage disponible, sous la forme de disques durs, cartes mémoire, disques DVD et Blu-Ray. Ainsi, aujourd'hui, beaucoup de gens disposent de collections multimédia personnelles impressionnantes.

A ces collections stockées sur les ordinateurs personnels, s'ajoutent aussi des sites web (comme Facebook, Instagram, YouTube etc.) qui permettent aux utilisateurs de déposer et *partager* leurs créations avec d'autres, et permet aussi de spécifier des étiquettes aux contenus (des labels comme le nom des personnes présentes dans une photo, le lieu où la photo a été prise, l'évènement etc.).

7.1.2 La nécessité d'organiser

Avec autant de fichiers multimédia, il devient difficile pour les utilisateurs de retrouver rapidement un certain élément déposé quelques temps auparavant. La collection multimédia doit être très bien organisée (par exemple par date d'acquisition de chaque photo, par évènement représenté, par lieu d'acquisition etc.). Une telle organisation pourrait être faite en ajoutant à chaque photo des informations complémentaires sous la forme d'*étiquettes sémantiques*.

Déjà, les appareils photo ou vidéo sont capables d'ajouter certaines informations automatiquement, par exemple la date et l'heure d'acquisition, et certains modèles peuvent aussi intégrer les coordonnées géographiques du lieu d'acquisition (en utilisant un module GPS intégré), mais cette information n'est pas toujours suffisante pour retrouver un certain élément multimédia. L'utilisateur doit alors introduire lui-même des informations supplémentaires. Par exemple, une photo avec la description "maman prépare un gâteau" pourrait avoir les étiquettes suivantes : "maman", "cuisiner", "cuisine", "gâteau" qui pourraient être utilisées comme mots-clés pour une recherche automatique dans la collection multimédia. Le fait d'introduire manuellement de telles descriptions et mots-clés demande un temps important et certainement ennuyeux à l'utilisateur.

Les collections multimédia en ligne ont des problèmes similaires, mais à une échelle encore plus grande. Par exemple, sur le site YouTube[1], 100 nouvelles heures de contenu

[1] `www.youtube.com`

sont mises en ligne chaque minute par la totalité des utilisateurs[2]. Si ces vidéos ne sont pas proprement labélisées par les utilisateurs (avec un titre et des mots-clés représentatifs), elles ne pourront pas être trouvées par les autres, et le fait de les avoir mis en ligne perd son sens. Actuellement, cette labélisation doit être faite manuellement par les utilisateurs. De plus, si la fonctionnalité "pouvoir retrouver une certaine sous-partie d'une vidéo" est demandée, la vidéo doit être annotée non seulement à un niveau global, mais au niveau de ces sous-parties (sous-séquences vidéo, "shots"), ce qui prend encore du temps.

Par conséquent, un système capable de faire une *indexation sémantique automatique* du contenu multimédia prend du sens, et évitera à l'utilisateur de passer des heures à annoter sa collection manuellement. Un tel système, appelé *système d'indexation multimédia par le contenu*, utilise des techniques de vision par ordinateur pour analyser le contenu multimédia et détecter automatiquement des divers concepts sémantiques (type de contenu, lieu où se passe l'action, qui sont les participants, quels objets sont présents dans la scène, s'il y a des éléments inhabituels etc.).

7.1.3 Contexte des travaux et contribution

Le but de cette thèse est l'indexation sémantique automatique de collections vidéo. Ceci est un sujet à la frontière de plusieurs domaines, comme illustré dans la Figure 7.1. Il nécessite des compétences dans les domaines de la vision par ordinateur, le traitement et l'analyse d'images, l'apprentissage automatique et la fusion d'informations :

- Les techniques de traitement et analyse d'images et de vidéos sont utilisées pour extraire d'une vidéo des descripteurs d'un niveau sémantique très bas (couleurs dominantes, orientations des contours, directions du mouvement etc.) ; celles-ci caractérisent une vidéo dans une forme compacte et compréhensible par l'ordinateur.

- La vision par ordinateur sert à agréger les descripteurs ci-dessus dans des représentations (comme la représentations par Sac-de-Mots vue dans la section 2.3.4).

- L'apprentissage automatique sert à déterminer automatiquement les liaisons entre les descripteurs (ou descripteurs agrégés) et les concepts sémantiques présents dans la vidéo. Par exemple, une couleur dominante verte peut indiquer la présence de végétation.

- A la fin, les techniques de fusion d'informations peuvent combiner les données issues de plusieurs sources (couleurs, contours, mouvement etc.) pour améliorer la détection de concepts sémantiques.

Notre but est de concevoir un système d'indexation sémantique automatique très générique, capable de travailler avec n'importe quel type de vidéo et d'annoter une très large gamme de concepts sémantiques (objets divers, actions, personnes, situations etc.). Cela signifie qu'il n'est pas possible d'utiliser des détecteurs spécialement conçus pour chaque concept sémantique, mais d'utiliser des détecteurs génériques, ce qui augmente la difficulté d'obtenir de bons résultats.

[2] http://www.youtube.com/yt/press/statistics.html

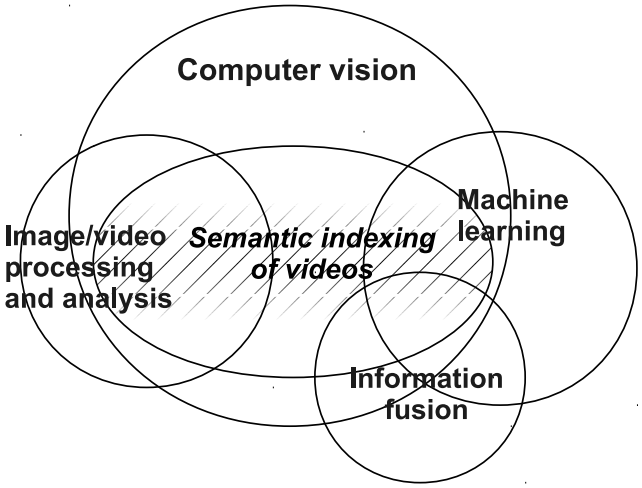


FIGURE 7.1 – Domaines scientifiques concernées par cette thèse : l'indexation sémantique des vidéos nécessite des techniques de vision par ordinateur, traitement et analyse d'images et vidéos, apprentissage automatique et fusion d'information.

La conception *intégrale* d'un système d'indexation automatique est une tâche très difficile qui ne peut pas être achevée pendant la durée d'une thèse. Pour cette raison, nous nous basons sur le système d'indexation développé au sein du consortium français IRIM[3] (décrit dans la Section 2.2). Nous améliorons plusieurs aspects de ce système, ce qui constitue notre contribution dans cette thèse :

- La première contribution est dans le domaine du traitement et de l'analyse d'images et vidéos. Nous proposons une méthode pour *améliorer les descripteurs d'image standard basés sur la représentation des gradients d'intensité, comme SIFT [Lowe 2004a] ou SURF [Bay 2008]*, dans le but d'améliorer leur généricité et leurs résultats pour la détection de concepts. Cette méthode est basée sur un *pré-traitement des trames vidéo en utilisant le modèle d'une rétine biologique* de [Benoit 2010]. Les descripteurs SIFT/SURF sont basés sur des histogrammes des orientations des gradients spatiaux de l'intensité lumineuse, donc ils sont des descripteurs purement spatiaux. Le pré-traitement rétinien améliore les résultats globaux de détection de concepts, mais il étend aussi les descripteurs de type SIFT/SURF en intégrant des comportements *spatio-temporels*. Cette méthode est présentée dans le Chapitre 3.

- La deuxième contribution est aussi dans le domaine du traitement et de l'analyse de vidéos. Elle consiste en une *batterie de descripteurs de trajectoires, dédiés à la représentation du mouvement*. Ces descripteurs, inspirés de l'état de l'art, sont présentés dans le Chapitre 4. De cette façon, nous avons pour commencer une description purement spatiale, suivie d'une description spatio-temporelle, finalement complétée avec une description temporelle du contenu vidéo. Ce qui nous amène à notre dernière contribution.

[3] http://mrim.imag.fr/en/

- La troisième contribution est dans le domaine de la fusion d'information. Comme nous avons maintenant une description très riche du contenu vidéo, nous exploitons la *complémentarité* entre les différents descripteurs en faisant des *fusions tardives* des scores de classification supervisée obtenus avec chaque descripteur. Nous comparons plusieurs approches de fusion tardive dans le Chapitre 5.

7.2 Etat de l'art

Quand un utilisateur cherche certains éléments dans une collection multimédia, il formule une requête textuelle, par exemple "discours de Barack Obama après les élections", qui peut être convertie dans des mots-clés à rechercher dans la collection : "Obama", "président" (si le système sait aussi que Obama est le président), "discours", "élection(s)". La collection doit être annotée avec des tels mots-clés si on veut qu'une requête de ce type donne un résultat.

7.2.1 La base vidéo TRECVid

Nos travaux sont centrés sur la base TRECVid, car elle correspond bien à notre objectif de concevoir un système d'indexation très générique. Cette base vidéo comporte un ensemble très riche de concepts de haut niveau sémantique à détecter (et annoter) dans les vidéos. Les 346 concepts proposés peuvent être des objets (Bus, Arbre, Voiture, Téléphone, Chaise), des actions (Chanter, Manger, Poignée de main), des situations ou des types de scènes (Paysage au bord de l'eau, Scène d'intérieur, Cuisine, Chantier), concepts abstraits (Science/Technologie), des types de personnes (Chef d'entreprise, Femme, Personne asiatique, Membre de gouvernement) ou même des personnes spécifiques (Hu Jintao, Donald Rumsfeld). La collection TRECVid est organisée en *"shots"* vidéo d'une longueur de l'ordre de quelques secondes ou quelques dizaines de secondes maximum, et ces shots doivent être annotés avec la présence ou l'absence des 346 concepts. Les shots sont très divers aussi, pouvant être acquis par n'importe qui, avec n'importe quel dispositif, dans n'importe quel contexte.

La base vidéo associée à la campagne TRECVid est divisée en deux parties, une partie de développement et une partie de test. Pour la partie de développement, l'annotation est fournie, dans le but de pouvoir entraîner des classifieurs supervisés (qui pourront déduire la relation entre les descripteurs et la présence ou l'absence d'un concept sémantique). Après avoir extrait des descripteurs et entrainé des classifieurs supervisés, les participants au concours appliquent leurs algorithmes sur la partie test, et ils déduisent, pour chaque shot et pour chacun des 346 concepts, un score de classification qui indique la "probabilité" pour un shot, de contenir un concept. Sur la base de test, pour chaque concept, les participants construisent ensuite une liste de maximum 2000 shots, en ordre décroissant de leur "probabilité" de contenir le concept (de la même façon qu'une liste de pages web fournie par un moteur de recherche Internet). Finalement, la qualité de ces listes pour chaque concept (il faut que les shots qui contiennent vraiment le concept soient concentrés vers le début de la liste) est évaluée par les organisateurs du challenge, en utilisant la *précision moyenne par inférence* [Yilmaz 2006, Yilmaz 2008].

7.2.2 La chaîne de traitement

Nous présentons ensuite la chaîne de traitement que nous utilisons au sein du consortium IRIM. Même si la chaîne est appliquée sur la base TRECVid, elle peut être étendue avec des adaptations minimales à n'importe quelle autre collection vidéo. Les étapes sont les suivantes, présentées aussi dans la Figure 7.2 :

1. *Extraction de descripteurs* à partir des shots vidéo ; les descripteurs caractérisent différents aspects (et modalités) de la vidéo, par exemple les couleurs dominantes, les orientations spatiales dominantes, les principales directions de mouvement, les sons, le texte superposé etc.

2. Facultatif : une *optimisation des descripteurs*, pour améliorer les résultats de classification supervisée. L'optimisation consiste en une transformation suivant une loi de puissance suivie par une Analyse en Composantes Principales (ACP) [Safadi 2013].

3. *Classification supervisée* : les exemples annotés de la base de développement sont utilisés pour entraîner des classifieurs supervisés (KNN - K plus proches voisins ; MSVM - une approche multi-apprentissage basée sur des machines à vecteurs-support (SVM) [Safadi 2010]). Les classifieurs sont ensuite utilisés pour obtenir des scores de classification sur la base de test. Un classifieur est entrainé et appliqué pour chaque descripteur et pour chaque concept. La combinaison d'un descripteur et d'un classifieur supervisé est appelée "*expert*".

4. *Fusion des résultats KNN-MSVM* : pour un même descripteur et un même concept, une moyenne pondérée est faite entre les scores en sortie du classifieur KNN et ceux en sortie du classifieur MSVM. On obtient ce qu'on appelle des experts "FUSEB".

5. *Fusion tardive* : Comme les experts basés sur des descripteurs de types différents encodent des informations différentes, les experts associés sont *complémentaires*. On fait alors une fusion tardive pour combiner l'information venant de ces différentes sources, pour améliorer les résultats d'indexation.

6. Facultatif : Les shots de la base TRECVid sont obtenus en découpant des vidéos plus longues selon les changements de plan. Il y a alors une corrélation entre les contenus présents dans des shots consécutifs d'une même vidéo. On ajoute une étape de *re-scoring temporel* [Safadi 2011] pour prendre en compte ce contexte temporel des concepts dans les shots, afin d'améliorer l'indexation.

7. Facultatif : jusqu'à maintenant, nous avons traité les concepts indépendamment, sans prendre en compte les éventuelles relations entre concepts. Une étape additionnelle de *feedback conceptuel* [Hamadi 2013] sert à prendre en compte le contexte sémantique et améliore encore les résultats.

Nos contributions sont au niveau des étapes de description du contenu vidéo et au niveau de la fusion tardive. Nous allons en faire une très courte description de l'état de l'art.

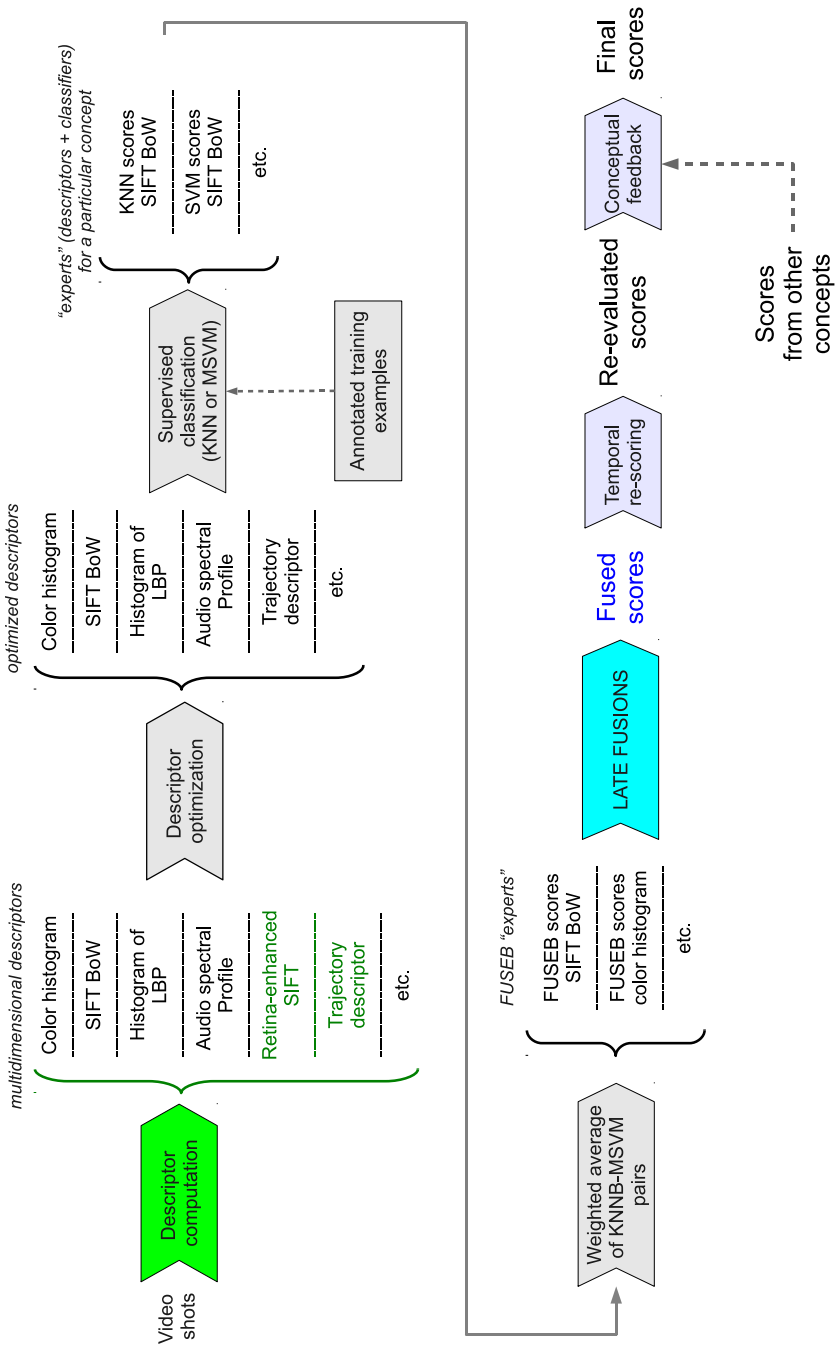


FIGURE 7.2 – La chaine de traitement pour l'indexation sémantique, que nous utilisons pour la collection TRECVid, au sein du consortium IRIM. Voir texte pour détails.

7.2.3 Descripteurs pour le contenu vidéo

Les descripteurs ont le rôle de fournir des descriptions compactes des vidéos, avec lesquelles les classifieurs supervisés vont travailler. Idéalement, les descripteurs encodent l'information utile des vidéos et sont robustes aux variations ou perturbations qui n'apportent pas d'information (comme par exemple une caméra non stabilisée, variations d'éclairage, angle de prise de vue etc.).

Les descripteurs peuvent représenter différentes informations, par exemple les couleurs, les textures, les formes et contours, le mouvement ou l'audio. Certains concepts sémantiques sont mieux détectés par certains descripteurs (la couleur verte peut indiquer la présence de la végétation) et d'autres concepts sont mieux détectés avec d'autres (le concept "Dancer" est mieux détecté avec un descripteur de mouvement). Les concepts de la base TRECVid étant très nombreux et hétérogènes, on aura besoin d'une batterie de descripteurs complémentaires pour capturer cette grande diversité d'informations.

Parmi les types de descripteurs, nous pouvons mentionner :

- descripteurs de couleur composés d'histogrammes de couleurs [Gosselin 2008] ;
- descripteurs de texture : histogrammes de "local binary patterns" [Ojala 1996, Delezoide 2011, Zhu 2011], bancs de filtres Gabor [Turner 1986], wavelets quaternioniques [Gosselin 2008] ;
- descripteurs audio : le spectre audio à court terme sous la forme de coefficients MFCC [Ballas 2012b] ;
- descripteurs spatiaux et spatio-temporels basés sur des caractéristiques locales (utilisant souvent le modèle Sac-de-Mots (Bag of Words, BoW) [Csurka 2004])

Nos travaux étant focalisés sur la dernière catégorie, nous la détaillons dans la suite.

7.2.3.1 Généralités concernant le modèle Sac-de-Mots

Si on veut caractériser une image, au lieu d'essayer de caractériser l'image dans son entier, nous pouvons sélectionner de petites sous-parties (des caractéristiques locales) et caractériser celles-ci, puis ensuite d'agréger ces caractéristiques. Dans le modèle Sac-de-Mots (Bag of Words, BoW), dont le principe est illustré dans la Figure 7.3, l'image est représentée comme une collection non-ordonnée de caractéristiques locales.

En pratique, pour créer un descripteur BoW, les étapes sont les suivantes (illustrées en Figure 7.3) :

1. choisir un jeu de caractéristiques locales, habituellement positionnées avec un pas régulier dans l'image (une grille dense), ou en utilisant un détecteur de points d'intérêt ;
2. décrire chaque caractéristique locale par un descripteur local (un descripteur de petits fragments d'image) ;

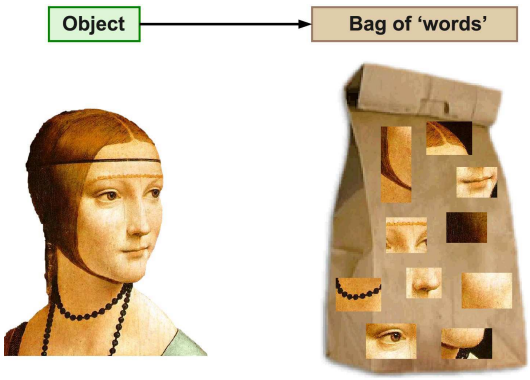


FIGURE 7.3 – Principe de base du modèle Sac-de-Mots : une image est représentée comme une collection non-ordonnée de petites sous-parties (deux yeux, un nez, une bouche etc.). La position de ces éléments n'est pas prise en compte dans ce modèle. Image provenant de : *Li Fei-Fei, Rob Fergus, and Antonio Torralba. Recognizing and learning object categories - short course. 2009*

3. sur une collection d'images (ou de vidéos) d'entrainement, extraire et décrire un grand nombre de caractéristiques locales, et déduire un *vocabulaire* par classification (clustering, par exemple *k-means* [Arthur 2007]) ;

4. pour une image/vidéo à représenter, on sélectionne un jeu de caractéristiques locales, on calcule leurs descripteurs locaux, on approxime le descripteur au mot de vocabulaire le plus proche, et on fait un *histogramme de mots du vocabulaire* qui apparaissent dans l'image/vidéo. Cet histogramme constitue le descripteur Sac-de-Mots de l'image ou vidéo.

L'avantage du modèle BoW par rapport à une représentation globale de l'image est la robustesse aux occlusions partielles ; de plus, parce que les positions relatives des éléments sont ignorées, le modèle est robuste aux changements de point de vue ou aux déformations.

7.2.3.2 Caractéristiques locales pour BoW

Les approches qui décrivent les caractéristiques locales spatiales par des histogrammes des orientations des gradients d'intensité (SIFT [Lowe 2004b], SURF [Bay 2008]) ou par des comparaisons entre les intensités des pixels d'un petit voisinage (BRIEF [Calonder 2010], ORB [Rublee 2011], BRISK [Leutenegger 2011], FREAK [Ortiz 2012]) donnent de bons résultats pour la classification d'images et la reconnaissance d'objets [Csurka 2004].

Malgré leurs bonnes performances sur des images statiques, les caractéristiques locales mentionnées ci-dessus n'exploitent pas l'information additionnelle contenue dans des vidéos : les changements se faisant dans le temps, le plus important étant le mouvement. Si l'on veut détecter dans une vidéo un concept sémantique qui n'a pas de relation avec le mouvement, les caractéristiques statiques marchent relativement bien. Elles sont par contre peu efficaces en termes de reconnaissance d'actions.

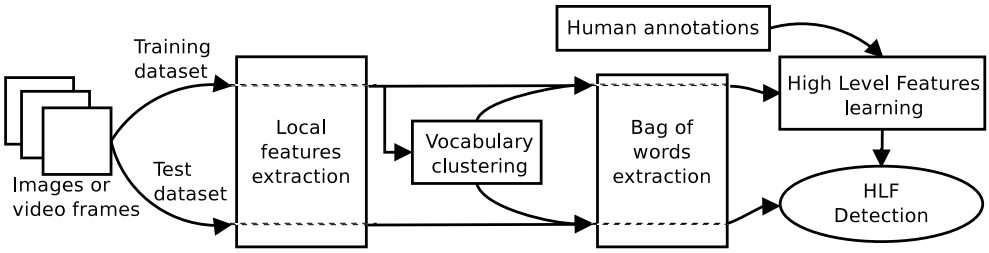


FIGURE 7.4 – Chaîne de traitement pour le modèle Sac-de-Mots (BoW)

Pour cela, le modèle BoW a été étendu à d'autres classes de caractéristiques locales, dans le but d'inclure non seulement de l'information spatiale, mais *spatio-temporelle*. Parmi les caractéristiques spatio-temporelles, nous pouvons évoquer :

- les *points d'intérêt spatio-temporels* [Laptev 2003, Ke 2005, Dollár 2005, Niebles 2008] : des points "intéressants" non seulement d'un point de vue spatial (comme un coin proéminent) mais aussi temporel (accélération, changement de direction du mouvement) ; le voisinage spatio-temporel de chaque point peut être ensuite décrit par des dérivées spatiales et/ou temporelles, par des histogrammes d'orientation du gradient d'intensité ou par des orientations du flot optique [Laptev 2007] ;

- *MoSIFT* [Chen 2009] est une extension du descripteur de caractéristique locale spatiale SIFT (SIFT est basé sur des histogrammes des orientations du gradient d'intensité). MoSIFT concatène au vecteur SIFT un vecteur obtenu à partir des orientations du flot optique, pour avoir une description non seulement de l'aspect local, mais aussi du mouvement local ;

- *trajectoires de points suivis* : des points peuvent être suivis le long des trames vidéo, pour construire des trajectoires. Ces trajectoires capturent une information de mouvement très riche, qui peut servir à la reconnaissance d'actions [Wang 2011, Ballas 2011].

7.2.4 Stratégies de fusion tardive

Après avoir extrait des descripteurs pour les shots vidéo, des classifieurs supervisés sont appliqués pour obtenir des scores d'appartenance à un concept pour chaque shot vidéo. Jusqu'à maintenant, les différents descripteurs n'ont pas été exploités conjointement, et pour une collection vidéo difficile (comme TRECVid), cela ne suffit pas pour obtenir des bons résultats. Il faut alors fusionner les informations venant de descripteurs différents, pour profiter de leur complémentarité afin d'améliorer les résultats.

Nos travaux se focalisent sur les *fusions tardives*, qui fusionnent les scores de classification supervisée donnés par les "experts" de la Figure 7.2. D'autres types de fusions existent, notamment les fusions précoces, qui combinent les descripteurs avant la classification supervisée. Cependant, comme l'entrainement d'un classifieur supervisé sur un descripteur de grande dimension est difficile, et la pondération entre chaque partie de ce grand descripteur n'est pas triviale non plus, nous préférons les fusions tardives.

Les fusions tardives peuvent être aussi simples qu'une moyenne arithmétique (pondérée ou non) des scores de classification venant de différents experts, ou peuvent prendre en compte les interdépendances entre les experts, comme dans l'intégrale de Choquet [Cliville 2004]. Un algorithme de fusion tardive par moyenne pondérée qui donne souvent des bons résultats est AdaBoost [Freund 1997, Schapire 1999], qui combine les experts d'un telle façon que la complémentarité est bien exploitée.

7.2.5 Améliorations proposées

Partant de cette analyse de l'état de l'art (voir Chapitre 2 pour plus de détails), nous avons identifié les besoins suivants pour la problématique de l'indexation multimédia :

- un besoin de *descripteurs spatio-temporels* pour le contenu vidéo, qui profitent de l'information temporelle additionnelle et améliorent les résultats d'indexation, sans trop augmenter la complexité de calcul ; ces descripteurs seront *génériques*, capables de détecter non seulement des concepts statiques, mais aussi dynamiques, avec de bonnes performances ;
- des stratégies de *fusion d'information* adaptées à notre contexte de travail, capables de gérer une grande diversité d'experts et d'exploiter leur complémentarité afin d'améliorer les résultats ;

La façon dont nous adressons ces deux besoins constitue la contribution de cette thèse :

- Un ensemble de descripteurs spatiaux et spatio-temporels basés sur des caractéristiques locales SIFT/SURF dans un modèle Sac-de-Mots (BoW) est proposé dans le Chapitre 3. Ces descripteurs, basés sur le *pré-traitement des vidéos avec le modèle de rétine humaine* de [Benoit 2010], ont des taux de reconnaissance de concepts améliorés et ils sont plus génériques, capables de fonctionner avec non seulement des concepts spatiaux, mais aussi spatio-temporels.
- Dans TRECVid, il y a très peu de descripteurs dédiés au mouvement, à cause de la grande complexité de calcul nécessaire pour analyser des vidéos par rapport aux images statiques. Nous proposons dans le Chapitre 4 une batterie de descripteurs de mouvement qui sont des *Sacs-de-Mots de trajectoires* de points suivis, qui donnent une description fortement temporelle du contenu vidéo.
- Nous avons à notre disposition non seulement les descripteurs SIFT/SURF BoW utilisant un modèle de rétine et les descripteurs BoW de trajectoires, mais aussi d'autres descripteurs très divers fournis par le groupe IRIM. Nous explorons dans le Chapitre 5 des stratégies de *fusion tardive automatique* pour exploiter cette diversité d'experts.

7.3 Pré-traitement rétinien pour descripteurs SIFT/SURF BoW

Comme nous avons vu précédemment, les descripteurs Sac-de-Mots (BoW) SIFT ou SURF donnent de bons résultats dans la détection ou la reconnaissance d'objets ou de scènes, et

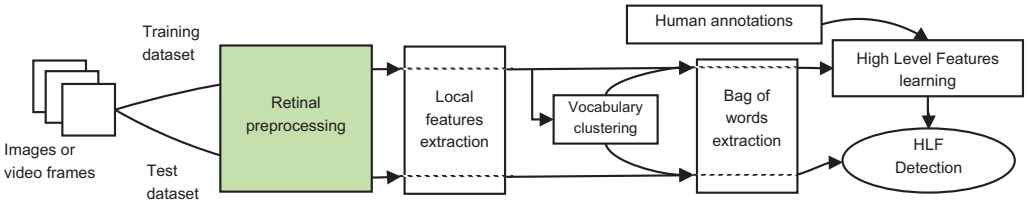


FIGURE 7.5 – Chaîne de traitement modifiée pour l'extraction de descripteurs SIFT/SURF BoW : un prétraitement rétinien des vidéos est ajouté avant l'étape d'extraction de caractéristiques locales SIFT ou SURF.

ils sont classiquement les meilleurs descripteurs dans TRECVid [Over 2012]. Cependant, malgré ces bonnes performances, ces descripteurs peuvent être perturbés par les dégradations d'image (bruit, artéfacts de compression). De plus, ils ne peuvent décrire l'information spatio-*temporelle*, ce qui les rend moins appropriés pour la reconnaissance de concepts liés au mouvement.

Nous proposons de rendre les descripteurs SIFT/SURF BoW plus robustes aux dégradations d'image/vidéo et également de les rendre sensibles au contenu spatio-*temporel*. Pour cela, nous utilisons le modèle de rétine humaine de [Benoit 2010] pour prétraiter les vidéos avant d'extraire les Sacs-de-Mots, comme illustré dans la Figure 7.5.

Dans la suite, nous décrivons le comportement du modèle de rétine utilisé, puis comment nous utilisons ce comportement pour construire des descripteurs SIFT/SURF BoW améliorés.

7.3.1 Le modèle rétinien

Le modèle rétinien de [Benoit 2010] que nous utilisons traite les vidéos d'entrée et génère deux canaux de sortie, appelés le canal (ou la voie) *parvocellulaire* et le canal (voie) *magnocellulaire*.

Le canal *parvocellulaire* traite les détails spatiaux et les couleurs. Il normalise les couleurs, augmente le contraste local, répond bien aux signaux temporellement constants et il lisse les variations temporelles rapides. Une propriété intéressante est que le canal parvocellulaire transmet d'abord une information spatiale à faible résolution, pour ensuite transmettre l'information de plus haute résolution (effet "coarse to fine") : quand la rétine est initialisée (quand on "ouvre les yeux") ou quand un événement spatio-temporel se produit (apparition ou mouvement d'un objet), la rétine transmet uniquement les basses fréquences spatiales (une image lisse, mais avec un bon rapport signal/bruit) ; plus tard, quand la réponse du canal se stabilise, des fréquences spatiales plus hautes sont transmises pour une analyse plus détaillée du contenu.

D'un point de vue d'implémentation, le canal parvocellulaire consiste en une séquence d'images avec des détails spatiaux mieux mis en évidence, des couleurs corrigées, des détails plus visibles dans les zones sombres, un bruit réduit et des artéfacts de compression vidéo réduits également. Un exemple de l'effet du canal parvocellulaire sur une vidéo est donné dans la Figure 7.6. L'effet "coarse to fine" peut être observé sur le visage de la

personne et sur les nuages de l'arrière-plan, pour lesquels les détails sont améliorés et la luminance globale atténuée après un certain temps (Figure 7.6b). Les nuages sont à peine visibles dans les Figures 7.6a et 7.6b, mais ils sont plus visibles dans la Figure 7.6d.

Concernant les limitations du modèle, le canal parvocellulaire ne peut pas totalement éliminer toutes les perturbations : il diminue le bruit introduit par les capteurs électroniques, mais pour les effets de compression, si le taux compression est trop grand, les effets de blocs ne peuvent pas être complètement éliminés, ils sont justes lissés.

L'autre sortie du modèle rétinien est le canal *magnocellulaire*. Il ne distingue pas les couleurs, mais il est sensible aux événements spatio-temporels, répond fortement aux signaux transitoires (mouvement, apparition ou disparition d'un objet etc.) et faiblement aux signaux ayant des variations lentes. Concernant l'évolution de la réponse, magnocellulaire, nous pouvons aussi parler d'une phase transitoire et d'une phase stable pour ce canal. Juste après l'initialisation de la rétine, les fréquences spatiales basses sont transmises un court moment, donnant une réponse forte sur les grandes frontières spatiales jusqu'à la fin de la phase transitoire (Figure 7.6e). Ceci permet d'utiliser le canal magnocellulaire comme un détecteur de zones spatiales potentiellement intéressantes. Quand la rétine entre dans la phase stable, ce canal répond uniquement aux zones en mouvement (Figure 7.6f), ce qui fait que dans cette phase, le canal magnocellulaire peut être utilisé comme un détecteur de zones temporellement transitoires (ce qui correspond d'habitude au mouvement).

Concernant les limitations du modèle, les effets de blocs sévères peuvent provoquer de fausses alarmes sur le canal magnocellulaire, mais les détecteurs de points d'intérêt classiques ont aussi ce problème. Cependant, le filtrage spatio-temporel de la rétine diminue les perturbations, ce qui conduit tout de même à un nombre plus faible de fausses détections.

Segmentation de blobs d'intérêt : En se basant sur la sortie du canal magnocellulaire, nous utilisons un algorithme qui nous permet de sélectionner des zones d'intérêt (blobs) à partir des vidéos. L'algorithme, plus complexe qu'un simple seuillage du canal magnocellulaire, est décrit en détail dans la Section 3.1.3.2. Il permet de sélectionner des blobs stables dans le temps et avec moins de sélections accidentelles dues au bruit résiduel. En plus, cet algorithme sélectionne des zones non seulement intéressantes par rapport à un seuil fixe prédéfini, mais intéressantes par rapport à leur voisinage et à leur contexte (voisinage encore plus grand). Des exemples de blobs segmentés sont donnés dans les Figures 7.6g et 7.6h : 5 trames après le début du traitement, les contours de grande taille sont sélectionnés, ce qui correspond à la "saillance" (ce n'est pas une saillance d'aussi haut niveau que celle décrite dans l'état de l'art, mais un modèle plus basique et moins couteux) spatiale ; 40 trames après le début, uniquement les blobs qui correspondent aux zones en mouvement sont sélectionnés.

7.3.2 Descripteurs SIFT/SURF BoW améliorés proposés

Comme mentionné précédemment, les Sacs-de-Mots utilisant des caractéristiques locales purement spatiales (comme SIFT ou SURF) ne sont pas bien adaptés à la reconnaissance de concepts liés au mouvement. En plus, ils sont sensibles aux dégradations d'image comme

(a) Trame d'entrée, 5 trames après le début

(b) Trame d'entrée, 40 trames après le début

(c) Sortie parvocellulaire, 5 trames après le début

(d) Sortie parvocellulaire, 40 trames après le début

(e) Sortie magnocellulaire, 5 trames après le début

(f) Sortie magnocellulaire, 40 trames après le début

(g) Blobs d'intérêt segmentés, 5 trames après le début

(h) Blobs d'intérêt segmentés, 40 trames après le début

FIGURE 7.6 – Exemple de pré-traitement rétinien, 5 et 40 trames après l'initialisation de la rétine. Après 5 trames, la rétine est encore dans sa phase transitoire : la voie parvocellulaire transmet beaucoup de luminance et les détails ne sont pas encore améliorés ; en même temps, la voie magnocellulaire répond aux grandes structures spatiales. Après 40 trames, la rétine est dans la phase stable : la voie parvocellulaire transmet moins de luminance et augmente les détails spatiaux ; en même temps, la voie magnocellulaire répond généralement aux zones en mouvement (le visage de la personne qui parle). Les blobs d'intérêt segmentés sont obtenus en traitant la sortie de la voie magnocellulaire : après 5 trames, on sélectionne des zones potentiellement intéressantes d'un point de vue spatial, et après 40 trames, on sélectionne plutôt les zones en mouvement.

le bruit ou les artefacts de compression. Pour améliorer ces aspects, nous proposons de prétraiter les vidéos avec le modèle de rétine avant d'extraire des caractéristiques locales, comme vu dans la Figure 7.5 et nous construisons les descripteurs suivants.

7.3.2.1 Descripteurs d'image-clé

Une première classe est celle des descripteurs Sac-de-Mots avec des caractéristiques locales collectées à partir d'une seule image clé du shot vidéo. Nous décrivons ces images avec des caractéristiques locales de type OpponentSIFT :

- *SIFT* : caractéristiques OpponentSIFT collectées sur une grille dense appliquée sur l'image clé (Figure 7.7a) ; c'est le descripteur de référence (sans prétraitement rétinien) ;

- *SIFT retina* : au lieu de collecter les caractéristiques OpponentSIFT sur l'image originale, on les extrait sur la sortie parvocellulaire au moment de l'image clé (Figure 7.7b) ; l'avantage est de bénéficier des propriétés de réduction des perturbations sur la voie parvocellulaire, et de l'augmentation du contraste local ;

- *SIFT multichannel* : le canal magnocellulaire encode des informations liés au mouvement, donc une signature SIFT collectée sur ce canal donne de l'information sur le mouvement local ; nous concaténons la signature locale OpponentSIFT du canal parvocellulaire avec la signature locale SIFT, au même endroit, du canal magnocellulaire, pour construire des caractéristiques locales spatio-temporelles (Figure 7.7c) ;

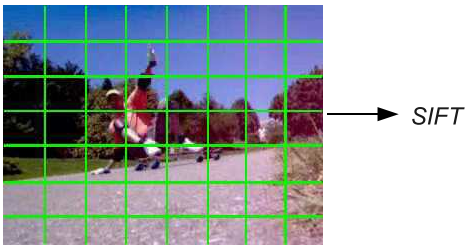
7.3.2.2 Descripteurs à fenêtres temporelles et masquage de blobs

Par rapport à la classe précédente, nous ne prenons plus une seule image, mais une série d'images (entre 20 et 40, en fonction du paramétrage) autour de l'image clé, à partir desquelles nous collectons des caractéristiques locales. En plus, nous utilisons l'algorithme de segmentation de blobs présenté précédemment pour ne collecter que les caractéristiques qui sont potentiellement plus intéressantes. Nous obtenons les descripteurs suivants :

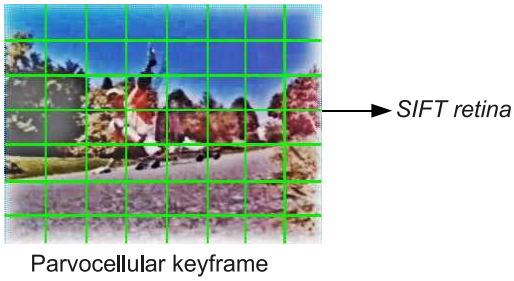
- *SIFT simple masking* : caractéristiques OpponentSIFT collectées sur les images originales, mais dans une fenêtre temporelle et avec une sélection de zones intéressantes (Figure 7.8a) ;

- *SIFT retina masking* : similaire au précédent, mais les caractéristiques locales sont collectées sur la voie parvocellulaire (Figure 7.8b) ;

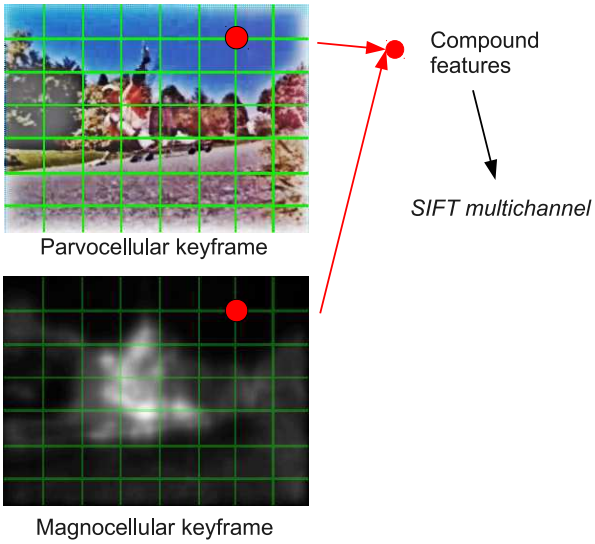
- *SIFT multichannel masking* : utilise des caractéristiques composées spatio-temporelles (parvo-magno) (Figure 7.8c) ;



(a) SIFT

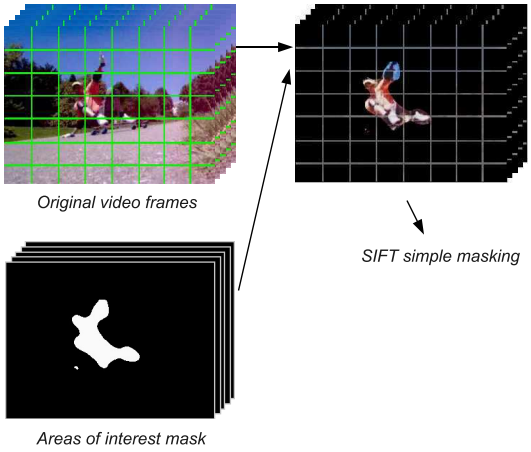


(b) SIFT retina

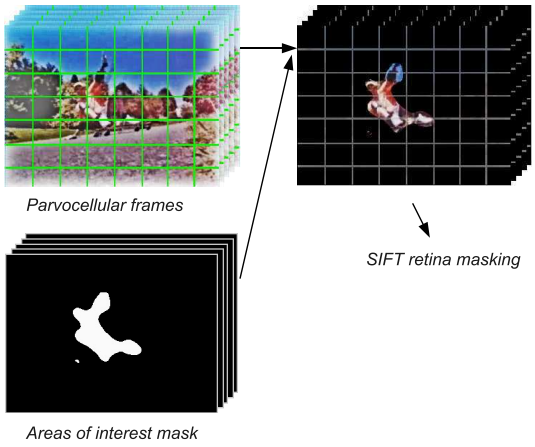


(c) SIFT multichannel

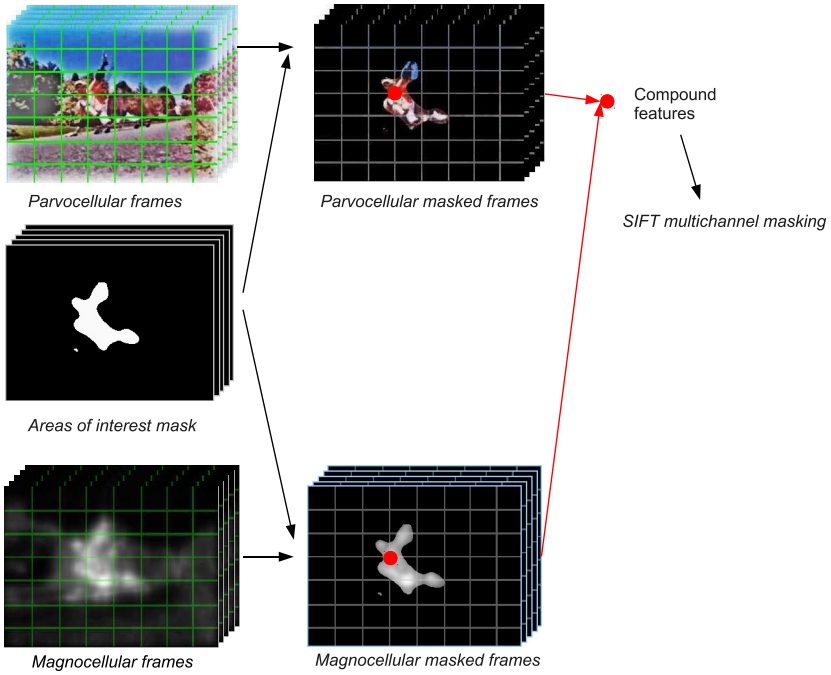
FIGURE 7.7 – Descripteurs d'image-clé utilisant le pré-traitement rétinien



(a) SIFT simple masking



(b) SIFT retina masking



(c) SIFT multichannel masking

FIGURE 7.8 – Descripteurs à fenêtres temporelles avec masquage de blobs

TABLE 7.1 – Précision moyenne par inférence (infAP) des descripteurs Sac de Mots utilisant la rétine, sur tous les 346 concepts de la base. On observe le gain apporté par rapport à la référence *SIFT* qui n'utilise pas de prétraitement rétinien.

Descripteur	infAP	gain
SIFT	0.0830	baseline
SIFT retina	**0.0904**	9%
SIFT multichannel	0.0878	5.7%
SIFT retina masking	0.0843	2%
SIFT multichannel masking	0.0857	3.2%

7.3.3 Validation sur la base TRECVid 2012

Nous testons nos améliorations proposées sur la partie de développement de la base TRECVid SIN 2012, avec des descripteurs Sac-de-Mots basés sur OpponentSIFT (voir Chapitre 3 pour les résultats avec OpponentSURF sur TRECVid 2010 et 2011). La mesure de performance utilisée est la précision moyenne par inférence (infAP) [Yilmaz 2006, Yilmaz 2008].

La Table 7.1 montre les résultats globaux (infAP moyennes sur tous les concepts de la base). On peut voir que tous les descripteurs utilisant notre prétraitement ont des performances globales supérieures à celles de l'approche basique.

Concernant les résultats concept par concept, même si *SIFT retina* est globalement le meilleur, il n'est pas le meilleur pour tous les concepts. Par exemple, *SIFT* est le meilleur pour 48 concepts, *SIFT retina* pour 50 concepts, *SIFT retina masking* pour 15 et *SIFT multichannel masking* pour 41. Ceci justifie l'extraction de l'ensemble du jeu de descripteurs et l'exploitation de leur *complémentarité* à travers une fusion. Par exemple, une fusion par moyenne arithmétique des scores de classification (chaque "expert") augmente l'infAP jusqu'à 0.1220.

7.3.4 Conclusions

Les descripteurs proposés, basés sur des Sacs-de-Mots de caractéristiques visuelles SIFT ou SURF, peuvent profiter du prétraitement rétinien pour améliorer leurs performances et pour obtenir une meilleure sensibilité au contenu spatio-*temporel* (surtout avec *SIFT multichannel masking*).

Le chapitre 4 enrichit la description spatio-temporelle des shots vidéo en ajoutant des descripteurs Sac-de-Mots de trajectoires, ce qui augmente encore plus la sensibilité aux événements (concepts) spatio-temporels.

7.4 Descripteurs Sac-de-Mots de trajectoires

Les trajectoires sont des indices importants pour identifier les types de mouvements, donc les types d'actions présentes dans les vidéos. Par rapport à une caractéristique spatio-temporelle locale vue dans la section précédente (*SIFT multichannel*), une trajectoire donne une information moins locale sur le mouvement. Une trajectoire peut suivre un objet ou un

point d'intérêt pendant une durée plus grande (allant jusqu'à quelques secondes), par rapport à une caractéristique *SIFT multichannel* qui décrit le mouvement local autour d'un point dans une seule trame vidéo.

7.4.1 Principe

Nous proposons donc d'extraire des trajectoires en faisant du suivi de points d'intérêt le long des trames vidéo. Nous considérons chaque trajectoire d'un shot vidéo comme une caractéristique locale, et nous regroupons ces trajectoires dans un modèle Sac de Mots. La chaîne de traitement pour extraire les trajectoires est la suivante :

1. *choisir un jeu de points d'intérêt* pour les suivre au long du shot vidéo ; nous utilisons le détecteur de points d'intérêt Good Features to Track (GFTT) [Shi 1994b] pour détecter de temps en temps de nouveaux points pour initialiser de nouvelles trajectoires ;

2. *suivre chaque point d'intérêt* au long du shot vidéo en utilisant le flot optique [Bouguet 2000], jusqu'à ce que la trajectoire associée devienne trop longue et entraine une erreur de suivi ou jusqu'à ce que le point quitte l'image ;

3. pendant le suivi, *estimer le mouvement de la caméra* en chaque point des trames vidéo, pour pouvoir le prendre en compte plus tard quand les trajectoires seront décrites ;

4. *ajouter de nouveaux points à suivre* de temps en temps ou lorsque le nombre courant de points actifs devient trop petit ;

5. à la fin du shot vidéo, *sélectionner* uniquement les trajectoires qui ont du mouvement et les *post-traiter* pour éliminer les zones statiques au début et à la fin ;

6. pour chaque trajectoire, *calculer des descripteurs de trajectoire*, par exemple un histogramme d'orientations du mouvement le long de la trajectoire (d'un point de vue du modèle BoW, soit l'équivalent de calculer la signature SIFT d'une caractéristique locale spatiale) ; plusieurs descripteurs de trajectoire sont proposés (voir Section 7.4.2, chacun donnant lieu à un modèle BoW distinct ;

7. pour chaque façon de décrire une trajectoire (voir point précédent), générer un modèle BoW pour lequel les trajectoires sont les "caractéristiques locales", et décrire le shot vidéo par un histogramme de "mots-trajectoires".

7.4.2 Descripteurs de trajectoire

Pour caractériser une trajectoire, nous calculons les descripteurs suivants (détails dans la Section 4.1.6) :

- le *descripteur spatial BRIEF* [Calonder 2010] du premier point de la trajectoire ;
- deux *histogrammes des directions de mouvement* le long de la trajectoire ;

- deux *histogrammes des directions d'accélération* le long de la trajectoire ;
- *vecteurs normalisés de déplacement* le long de la trajectoire : les vecteurs de déplacement d'une image à l'autre sont re-échantillonnés en temps pour avoir des durées et déplacements totaux constants ;
- *vecteurs normalisés d'accélérations* le long de la trajectoire (idée similaire) ;

pour chaque approche, les informations fournies par ces descripteurs sont finalement présentées selon une représentation en Sac de Mots spécifique.

7.4.3 Validation sur la base KTH

Nous avons fait une première série d'expérimentations sur la base KTH qui est spécialisée sur la reconnaissance d'actions. Cette base comporte 6 actions (boxer, applaudir, agiter les mains, trotter, courir, marcher) réalisées par différentes personnes dans des contextes simples. Les résultats sont donnés dans la Table 7.2.

Les meilleurs résultats sont obtenus pour les vecteurs normalisés de déplacement et ceux d'accélération, ainsi que pour l'histogramme de directions de mouvement avec un bin zéro. Même si ces résultats ne sont pas au même niveau que les approches les plus performances de l'état de l'art sur cette base (qui peuvent dépasser 95% de précision), nos rappelons que nous n'avons fait aucune optimisation, car notre but final est l'indexation sémantique sur une base beaucoup plus générique, comme TRECVid. Pour une approche non-optimisée, nos résultats sont bons et ils valident l'approche, et nous passons maintenant aux expérimentations sur la base TRECVid.

7.4.4 Expérimentations sur la base TRECVid SIN 2012

Pour cette série de tests, nous avons fait certaines optimisations de paramètres pour notre méthode (détails dans le Chapitre 4) et nous avons introduit une extension du modèle Sac de Mots.

Dans le modèle Sac de Mots classique, toutes les trajectoires ont le même poids lorsque l'on génère l'histogramme des mots visuels (mots-trajectoires dans notre cas). Le problème sur la base TRECVid est qu'elle est très générique et que les shots vidéo peuvent être assez longs par rapport à la durée d'une action isolée. La conséquence est que beaucoup de trajectoires ne sont pas pertinentes pour une action qui ne dure qu'une petite fraction de la durée du shot. Nous proposons donc un *modèle de Sac de Mots différentiel*, qui redonne du poids aux types de mouvements qui n'apparaissent qu'à certains moments isolés du shot vidéo, dans le but de renforcer leur importance, même dans les shots longs. Cette approche est détaillée dans la Section 4.3.2.

La Table 7.3 montre les résultats globaux (moyenne sur tous les concepts) évalués sur la base TRECVid 2012y, avec entrainement sur 2012x. Même si les valeurs semblent basses, elles sont bien au-dessus des performances obtenues avec un tirage aléatoire. Cela est encourageant surtout en sachant que beaucoup de concepts de la base TRECVid n'ont pas nécessairement un lien direct avec le mouvement. Malgré le nombre réduit de concepts pouvant être associés au mouvement, 129 concepts (sur un total de 346) ont eu des résultats

TABLE 7.2 – Résultats de reconnaissance d'actions sur la base KTH avec l'approche Sac de Mots de trajectoires, en utilisant différents descripteurs de trajectoire. La taille de vocabulaire est le nombre de "mots-trajectoires" utilisés dans le modèle Sac de Mots. La précision de classification est donnée pour les descripteurs sans prendre en en compte le mouvement de la caméra, ainsi qu'avec une compensation du mouvement de la caméra.

descripteur de trajectoire	taille vocab.	P (%)	P (%) comp. mouv. caméra
hist. dir. mouv.	32	67.13	62.04
	64	71.30	67.59
	128	69.44	70.83
hist. dir. mouv. avec bin 0	32	**79.17**	70.37
	64	**77.78**	73.15
	128	**79.63**	**75.93**
hist. dir. accel.	32	62.96	51.39
	64	61.57	51.85
	128	63.43	54.63
hist. dir. accel. avec bin 0	32	59.26	53.24
	64	62.04	55.56
	128	61.57	54.17
vect. déplacement 8 échantillons	96	**75.46**	**75.00**
	192	**81.94**	**76.39**
	384	**81.84**	75.00
vect. déplacement 16 échantillons	96	68.98	72.69
	192	**76.39**	**75.46**
	384	**80.09**	**76.39**
vect. accél 7 échantillons	96	75.93	62.96
	192	70.83	67.13
	384	68.52	61.57
vect. accél 15 échantillons	96	59.72	57.87
	192	66.67	60.65
	384	64.81	57.87

TABLE 7.3 – Résultats globaux (infAP moyennés sur tous les 346 concepts) sur TRECVid 2012y (entraînement sur 2012x), pour les différentes descriptions possibles d'une trajectoire, ainsi que quelques descriptions concaténées. "c.c." signifie l'emploi de la compensation du mouvement de la caméra, "diff" signifie des Sacs de Mots différentiels. Les résultats en gras mettent en évidence les améliorations significatives grâce à la compensation du mouvement de la caméra et/ou aux sacs de mots différentiels.

descripteur de trajectoire	vocab. K	AP	AP c.c.	AP diff.	AP c.c. diff.
BRIEF du début traject.	256	**0.0588**	-	0.0489	-
	512	0.0564	-	0.0473	-
hist. dir. mouv.	64	0.0367	0.0371	0.0360	0.0364
(1)	128	0.0385	0.0384	0.0377	0.0378
	256	0.0391	0.0386	0.0385	0.0373
hist. dir. mouv. avec bin 0	64	0.0346	0.0281	0.0341	**0.0321**
(2)	128	0.0366	0.0285	0.0368	**0.0338**
	256	0.0367	0.0282	0.0379	**0.0340**
hist. dir.accel.	64	0.0396	0.0358	0.0378	0.0351
(3)	128	0.0403	0.0375	0.0391	0.0371
	256	0.0408	0.0386	0.0392	0.0377
hist. dir. accel. avec bin 0	64	0.0281	0.0242	**0.0303**	**0.0283**
(4)	128	0.0304	0.0247	**0.0328**	**0.0300**
	256	0.0311	0.0254	**0.0336**	**0.0311**
vect. dépl. 8 échantillons	192	0.0379	**0.0408**	0.0370	0.0413
(5)	384	0.0385	**0.0425**	0.0382	0.0421
	768	0.0389	**0.0420**	0.0386	0.0411
vect. dépl. 16 éch.	192	0.0374	**0.0413**	0.0366	0.0411
(6)	384	0.0386	**0.0419**	0.0386	0.0420
	768	0.0381	**0.0429**	0.0379	0.0418
vect. accél. 7 éch.	192	0.0403	0.0396	0.0387	0.0372
(7)	384	0.0413	0.0412	0.0392	0.0376
	768	0.0412	0.0403	0.0390	0.0380
vect. accél. 15 éch.	192	0.0410	0.0421	0.0398	0.0388
(8)	384	0.0428	0.0431	0.0413	0.0411
	768	0.0444	0.0436	0.0430	0.0418
combinaisons :	vocab. K	AP	AP c.c.	AP diff.	AP c.c. diff.
C1 = 1+2+3+4+5+6+7+8	192	0.0423	**0.0443**	0.0416	0.0439
	384	0.0438	**0.0451**	0.0436	0.0440
C2 = C1 non c.c. +	192	0.0445	(same)	**0.0463**	(same)
+ C1 avec c.c.	384	0.0472	(same)	**0.0483**	(same)
C3 = BRIEF + (1 non c.c.)	1024	0.0551	-	0.0453	-
	2048	0.0514	-	0.0420	-
C4 = BRIEF + (1 non c.c.) +	1024	0.0541	(same)	0.0451	(same)
+ (1 avec c.c.)	2048	0.0517	(same)	0.0423	(same)

supérieurs à l'aléatoire. Parmi eux, 30 concepts ont été mieux détectés avec un descripteur basé sur des trajectoires qu'avec un descripteur plus spatial comme *SIFT retina* vu dans la section précédente.

Nous avons montré aussi que les différents descripteurs basés sur des trajectoires sont complémentaires, et même une fusion tardive simple, par moyenne arithmétique des différents "experts" que nous obtenons, arrive à exploiter cette complémentarité et augmente l'infAP jusqu'à 0,0670.

7.4.5 Conclusion

Nous avons montré que notre approche avec des Sacs de Mots de trajectoires fonctionne pour la reconnaissance non seulement d'actions sur une base simple (KTH), mais aussi pour la reconnaissance de concepts sémantiques plus génériques sur la base TRECVid, même si intuitivement, ces concepts ne semblent pas liés au mouvement. Les performances pourront être encore améliorées en faisant des optimisations additionnelles. Durant cette thèse, le coût de calcul élevé exigé pour chaque expérimentation nous a limité sur le nombre de configurations testées. Dans tous les cas, l'approche est validée et nous avons montré qu'un gain d'infAP peut être obtenu en faisant des fusions tardives. Dans la section suivante, nous explorons plus en détail les approches de fusion pour maximiser l'exploitation de la complémentarité entre différents descripteurs.

7.5 Fusion tardive de scores de classification

Comme nous l'avons vu précédemment, un seul "expert" (descripteur de shot + classifieur supervisé) ne peut pas être le meilleur pour chaque concept ; nous parlons alors d'une complémentarité au niveau des concepts. En plus, même pour un seul concept, certains experts peuvent mieux le détecter dans certaines conditions que dans d'autres ; nous parlons alors d'une complémentarité au niveau du contexte. Pour ces raisons, et pour obtenir une chaîne de traitement universelle, beaucoup de systèmes *fusionnent* un grand ensemble d'experts, chacun basé sur différents descripteurs et éventuellement aussi avec différents classifieurs supervisés.

Dans la chaîne de traitement que nous utilisons, nous combinons ces différentes informations en faisant des *fusions tardives* après l'étape de classification supervisée, comme illustré dans la Figure 7.2.

7.5.1 Principes

Nos approches de fusion tardive partent de la remarque suivante de *Ng et Kantor* :

> Les [experts] qui donnent des sorties dissimilaires mais avec des performances similaires vont plus probablement pouvoir constituer des fusions tardives simples et efficaces.

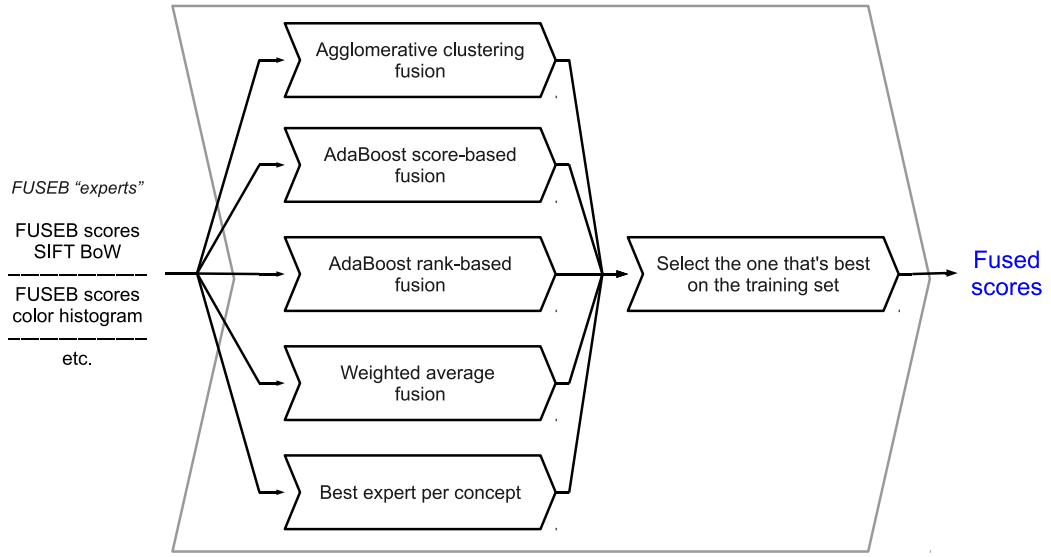


FIGURE 7.9 – Approche de fusion tardive proposée, appliquée indépendamment pour chaque concept à détecter : cinq fusions sont appliquées en parallèle sur les experts d'entrée, et la fusion qui a donné les meilleurs résultats sur la base d'entraînement est sélectionnée afin d' être utilisée pour la base de test.

C'est à dire que les experts qui donnent des infAP similaires, mais qui détectent les concepts-cibles dans des contextes différents, ont plus de chance de donner une augmentation de performance en faisant une fusion tardive par moyenne arithmétique (pondérée) des scores.

En se basant sur cette idée, dans le contexte de TRECVid, notre approche de fusion tardive comporte les étapes suivantes :

1. regroupement d'experts élémentaires qui sont similaires ;
2. fusion intra-groupe, qui donne un seul expert pour chaque groupe ; ces deux étapes ont le rôle d'équilibrer les différentes classes d'experts ;
3. fusion finale inter-groupe, qui donne le gain principal de performance en fusionnant les groupes (qui sont complémentaires à cause de la dissimilarité entre les groupes) ;

Nous appelons notre approche de fusion *"regroupement agglomératif"* et nous étudions sa performance en fusionnant un grand ensemble d'experts fournis par le consortium IRIM. De plus, nous testons aussi deux fusions AdaBoost [Freund 1997], une moyenne pondérée simple et une sélection du meilleur expert par concept. A la fin, nous ajoutons une couche de fusion supplémentaire, qui combine les résultats de toutes ces approches, comme illustré dans la Figure 7.9.

7.5.2 Résultats sur la base TRECVid 2013

Nous avons fait une première série d'expérimentations, qui nous a permis de mettre en évidence la complémentarité entre les descripteurs spatio-temporels basés sur des Sacs de

TABLE 7.4 – InfAP (moyenne sur tous les concepts) pour les différentes méthodes de fusion tardive : basique (sans aucun post-traitement), +RS (avec re-scoring temporel), +RS+CF (avec RS suivi par feedback conceptuel), RS+CF+RS (+RS+CF suivi par un deuxième re-scoring temporel).

	basique	+RS	+RS+CF	+RS+CF+RS
Fusion hiérarchique manuelle	0.2576	0.2695	0.2758	0.2848
AdaBoost basé sur scores	0.2500	0.2630	-	-
AdaBoost basé sur rangs	0.2346	0.2534	-	-
Regroupement agglomératif	0.2383	0.2516	-	-
Moyenne pondérée	0.2264	0.2409	-	-
Meilleur expert par concept	0.2162	0.2367	-	-
Selected best from 5 above	0.2495	0.2631	-	-

Mots SIFT utilisant la rétine, et les Sacs de Mots de trajectoires. Sur la base TRECVid SIN 2012, l'algorithme de fusion mentionné précédemment nous a permis d'augmenter l'infAP de 31% par rapport au meilleur "expert" par concept. Ces résultats sont détaillés dans la Section 5.4.1.

Après ces premières expérimentations, nous avons appliqué notre approche à un ensemble encore plus grand d'experts, fournis par le consortium IRIM. Les experts d'entrée étaient très divers : histogrammes couleur, ondelettes quaternioniques, filtres de Gabor, Sacs de Mots de caractéristiques locales, trajectoires, descripteurs audio, présence de concepts sémantiques de niveau intermédiaire etc.

Nous avons ensuite amélioré les résultats en utilisant l'algorithme de re-scoring temporel de [Safadi 2011] pour prendre en compte la corrélation temporelle entre les shots consécutifs d'un même vidéo, et l'algorithme de feedback conceptuel de [Hamadi 2013] pour prendre en compte les relations entre les concepts. Nous comparons nos fusions automatiques avec une fusion hiérarchique manuelle présentée dans [Ballas 2012b].

La Table 7.4 montre les résultats obtenus par les différentes fusions tardives. La fusion hiérarchique manuellement optimisée est la meilleure, mais les méthodes automatiques ne sont pas loin. Parmi ces dernières, la méthode AdaBoost basée sur les scores de classification et la Sélection de la meilleure approche sur la base d'entrainement donnent les meilleurs résultats. Par rapport au meilleur expert par concept, ces deux approches augmentent les performances d'environ 16%. Plus de détails sont donnés dans le Chapitre 5, y compris une analyse concept par concept.

7.5.3 Conclusion concernant la fusion

Les approches de fusion tardive que nous avons testées montrent qu'elles sont capables d'exploiter la complémentarité entre les différents experts et de donner des gains de performance substantiels. Ces gains sont importants surtout quand les experts fusionnés sont de types différents, donc sensibles aux différents aspects de la vidéo. En plus, les fusions d'un niveau supérieur (contexte temporel et sémantique) peuvent donner un gain supplémentaire d'infAP.

7.6 Conclusions et perspectives

Dans ces travaux, nous avons exploré le sujet de l'indexation sémantique de contenu vidéo divers. Nous avons pris une chaîne de traitement classique (Figure 7.2) que nous avons enrichie avec des descriptions spatio-temporelles et avec des méthodes de fusion d'information.

Nos expérimentations ont montré que la stratégie proposée de pré-traitement rétinien des vidéos peut servir à donner un ensemble de descripteurs vidéo plus performants et complémentaires, qui sont un bon compromis entre la complexité de calcul et la qualité des résultats d'indexation sémantique. La diversité spatio-temporelle a été étendue encore vers le mouvement en incluant un ensemble de descripteurs basés sur des Sacs de Mots de trajectoires, qui ont été validés sur les bases KTH et TRECVid.

Enfin, cet ensemble déjà complémentaire d'"experts" ajouté à celui fournis par le consortium IRIM, nous a encouragé à développer des algorithmes automatiques de fusion tardive. Ces algorithmes de fusion nous ont permis de bénéficier de l'information complémentaire donnée par les différents experts et d'améliorer les résultats d'indexation sémantique.

Parmi les futures directions d'étude, nous pouvons énoncer une étude de l'impact d'un traitement multi-échelle pour les Sacs de Mots SIFT/SURF utilisant la rétine. Ce travail est en cours et nous sommes en train de chercher un bon paramétrage de la grille dense multi-échelle, de la rétine et de la fenêtre temporelle.

Une deuxième direction d'étude à court terme est l'extension du pré-traitement rétinien aux autres types de caractéristiques locales comme FREAK [Alahi 2012], pour vérifier si on obtient un comportement similaire à celui obtenu avec les descripteurs SIFT et SURF.

Comme la réponse de la rétine est influencée par le mouvement de la caméra, une autre direction d'étude serait une rétine avec compensation du mouvement de la caméra. L'effet principal de cette amélioration sera une réponse plus faible de la voie magnocellulaire sur les zones d'arrière-plan en mouvement à cause de la caméra mobile.

A long terme, il serait intéressant d'inclure dans notre traitement bio-inspiré, des niveaux supérieurs du système visuel humain, par exemple le cortex V1, qui peut servir à la reconnaissance d'objets ou de visages et au suivi [Benoit 2010]. Il serait encore plus intéressant d'inclure des couches supérieures au cortex V1 afin de simuler une grande partie du système visuel, mais le fonctionnement de toutes ces zones corticales n'est pas encore connu en détail [Hérault 2010].

Concernant les descripteurs Sacs de Mots de trajectoires, leur paramétrage peut être encore optimisé. En plus, nous pouvons tester d'autres façons de décrire une trajectoire, chacune sensible à un certain aspect du mouvement mais robuste à un autre, afin d'avoir des représentations encore plus complémentaires. Il serait intéressant aussi de tester d'autres modèles pour agréger les différentes trajectoires, pas seulement le modèle Sac de Mots. Le modèle Actom Sequence Model [Gaidon 2011] donne des performances améliorées, mais dans notre contexte TRECVid, il pose des problèmes pour l'étape d'entrainement, à cause du manque d'annotations détaillées.

Finalement, les approches de fusion tardive pourront être étendues aux autres types de données multimédia, car elles sont indépendantes de la nature exacte des experts. Par

exemple, il serait intéressant de tester ces approches sur une base d'images statiques comme ImageCLEF[4].

En conclusion, les sujets traités dans cette thèse ouvrent plusieurs pistes d'expérimentation et d'applications liées à l'indexation sémantique de bases vidéo. D'un point de vue applicatif, ce travail peut être utilisé dans plusieurs domaines, comme l'indexation sémantique de vidéos téléchargées dans les collections en-ligne, l'indexation des archives vidéo des chaînes de télévision, les application de vidéo à la demande dans lesquelles un utilisateur cherche des vidéos similaires à une requête, ou même dans la vidéo surveillance, dans laquelle les objets ou personnes suspectes pourront être détectés automatiquement.

[4] http://www.imageclef.org/

Appendix A

The human retina model

Contents

The eyes are the first component of the visual system. An optical system forms an image of the world around us onto a layer inside the eye called the *retina* (see Figure A.1). Photoreceptor cells are located on the retina, which convert photons into neural signals. These retinal signals undergo several processing steps right inside the retina, before being sent down the optic nerve to the next components of the visual system (via the Lateral Geniculate Nucleus to the Primary Visual Cortex (V1), and also to the Superior Colliculus) [Hérault 2010].

There are many models of parts of the human visual system for various applications: the Retinex filter for enhancing digital images [Jobson 1997], [Senane 2001] for information coding, [VanRullen 2002] that models neural impulses (spikes) at the retinal ganglion and visual cortex levels, [Walther] that models top-down interactions but does not include complete low-level retinal processing, or [Daly 1994, Le Meur 2006a, Marat 2008] that deal mainly with what happens beyond the V1 cortex.

For our application of extending SIFT/SURF descriptors, we decided upon the retinal model from [Benoit 2010], as it includes the low-level spatio-temporal retinal processing that we need. We describe this model in the following.

The retina is composed of the Outer Plexiform Layer (OPL) and the Inner Plexiform Layer (IPL) (Figure A.2). Biologically, the OPL contains photoreceptors (cells sensitive to light) and horizontal cells that interconnect the photoreceptors. The IPL contains bipolar cells, ganglion cells and amacrine cells. The two retinal outputs that we use in Chapter 3, the parvocellular and magnocellular channel, are obtained at the output of the IPL. The implementation of the OPL and IPL is detailed in the following section.

A.1 The Outer Plexiform Layer

The photoreceptors are the first “cells" in the processing chain. They can adjust their sensitivity according to the local luminance of their neighborhood, performing luminance

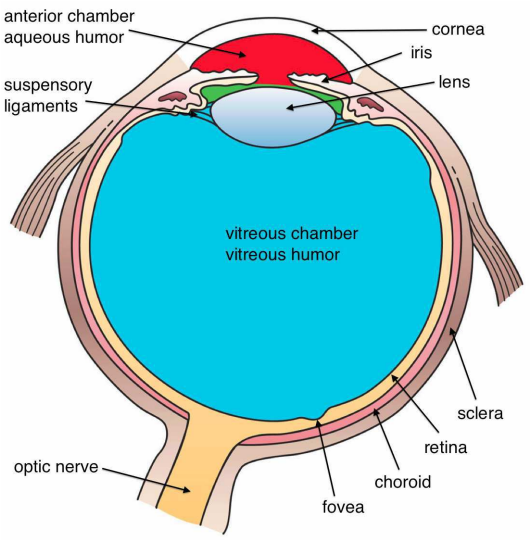


Figure A.1: Internal structure of the human eye. Image source: `http://upload.wikimedia.org/wikipedia/commons/8/8a/Three_Internal_chambers_of_the_Eye.png`

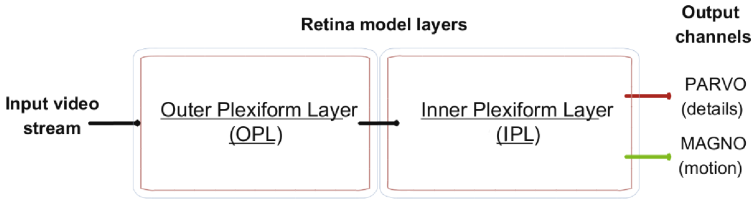


Figure A.2: The retina model from [Benoit 2010] contains two layers: the Outer Plexiform Layer (OPL) and the Inner Plexiform Layer (IPL). Two output channels are generated at the IPL: the parvocellular channel dealing with spatial details, and the magnocellular channel dealing with motion. Figure credit: [Benoit 2010]

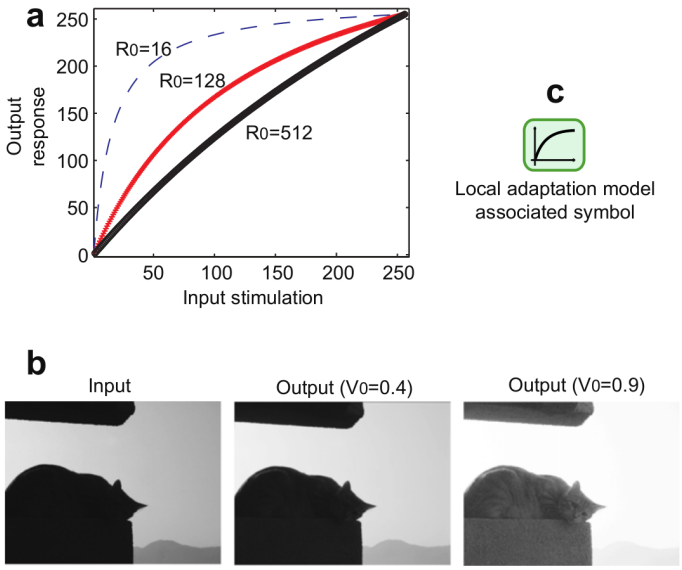


Figure A.3: Photoreceptor adaptation to local luminance: (a) illustrates compression curves for different values of R_0 (lower values give stronger amplification in dark areas); (b) illustrates the effect of an image with a very dark area; (c) is the symbol for local adaptation that will be used in other figures. Figure credit:[Benoit 2010]

compression, as in Equation A.1 [Benoit 2010]:

$$\begin{aligned} C(p) &= \frac{R(p)}{R(p) + R_0(p)} \cdot V_{max} + R_0(p) \\ R_0(p) &= V_0 \cdot L(p) + V_{max}(1 - V_0) \end{aligned} \tag{A.1}$$

where p is a photoreceptor, $R(p)$ is the current luminance at the photoreceptor, $C(p)$ is the corrected luminance, $R_0(p)$ is the compression parameter which is determined by the local luminance $L(p)$ (more about the local luminance later). V_{max} is the maximum allowed pixel value (255 for 8-bit images) and V_0 is a parameter in the range $[0; 1]$ for adjusting the strength of the local adaptation effect (a value of 0.90 is generally good). Examples of compression curves for different levels of local luminance are given in Figure A.3 a. The effect of the photoreceptor adaptation to local luminance is a greater amplification in dark areas of an image, making details more visible (see Figure A.3).

After luminance adaptation, the photoreceptors and the horizontal cells of the OPL each perform a spatio-temporal filtering of the signal, represented by F_{ph} and F_h in Figure A.4a, modeled by the following equations:

$$F_{OPL}(f_s, f_t) = F_{ph}(f_s, f_t) \cdot [1 - F_h(f_s, f_t)] \tag{A.2}$$

where

$$F_{ph}(f_s, f_t) = \frac{1}{1 + \beta_{ph} + 2\alpha_{ph} \cdot (1 - \cos(2\pi f_s)) + j2\pi\tau_{ph} f_t} \tag{A.3}$$

$$F_h(f_s, f_t) = \frac{1}{1 + \beta_h + 2\alpha_h \cdot (1 - \cos(2\pi f_s)) + j2\pi\tau_h f_t} \tag{A.4}$$

$F_{OPL}(f_s, f_t)$ is the transfer function of the OPL. f_s and f_t are spatial and temporal frequencies (we are dealing with discrete time and space signals, therefore f_s and f_t are in the range $[-0.5; 0.5]$, where 1 corresponds to the sampling frequency). F_{ph} and F_h are the transfer functions of the photoreceptors and of the horizontal cells, which depend on β_{ph}, α_{ph}, τ_{ph} and β_h, α_h, τ_h respectively.

F_{ph} and F_h attenuate high spatial frequencies and high temporal frequencies; $1 - F_h$ has an opposite effect, attenuating low spatial and temporal frequencies. When combined in F_{OPL}, this creates a spatio-temporal band-pass effect (the spatial and temporal constants of F_{ph} and F_h are not the same), illustrated in Figure A.4b.

For low temporal frequencies (static or almost static images), F_{OPL} has a spatial band-pass behaviour, while for higher temporal frequencies, it has a spatial low-pass effect; for low spatial frequencies, it has a temporal band-pass effect (of wide band), and for higher spatial frequencies, it has a temporal low-pass effect. This has the effect of enforcing local contrasts (mid spatial frequencies) that do not move a lot, while reducing noise (which occupies the high spatial and temporal frequencies) [Benoit 2010].

Concerning the filter parameters, it can be said that β_h of F_h regulates the transmission of the (local) continuous component of the video: if $\beta_h = 0$, then $F_h(0,0) = 1$, therefore $F_{OPL}(0,0) = 0$ from Equation A.2. If it is desired to let some of the continuous component pass through F_{OPL}, then a higher value for β_h can be set.

Also, because F_{ph} and F_h reject high spatial and temporal frequencies, the output at F_h contains very low spatial frequencies (F_{ph} and F_h are cascaded in Figure A.4a). Therefore, the output of F_h can be used as the local luminance $L(p)$ in Equation A.1.

Regarding the bipolar cells performing the subtraction in F_{OPL} from Equation A.2, it is to note that biological neurons cannot encode negative values, therefore, a Bipolar ON signal encodes the positive part of the difference, while a Bipolar OFF signal encodes the negative part (see Figure A.4a).

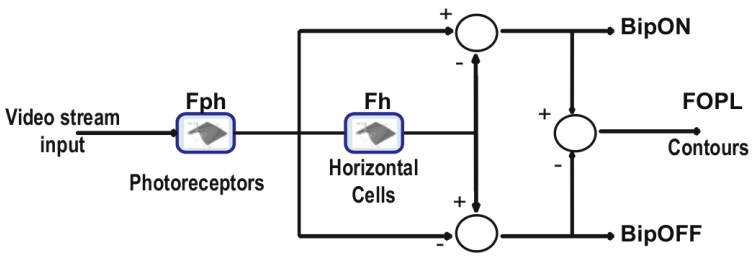
A.2 The Inner Plexiform Layer

After the OPL, the Bipolar ON and OFF signals, which are the positive and negative parts of the F_{OPL} filter response, are passed on to the IPL. The IPL further processed these signals and generates the two retinal outputs: the *parvocellular* channel and the *magnocellular* channel.

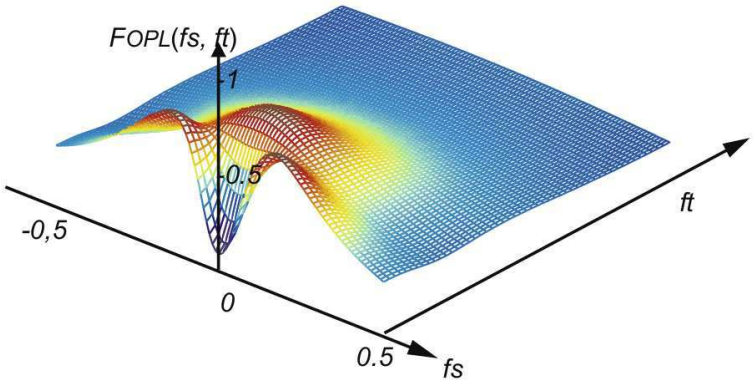
A.2.1 The parvocellular channel

To construct the *parvocellular* channel, the ganglion cells from the IPL perform a logarithmic compression (which resembles the one done by the photoreceptors) of the BipON and BipOFF signals, as shown in Figure A.5a, amplifying contrast in these signals. Afterwards, the two signals are recombined to form the parvocellular channel, one of the two main outputs of the human retina.

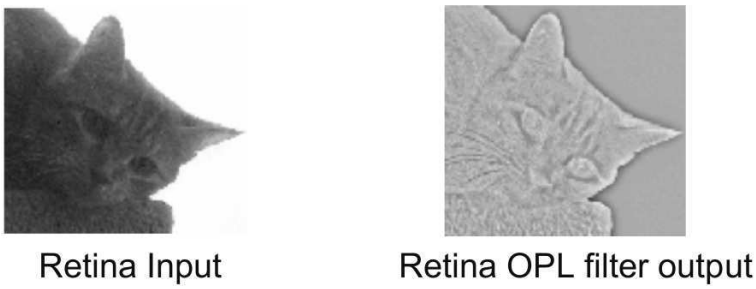
The images coming out of the parvocellular channel will have attenuated low-frequency components, attenuated moving details (motion blur) and reduced high-frequency noise thanks to the F_{OPL} filtering, and stronger contours (medium spatial frequencies which are



(a) Interconnections of the OPL

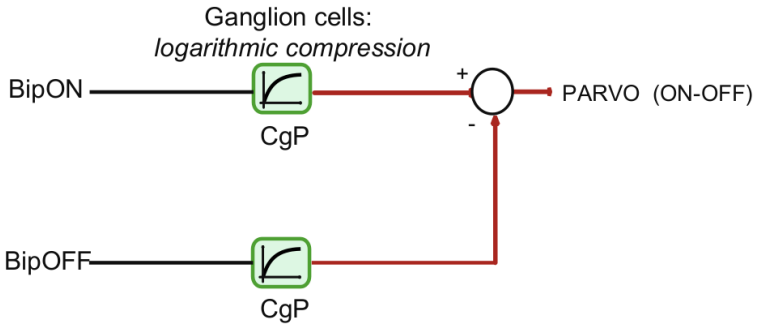


(b) Transfer function of the OPL

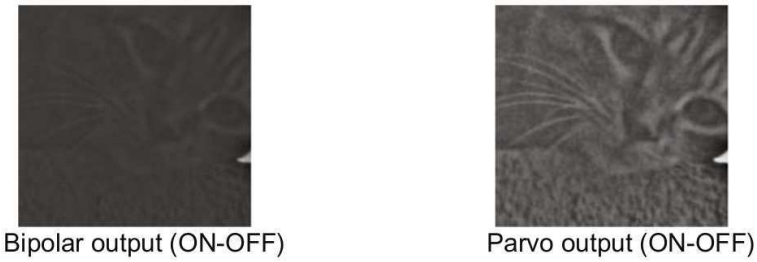


(c) Output example of the OPL

Figure A.4: The OPL model. In A.4a, bipolar cells subtract the signals coming from the photoreceptors and from the horizontal cells. A.4b shows the spatio-temporal transfer function of the OPL. A.4c gives an example of the output FOPL from A.4a: only contour information is kept (gray corresponds to 0, white to positive values and black to negative values). Figure credit:[Benoit 2010]

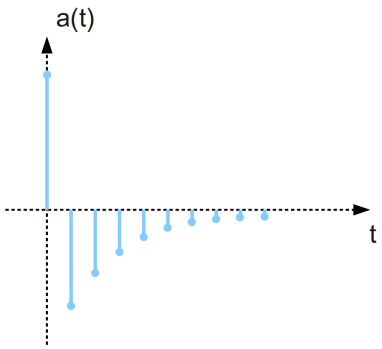


(a) Forming the parvocellular channel

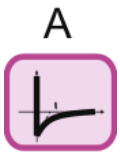


(b) Ganglion cell contrast boosting

Figure A.5: Ganglion cells boost contrast in the BipON and BipOFF signals and these signals are then recombined to form the output parvocellular channel. Figure credit:[Benoit 2010]



(a) Amacrine cell impulse response



(b) Amacrine cell symbol

Figure A.6: Impulse response of amacrine cells; amacrine cell symbol from [Benoit 2010]

passed by F_{OPL}) in regions that do not move a lot (at low temporal frequencies in F_{OPL}) thanks to the contrast-boosting ganglion cells.

A.2.2 The magnocellular channel

The other retinal output, the magnocellular channel, is also obtained from the BipON and BipOFF signals, but in a different manner. The BipON and BipOFF signals first undergo a temporally high-pass filtering by amacrine cells in the IPL, followed by spatio-temporal filtering and contrast boosting in ganglion cells. Figure A.7 illustrates the processing chain for the magnocellular channel.

The amacrine cells have a transfer function of the following form [Benoit 2010]:

$$A(z) = b \cdot \frac{1 - z^{-1}}{1 - b \cdot z^{-1}} \quad \text{(A.5)}$$

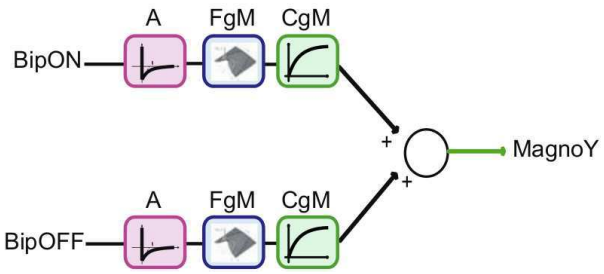
with

$$b = e^{-\Delta t / \tau_A}$$

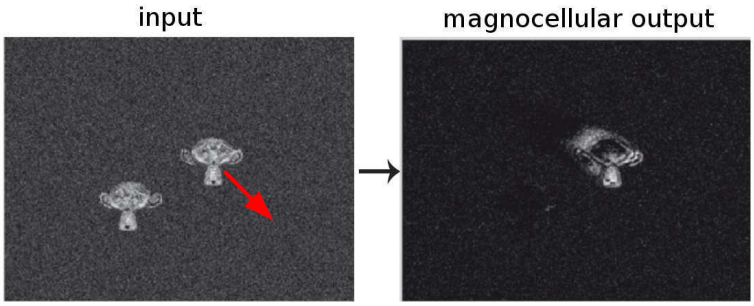
where $\Delta t = 1$ is the discrete time step, and τ_A is the temporal constant of the filter. This gives an impulse response similar to the one in Figure A.6a, which constitutes a high-pass temporal filter.

After filtering by the amacrine cells, the signals are sent to other ganglion cells that perform a spatio-temporal filtering step FgM (similar to F_{ph} or F_h) and then contrast boosting through logarithmic compression CgM similar to what was done for the parvocellular channel. At the end, the filtered and contrast-boosted signals are recombined to produce the *magnocellular* channel, as in Figure A.7.

The amacrine cells compensate for the attenuation of high temporal frequencies by the F_{ph}, F_h and FgM filters to give sensitivity to temporal events (motion) and attenuate low temporal frequencies. This retains only moving contours in the video (especially contours perpendicular to the motion direction), and the visibility of these moving contours is increased by the compression step CgM.



(a) Magno. channel diagram



(b) Magno. output example

Figure A.7: To form the magnocellular channel, amacrine cells filter the Bipolar On and Off signals, followed by spatio-temporal filtering and contrast boosting by ganglion cells (A.7). In A.7b, the upper-right element is moving and is highlighted by the magnocellular channel, while the stationary bottom-left element gives no response. Figure credit: [Benoit 2010]

The magnocellular channel can therefore act as a detector of moving elements (moving contours), while at the same time attenuating high-frequency spatio-temporal noise, as can be see in Figure A.7b [Benoit 2010]: only the moving element appears in the output, while noise is reduced.

A.3 Behaviour of the retina model

A synthesis of all that has been said so far about the retinal model is that:

- The *parvocellular* channel processes spatial details; it enhances local contrast, intensifies contours, removes high-frequency noise and responds well to temporally-sustained signals, while smoothing out fast temporal variations. Even compression artifacts are reduced, as long as they are not identical from one frame to the next. This channel is also concerned with color information processing and it can normalize colors thanks to the photoreceptor adaptation process.

- The *magnocellular* channel deals with spatio-*temporal* events, such as contours in motion or objects appearing or disappearing from the frame. It does not process color information, giving a grayscale output.

An effect that has not been stated before is that the retina exhibits a transient state during a certain number of frames after processing has started. The start of processing is the equivalent of "opening the eyes" (an abrupt transition from a black frame to the image sequence of interest), but transient phases can also occur on videos during abrupt transitions such as cuts, or when an object suddenly appears is the scene. The transient phase is characterized by the following phenomena:

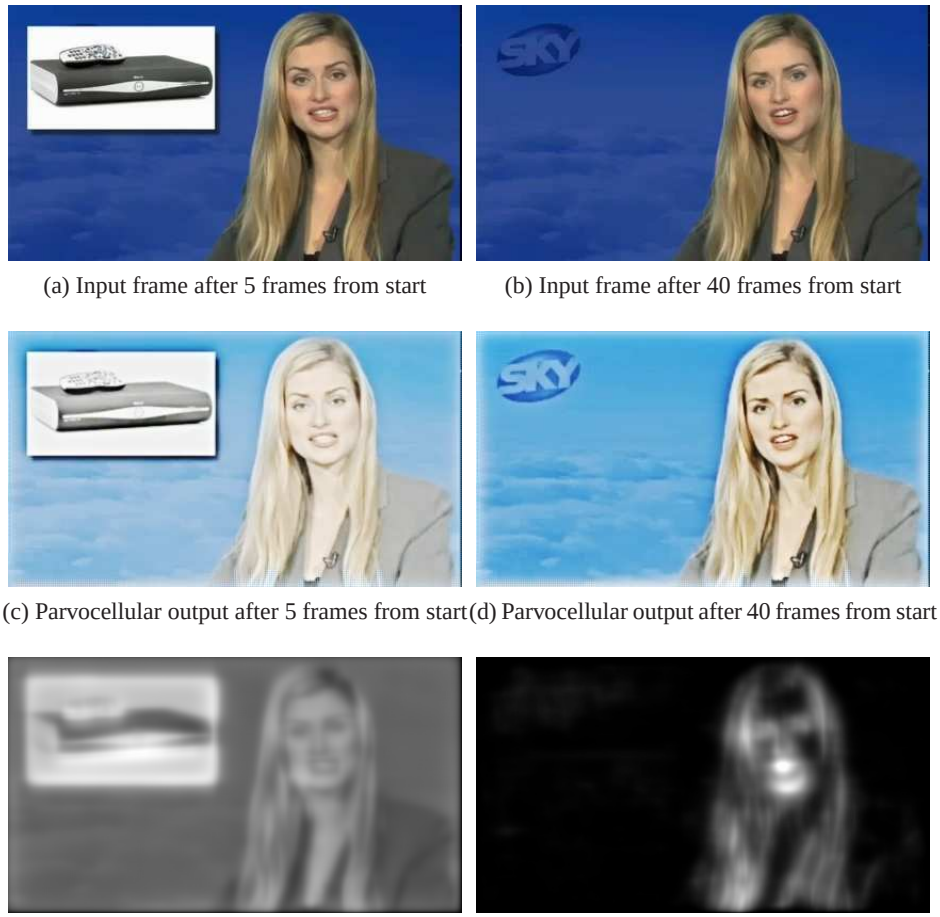
- The *parvocellular* channel outputs information in a "coarse-to-fine" way. At the onset of the spatio-temporal event (such as "opening the eyes"), only low spatial frequencies are transmitted; this is because the appearance of a new object or scene is a high temporal frequency element, and according to the F_{OPL} response from Figure A.4b, spatial details at high temporal frequencies are smoothed-out. But if after its appearance, the object (or the new scene) remains stationary, the parvocellular channel will start to transmit (and enforce) spatial details. This coarse-to-fine processing model is not unlike what happens in the Human Visual System: when examining a new scene, the retina supplies the brain with a coarse, low-resolution image, to get a general idea of the scene content; only afterwards does it supply more spatially-detailed information.

- At the onset of a new visual scene, the *magnocellular* channel briefly transmits low spatial frequencies, and a strong response is generated on large spatial boundaries until the end of the transient phase. During the transient phase, the magnocellular channel can therefore be used as a detector of potential spatial areas of interest. After the retina reaches its stable state, the magnocellular channel only fires on moving parts, therefore the channel now acts as a transient area detector and more generally as a motion detector.

In the retinal model that we use, the parvocellular channel is implemented as a sequence of color images with enhanced spatial details, corrected colors (with respect to the color temperature), enhanced details in the shadows and also reduced noise and reduced video compression artifacts. The magnocellular channel is implemented as a sequence of gray-level images, and we use it as a low-level spatio-temporal region of interest detector, responding to spatial features during the transient state and to moving contours afterwards.

An example of parvocellular and magnocellular responses is given in Figure A.8, in which a TV presenter is talking. Depending on the temporal constants chosen for the retinal filters, the transient state usually lasts between 10 and 20 frames. After 5 frames since "opening the eyes", the retina is still in the transient state, while after 40 frames, it is in its stable state. In the transient state, the parvocellular channel has not yet started to regulate the mean luminance and to boost spatial details (Figure A.8c, while the magnocellular channel passes low spatial frequencies (Figure A.8e).

In the stable state, spatial detail enhancement in the parvocellular frame can be seen around the facial features, around the logo and in the details in the clouds; the increase in local contrast can even produce halo effects, such as those around the presenter's hair in Figure A.8d. In the stable state, the magnocellular channel responds only to moving elements such as the presenter's head and lips in Figure A.8f.



(a) Input frame after 5 frames from start

(b) Input frame after 40 frames from start

(c) Parvocellular output after 5 frames from start

(d) Parvocellular output after 40 frames from start

(e) Magnocellular channel output after 5 frames from start

(f) Magnocellular channel output after 40 frames from start

Figure A.8: Retinal processing example, after respectively 5 and 40 frames since the start of the processing (the initialization of the retina). After 5 frames, the retina is still in its transient phase: the parvocellular channel passes a large amount of luminance and details are not yet enhanced too much, while the magnocellular channel fires on large spatial structures. After 40 frames, the retina is in its stable state: the parvocellular channel passes less luminance and enhances spatial details, while the magnocellular channel fires mainly on moving areas (the presenter's face).

List of publications

Scientific journals:

Strat 2013a Sabin Tiberius Strat, Alexandre Benoit, Patrick Lambert and Alice Caplier. Retina enhanced SURF descriptors for spatio-temporal concept detection. Multimedia Tools and Applications, pages 1–27, 2013.

Book chapters:

- Sabin Tiberius Strat, Alexandre Benoit, Patrick Lambert, Hervé Bredin and Georges Quénot. Hierarchical late fusion for concept detection in videos (extended book chapter of [Strat 2012b]), submitted for review in ECCV IFCVCR 2012 workshop book)

International conferences and workshops:

Strat 2012a S.T. Strat, A. Benoit, P. Lambert and A. Caplier. Retina-enhanced SURF descriptors for semantic concept detection in videos. In Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on, pages 319–324, 2012

Strat 2012b Tiberius Strat, Alexandre Benoit, Hervé Bredin, Georges Quénot and Patrick Lambert. Hierarchical Late Fusion for Concept Detection in Videos. In Proceedings of European Conference of Computer Vision - ECCV 2012. Workshops and Demonstrations, Part III, volume 7585 of Lecture Notes in Computer Science (LNCS), pages 335–344, Firenze, Italy, October 2012. Springer Berlin. Oral session 1: WS21 - Workshop on Information Fusion in Computer Vision for Concept Recognition OSEO (French State agency for innovation) and ANR (French national research agency).

Strat 2013b S.T. Strat, A. Benoit and P. Lambert. Retina enhanced SIFT descriptors for video indexing. In Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on, pages 201–206, 2013

TRECVid workshops as members of the IRIM consortium:

- IRIM at TRECVID 2011: Semantic Indexing and Instance Search, 2011 TREC Video Retrieval Evaluation Notebook Papers and Slides
- IRIM at TRECVID 2012: Semantic Indexing and Instance Search, 2012 TREC Video Retrieval Evaluation Notebook Papers and Slides

Bibliography

[Alahi 2012] Alexandre Alahi, Raphaël Ortiz and Pierre Vandergheynst. *FREAK: Fast Retina Keypoint*. In IEEE Conference on Computer Vision and Pattern Recognition, 2012. CVPR 2012 Open Source Award Winner. (Cited on pages 36, 63, 121 and 150.)

[Ali 2011] Wafa Bel Haj Ali, Eric Debreuve, Pierre Kornprobst and Michel Barlaud. *Bio-inspired Bags-of-features for Image Classification*. In KDIR, pages 277–281, 2011. (Cited on page 36.)

[Arthur 2007] David Arthur and Sergei Vassilvitskii. *k-means++: the advantages of careful seeding*. In SODA, pages 1027–1035, 2007. (Cited on pages 15, 50, 59, 79 and 133.)

[Ayache 2007] Stéphane Ayache, Georges Quénot and Jérôome Gensel. *Image and video indexing using networks of operators*. J. Image Video Process., vol. 2007, no. 3, pages 1:1–1:13, November 2007. (Cited on page 112.)

[Ballas 2011] Nicolas Ballas, Betrand Delezoide and Françoise Prêteux. *Trajectories based descriptor for dynamic events annotation*. In Proceedings of the 2011 joint ACM workshop on Modeling and representing events, J-MRE '11, pages 13–18, New York, NY, USA, 2011. ACM. (Cited on pages 26, 66, 77 and 134.)

[Ballas 2012a] Nicolas Ballas, Benjamin Labbé, Aymen Shabou and Le Borgne. *CEA LIST at TRECVID 2012: Semantic Indexing and Instance Search*. In Proc. TRECVID Workshop, Gaithersburg, MD, USA, november 2012. (Cited on pages 26 and 111.)

[Ballas 2012b] Nicolas Ballas, Benjamin Labbé, Aymen Shabou, Hervé Le Borgne, Philippe Gosselin, Miriam Redi, Bernard Merialdo, Hervé Jégou, Jonathan Delhumeau, Rémi Vieux, Boris Mansencal, Jenny Benois-Pineau, Stéphane Ayache, Abdelkader Hamadi, Bahjat Safadi, Franck Thollard, Nadia Derbas, Georges Quenot, Hervé Bredin, Matthieu Cord, Boyang Gao, Chao Zhu, Yuxing Tang, Emmanuel Dellandrea, Charles-Edmond Bichot, Liming Chen, Alexandre Benoit, Patrick Lambert, Tiberius Strat, Joseph Razik, Sébastien Paris, Hervé Glotin, Tran Ngoc Trung, Dijana Petrovska-Delacrétaz, Gérard Chollet, Andrei Stoian and Michel Crucianu. *IRIM at TRECVID 2012: Semantic Indexing and Instance Search*. In Proceedings of the workshop on TREC Video Retrieval Evaluation (TRECVID), page 12p., Gaithersburg, MD, États-Unis, November 2012. CNRS, RENATER, several Universities, other funding bodies (see https://www.grid5000.fr). (Cited on pages 6, 11, 12, 13, 15, 28, 62, 80, 98, 107, 108, 110, 112, 132 and 149.)

[Bay 2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool. *Speeded-Up Robust Features (SURF)*. Comput. Vis. Image Underst., vol. 110, no. 3, pages 346–359, June 2008. (Cited on pages 6, 18, 21, 50, 128 and 133.)

[Benoit 2010] A. Benoit, A. Caplier, B. Durette and J. Herault. *Using Human Visual System modeling for bio-inspired low level image processing*. Computer Vision and Image Understanding, vol. 114, no. 7, pages 758 – 773, 2010. (Cited on pages 6, 29, 36, 50, 56, 59, 122, 128, 135, 136, 150, 153, 154, 155, 156, 157, 158, 159, 160 and 161.)

[Birchfield 2007] S. Birchfield. *KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker*. http://www.ces.clemson.edu/stb/klt/, 2007. (Cited on pages 25 and 69.)

[Bobick 2001] Aaron F. Bobick and James W. Davis. *The Recognition of Human Movement Using Temporal Templates*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 3, pages 257–267, March 2001. (Cited on page 22.)

[Bouguet 2000] Jean-Yves Bouguet. *Pyramidal implementation of the Lucas Kanade feature tracker*. Intel Corporation, Microprocessor Research Labs, 2000. (Cited on pages 69 and 143.)

[Breiman 1996] Leo Breiman and Leo Breiman. *Bagging Predictors*. In Machine Learning, pages 123–140, 1996. (Cited on page 27.)

[Brendel 2010] William Brendel and Sinisa Todorovic. *Activities as time series of human postures*. In Proceedings of the 11th European conference on Computer vision: Part II, ECCV'10, pages 721–734, Berlin, Heidelberg, 2010. Springer-Verlag. (Cited on page 22.)

[Cai 2007] Na Cai, Ming Li, Shouxun Lin, Yongdong Zhang and Sheng Tang. *AP-Based Adaboost in High Level Feature Extraction at TRECVID*. In Pervasive Computing and Applications, 2007. ICPCA 2007. 2nd International Conference on, pages 194–198, 2007. (Cited on pages 28 and 100.)

[Calonder 2010] Michael Calonder, Vincent Lepetit, Christoph Strecha and Pascal Fua. *BRIEF: binary robust independent elementary features*. In Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. (Cited on pages 21, 133 and 143.)

[Cao 2012] L. Cao, S.-F. Chang, N. Codella, C. Cotton, D. Ellis, L. Gong, M. Hill, G. Hua, J. Kender, M. Merler, Y. Mu, J. R. Smith and X. Yu Felix. *IBM Research and Columbia University TRECVID-2012 Multimedia Event Detection (MED), Multimedia Event Recounting (MER), and Semantic Indexing (SIN) Systems*. In NIST TRECVID Workshop, Gaithersburg, MD, December 2012. (Cited on page 28.)

[Chang 2001] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a Library for Support Vector Machines*, 2001. (Cited on page 25.)

[Chen 2009] M.-Y. Chen and A. Hauptmann. *MoSIFT: Recognizing Human Actions in Surveillance Videos*. Rapport technique CMU-CS-09-161, Carnegie Mellon University, 2009. (Cited on pages 25, 45, 66 and 134.)

[Cliville 2004] V. Cliville, L. Berrah and G. Mauris. *Information fusion in industrial performance: a 2-additive Choquet-integral based approach*. In Systems, Man and Cybernetics, 2004 IEEE International Conference on, volume 2, pages 1297–1302 vol.2, 2004. (Cited on pages 27 and 135.)

[Csurka 2004] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski and Cédric Bray. *Visual categorization with bags of keypoints*. In In Workshop on Statistical Learning in Computer Vision, ECCV, pages 1–22, 2004. (Cited on pages 15, 16, 132 and 133.)

[Daly 1994] S. Daly. *A visual model for optimizing the design of image processing algorithms*. In Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, volume 2, pages 16–20 vol.2, 1994. (Cited on page 153.)

[de Carvalho Soares 2012] R. de Carvalho Soares, I.R. da Silva and D. Guliato. *Spatial Locality Weighting of Features Using Saliency Map with a Bag-of-Visual-Words Approach*. In Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th International Conference on, volume 1, pages 1070–1075, Nov 2012. (Cited on page 40.)

[Delezoide 2011] Bertrand Delezoide, Frédéric Precioso, Philippe Gosselin, Miriam Redi, Bernard Mérialdo, Lionel Granjon, Denis Pellerin, Michele Rombaut, Hervé Jégou, Rémi Vieux, Boris Mansencal, Jenny Benois Pineau, Stéphane Ayache, Bahjat Safadi, Franck Thollard, Georges Quénot, Hervé Bredin, Matthieu Cord, Beno&#305, Alexandre t, Patrick Lambert, Tiberius Strat, Joseph Razik, Sébastion Paris and Hervé Glotin. *IRIM at TRECVID 2011: Semantic indexing and instance search*. In TRECVID 2011, 15th International Workshop on Video Retrieval Evaluation, 2011, National Institute of Standards and Technology, Gaithersburg, USA, Gaithersburg, UNITED STATES, 11 2011. (Cited on pages 15, 80 and 132.)

[Deng 2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. In CVPR09, 2009. (Cited on page 112.)

[Dollár 2005] Piotr Dollár, Vincent Rabaud, Garrison Cottrell and Serge Belongie. *Behavior recognition via sparse spatio-temporal features*. In In VS-PETS, pages 65–72, 2005. (Cited on pages 23 and 134.)

[Everingham 2010a] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The Pascal Visual Object Classes (VOC) Challenge*. International Journal of Computer Vision, vol. 88, no. 2, pages 303–338, June 2010. (Cited on page 30.)

[Everingham 2010b] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn and Andrew Zisserman. *The Pascal Visual Object Classes (VOC) Challenge*. Int. J. Comput. Vision, vol. 88, no. 2, pages 303–338, June 2010. (Cited on page 18.)

[Ewerth 2004] R. Ewerth, M. Schwalb, P. Tessmann and B. Freisleben. *Estimation of arbitrary camera motion in MPEG videos*. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 1, pages 512–515 Vol.1, 2004. (Cited on page 71.)

[Fei-Fei 2007] Li Fei-Fei, Rob Fergus and Pietro Perona. *Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories*. Comput. Vis. Image Underst., vol. 106, no. 1, pages 59–70, April 2007. (Cited on page 30.)

[Fischler 1981] Martin A. Fischler and Robert C. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Commun. ACM, vol. 24, no. 6, pages 381–395, June 1981. (Cited on page 73.)

[Fleet 2005] David J. Fleet and Yair Weiss. *Optical Flow Estimation*, 2005. (Cited on page 67.)

[Freund 1997] Yoav Freund and Robert E Schapire. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Journal of Computer and System Sciences, vol. 55, no. 1, pages 119–139, 1997. (Cited on pages 28, 103, 135 and 148.)

[Gaidon 2011] A. Gaidon, Z. Harchaoui and C. Schmid. *Actom sequence models for efficient action detection*. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11, pages 3201–3208, Washington, DC, USA, 2011. IEEE Computer Society. (Cited on pages 17, 23, 79, 123 and 150.)

[Gönen 2011] Mehmet Gönen and Ethem Alpaydın. *Multiple Kernel Learning Algorithms*. J. Mach. Learn. Res., vol. 12, pages 2211–2268, July 2011. (Cited on page 27.)

[González Díaz 2013] Iván González Díaz, Vincent Buso, Jenny Benois-Pineau, Guillaume Bourmaud and Rémi Megret. *Modeling Instrumental Activities of Daily Living in Egocentric Vision As Sequences of Active Objects and Context for Alzheimer Disease Research*. In Proceedings of the 1st ACM International Workshop on Multimedia Indexing and Information Retrieval for Healthcare, MIIRH '13, pages 11–14, New York, NY, USA, 2013. ACM. (Cited on page 40.)

[Gorisse 2010] David Gorisse, Frédéric Precioso, Philippe Gosselin, Lionel Granjon, Denis Pellerin, Michèle Rombaut, Hervé Bredin, Lionel Koenig, Rémi Vieux, Boris Mansencal, Jenny Benois-Pineau, Hugo Boujut, Claire Morand, Hervé Jégou, Stéphane Ayache, Bahjat Safadi, Yubing Tong, Franck Thollard, Georges Quénot M., Matthieu Cord, Alexandre Benoit and Patrick Lambert. *IRIM at TRECVID 2010: Semantic Indexing and Instance Search*. In TREC online proceedings, pages –, Gaithersburg, États-Unis, November 2010. GDR ISIS. (Cited on pages 51, 53 and 59.)

[Gosselin 2008] Philippe Henri Gosselin, Matthieu Cord and Sylvie Philipp-Foliguet. *Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval*. Comput. Vis. Image Underst., vol. 110, no. 3, pages 403–417, June 2008. (Cited on pages 15, 111 and 132.)

[Griffin 2007] G. Griffin, A. Holub and P. Perona. *Caltech-256 Object Category Dataset*. Rapport technique CNS-TR-2007-001, California Institute of Technology, 2007. (Cited on page 30.)

[Hall 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian H. Witten. *The WEKA data mining software: an update*. SIGKDD Explorations, vol. 11, no. 1, pages 10–18, 2009. (Cited on page 81.)

[Hamadi 2013] Abdelkader Hamadi, Georges Quénot and Philippe Mulhem. *Conceptual Feedback for Semantic Multimedia Indexing.* In Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on, Veszprém, Hungary, June 2013. (Cited on pages 13, 106, 115, 130 and 149.)

[Harris 1988] C. Harris and M. Stephens. *A combined corner and edge detector*. 1988. (Cited on page 69.)

[Hérault 2010] Jeanny Hérault. Vision : images, signals and neural networks : models of neural processing in visual perception. Progress in neural processing. World Scientific, New Jersey, London, 2010. (Cited on pages 35, 122, 150 and 153.)

[Ikizler-Cinbis 2010] Nazli Ikizler-Cinbis and Stan Sclaroff. *Object, scene and actions: combining multiple features for human action recognition*. In Proceedings of the 11th European conference on Computer vision: Part I, ECCV'10, pages 494–507, Berlin, Heidelberg, 2010. Springer-Verlag. (Cited on page 71.)

[Itti 1998] L. Itti, C. Koch and E. Niebur. *A Model of Saliency-Based Visual Attention for Rapid Scene Analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pages 1254–1259, Nov 1998. (Cited on pages 36 and 40.)

[Jiang 2012] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu and Chong-Wah Ngo. *Trajectory-Based modeling of human actions with motion reference points*. In Proceedings of the 12th European conference on Computer Vision - Volume Part V, ECCV'12, pages 425–438, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 26 and 71.)

[Jobson 1997] D. J. Jobson, Z. Rahman and G. A. Woodell. *A multiscale retinex for bridging the gap between color images and the human observation of scenes*. Trans. Img. Proc., vol. 6, no. 7, pages 965–976, July 1997. (Cited on page 153.)

[Juan 2009] Luo Juan and Oubong Gwun. *A Comparison of SIFT , PCA-SIFT and SURF*. International Journal of Image Processing IJIP, vol. 3, no. 4, pages 143–152, 2009. (Cited on page 51.)

[Ke 2005] Yan Ke, Rahul Sukthankar and Martial Hebert. *Efficient Visual Event Detection Using Volumetric Features*. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01, ICCV '05, pages 166–173, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on pages 23 and 134.)

[Kläser 2012] Alexander Kläser, Marcin Marszalek, Cordelia Schmid and Andrew Zisserman. *Human focused action localization in video*. In Proceedings of the 11th European conference on Trends and Topics in Computer Vision - Volume Part I, ECCV'10, pages 219–233, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on page 23.)

[Laptev 2003] Ivan Laptev and Tony Lindeberg. *Space-time Interest Points*. In ICCV, pages 432–439, 2003. (Cited on pages 23, 66, 80 and 134.)

[Laptev 2005] Ivan Laptev. *On Space-Time Interest Points*. International Journal of Computer Vision, vol. 64, no. 2-3, pages 107–123, 2005. (Cited on page 111.)

[Laptev 2007] Ivan Laptev, Barbara Caputo and Tony Lindeberg. *Local velocity-adapted motion events for spatio-temporal recognition*. CVIU, pages 207–229, 2007. (Cited on pages 23, 24, 31 and 134.)

[Laptev 2008] Ivan Laptev, Marcin Marszałek, Cordelia Schmid and Benjamin Rozenfeld. *Learning Realistic Human Actions from Movies*. In Conference on Computer Vision & Pattern Recognition, jun 2008. (Cited on pages 23, 24 and 82.)

[Lazebnik 2006] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society. (Cited on page 17.)

[Le Meur 2006a] O. Le Meur, P. Le Callet, D. Barba and D. Thoreau. *A coherent computational approach to model bottom-up visual attention*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 28, no. 5, pages 802–817, 2006. (Cited on page 153.)

[Le Meur 2006b] Olivier Le Meur, Patrick Le Callet, Dominique Barba and Dominique Thoreau. *A Coherent Computational Approach to Model Bottom-Up Visual Attention*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 5, pages 802–817, May 2006. (Cited on pages 36 and 40.)

[Leutenegger 2011] Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart. *BRISK: Binary Robust invariant scalable keypoints*. In Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society. (Cited on pages 18, 21 and 133.)

[Little 2012] Suzanne Little, Iveel Jargalsaikhan, Cem Direkoglu, Noel E. O'Connor, Alan F. Smeaton, Kathy Clawson, Hao Li, Marcos Nieto, Aitor Rodriguez, Pedro Sanchez, Karina Villarroel Peniza, Ana Martínez Llorens, Roberto Giménez, Raul Santos de la Camara and Anna Mereu. *SAVASA Project @ TRECVID 2012: Interactive Surveillance Event Detection*. In TRECVid Workshop, 2012. (Cited on page 26.)

[Liu 2011] L. Liu, L. Wang and X. Liu. *In defense of soft-assignment coding*. In Computer Vision (ICCV), 2011 IEEE International Conference on, page 2486–2493, 2011. (Cited on page 59.)

[Lowe 2004a] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. Int. J. Comput. Vision, vol. 60, no. 2, pages 91–110, November 2004. (Cited on pages 6, 18 and 128.)

[Lowe 2004b] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. Int. J. Comput. Vision, vol. 60, no. 2, pages 91–110, November 2004. (Cited on pages 19, 20 and 133.)

[Mantiuk 2005] Rafal Mantiuk, Scott Daly, Karol Myszkowski and Hans-Peter Seidel. *Predicting Visible Differences in High Dynamic Range Images - Model and its Calibration*. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas and Scott J. Daly, editeurs, Human Vision and Electronic Imaging X, IS&T/SPIE's 17th Annual Symposium on Electronic Imaging (2005), volume 5666, pages 204–214, 2005. (Cited on page 36.)

[Marat 2008] Sophie Marat, Tien Ho Phuoc, Lionel Granjon, Nathalie Guyader, Denis Pellerin and Anne Guérin-Dugué. *Spatio-temporal saliency model to predict eye movements in video free viewing*. In Proceedings of the 16th European Signal Processing Conference, EUSIPCO-2008, pages 1–5, Lausanne, Suisse, June 2008. Département Images et Signal Département Images et Signal. (Cited on page 153.)

[Marszalek 2009] M. Marszalek, I. Laptev and C. Schmid. *Actions in context*. In Proc. Conf. Computer Vision and Pattern Recog. (CVPR'09), pages 2929–2936, Miami Beach, Florida, June 2009. (Cited on pages 30, 31, 32 and 123.)

[Matikainen 2009] Pyry Matikainen, Martial Hebert and Rahul Sukthankar. *Trajectons: Action Recognition Through the Motion Analysis of Tracked Features*. In Workshop on Video-Oriented Object and Event Classification, ICCV 2009, September 2009. (Cited on pages 25 and 69.)

[Moosmann 2006] Franck Moosmann, Diane Larlus and Frédéric Jurie. *Learning saliency maps for object categorization*. In International Workshop on The Representation and Use of Prior Knowledge in Vision (in ECCV '06), Graz, Austria, May 2006. (Cited on page 40.)

[Muja 2009] Marius Muja and David G. Lowe. *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration*. In International Conference on Computer Vision Theory and Application VISSAPP'09), pages 331–340. INSTICC Press, 2009. (Cited on pages 51 and 59.)

[Negrel 2012] R. Negrel, D. Picard and P. Gosselin. *Compact tensor based image representation for similarity search*. In Image Processing (ICIP), 2012 19th IEEE International Conference on, pages 2425–2428, 2012. (Cited on page 111.)

[Ng 2000] Kwong Bor Ng and Paul B. Kantor. *Predicting the Effectiveness of Naive Data Fusion on the Basis of System Characteristics*. Journal of the American Society for Information Science, vol. 51, pages 1177–1189, November 2000. (Cited on pages 27, 99 and 100.)

[Niebles 2008] Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei. *Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words*. Int. J. Comput. Vision, vol. 79, no. 3, pages 299–318, September 2008. (Cited on pages 23 and 134.)

[Nistér 2005] David Nistér. *Preemptive RANSAC for live structure and motion estimation*. Machine Vision and Applications, vol. 16, no. 5, pages 321–329, 2005. (Cited on pages 71 and 73.)

[Nowak 2006] Eric Nowak, Frédéric Jurie and Bill Triggs. *Sampling strategies for bag-of-features image classification*. In Proceedings of the 9th European conference on Computer Vision - Volume Part IV, ECCV'06, pages 490–503, Berlin, Heidelberg, 2006. Springer-Verlag. (Cited on page 18.)

[Ojala 1996] Timo Ojala, Matti Pietikäinen and David Harwood. *A comparative study of texture measures with classification based on featured distributions*. Pattern Recognition, vol. 29, no. 1, pages 51–59, January 1996. (Cited on pages 15 and 132.)

[Ortiz 2012] Raphael Ortiz. *FREAK: Fast Retina Keypoint*. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12, pages 510–517, Washington, DC, USA, 2012. IEEE Computer Society. (Cited on pages 22 and 133.)

[Over 2011] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton and Georges Quénot. *TRECVID 2011 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics*. In Proceedings of TRECVID 2011. NIST, USA, 2011. (Cited on page 48.)

[Over 2012] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Barbara Shaw, Wessel Kraaij, Alan F. Smeaton and Georges Quéenot. *TRECVID 2012 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics*. In Proceedings of TRECVID 2012. NIST, USA, 2012. (Cited on pages 10, 30, 33, 34, 35 and 136.)

[Paris 2010] S. Paris and H. Glotin. *Pyramidal Multi-level Features for the Robot Vision@ICPR 2010 Challenge*. In Pattern Recognition (ICPR), 2010 20th International Conference on, pages 2949–2952, 2010. (Cited on page 111.)

[Redi 2011a] Miriam Redi and Bernard MÃ©rialdo. *Saliency moments for image categorization*. In ICMR 2011, 1st ACM International Conference on Multimedia Retrieval, April 17-20, 2011, Trento, Italy, Trento, ITALY, 04 2011. (Cited on pages 36, 40 and 41.)

[Redi 2011b] Miriam Redi and Bernard Merialdo. *Saliency moments for image categorization*. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11, pages 39:1–39:8, New York, NY, USA, 2011. ACM. (Cited on page 111.)

[Reinhard 2005] Erik Reinhard and Kate Devlin. *Dynamic Range Reduction Inspired by Photoreceptor Physiology*. IEEE Transactions on Visualization and Computer Graphics, vol. 11, pages 13–24, 2005. (Cited on page 36.)

[Rosales 1999] Romer Rosales and Stan Sclaroff. *Trajectory Guided Tracking and Recognition of Actions*. Rapport technique, Boston, MA, USA, 1999. (Cited on page 22.)

[Rosten 2010] Edward Rosten, R. Porter and Tom Drummond. *Faster and Better: A Machine Learning Approach to Corner Detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 1, pages 105–119, 2010. (Cited on page 18.)

[Rublee 2011] Ethan Rublee, Vincent Rabaud, Kurt Konolige and Gary Bradski. *ORB: An efficient alternative to SIFT or SURF*. In Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society. (Cited on pages 21 and 133.)

[Safadi 2010] Bahjat Safadi and Georges Quénot. *Evaluations of multi-learner approaches for concept indexing in video documents*. In Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO '10, pages 88–91, Paris, France, France, 2010. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE. (Cited on pages 13, 112 and 130.)

[Safadi 2011] Bahjat Safadi and Georges Quénot. *Re-ranking for Multimedia Indexing and Retrieval*. In ECIR 2011: 33rd European Conference on Information Retrieval, pages 708–711, Dublin, Ireland, apr 2011. Springer. (Cited on pages 13, 106, 115, 130 and 149.)

[Safadi 2013] Bahjat Safadi and Georges Quénot. *Descriptor Optimization for Multimedia Indexing and Retrieval*. In CBMI 2013, 11th International Workshop on Content-Based Multimedia Indexing, Veszprem, HUNGARY, jun 2013. (Cited on pages 13, 112 and 130.)

[Sánchez 2013] Jorge Sánchez, Florent Perronnin, Thomas Mensink and Jakob Verbeek. *Image Classification with the Fisher Vector: Theory and Practice*. International Journal of Computer Vision, vol. 105, no. 3, pages 222–245, 2013. (Cited on page 112.)

[Savarese 2006] S. Savarese, J. Winn and A. Criminisi. *Discriminative Object Class Models of Appearance and Shape by Correlatons*. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2, CVPR '06, pages 2033–2040, Washington, DC, USA, 2006. IEEE Computer Society. (Cited on page 17.)

[Schapire 1999] Robert E. Schapire and Yoram Singer. *Improved Boosting Algorithms Using Confidence-rated Predictions*. Mach. Learn., vol. 37, no. 3, pages 297–336, December 1999. (Cited on pages 28 and 135.)

[Schuldt 2004] Christian Schuldt, Ivan Laptev and Barbara Caputo. *Recognizing Human Actions: A Local SVM Approach*. In Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on pages 30 and 31.)

[Senane 2001] H. Senane, A. Saadane and D. Barba. *Design and Evaluation of an Entirely Psychovisual-Based Coding Scheme*. Journal of Visual Communication and Image Representation, vol. 12, no. 4, pages 401–421, 2001. (Cited on page 153.)

[Shabou 2012] Aymen Shabou and Hervé Le Borgne. *Locality-constrained and spatially regularized coding for scene categorization.* In CVPR, pages 3618–3625. IEEE, 2012. (Cited on page 111.)

[Shi 1994a] Jianbo Shi and Carlo Tomasi. *Good Features to Track*. In 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), pages 593 – 600, 1994. (Cited on page 18.)

[Shi 1994b] Jianbo Shi and Carlo Tomasi. *Good Features to Track*. pages 593–600, 1994. (Cited on pages 25, 69 and 143.)

[Smeaton 2010] Alan F. Smeaton, Paul Over and Aiden R. Doherty. *Video shot boundary detection: Seven years of TRECVid activity*. Comput. Vis. Image Underst., vol. 114, no. 4, pages 411–418, April 2010. (Cited on page 10.)

[Sprague 1965] James M. Sprague and Thomas H. Meikle Jr. *The role of the superior colliculus in visually guided behavior*. Experimental Neurology, vol. 11, no. 1, pages 115 – 146, 1965. (Cited on pages 36 and 41.)

[Strat 2012a] S.T. Strat, A. Benoit, P. Lambert and A. Caplier. *Retina-enhanced SURF descriptors for semantic concept detection in videos*. In Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on, pages 319–324, 2012. (Cited on pages 48 and 111.)

[Strat 2012b] Tiberius Strat, Alexandre Benoit, Hervé Bredin, Georges Quenot and Patrick Lambert. *Hierarchical Late Fusion for Concept Detection in Videos*. In Rita Cucchiara Andrea Fusiello Vittorio Murino, editeur, Proceedings of Computer Vision - ECCV 2012. Workshops and Demonstrations, Part III, volume 7585 of *Lecture Notes in Computer Science (LNCS)*, pages 335–344, Firenze, Italie, October 2012. Springer Berlin. Oral session 1: WS21 - Workshop on Information Fusion in Computer Vision for Concept Recognition OSEO (French State agency for innovation) and ANR (French national research agency). (Cited on pages 28, 92, 99, 100 and 113.)

[Strat 2013a] Sabin Tiberius Strat, Alexandre Benoit, Patrick Lambert and Alice Caplier. *Retina enhanced SURF descriptors for spatio-temporal concept detection*. Multimedia Tools and Applications, pages 1–27, 2013. (Cited on pages 48, 60 and 111.)

[Strat 2013b] S.T. Strat, A. Benoit and P. Lambert. *Retina enhanced SIFT descriptors for video indexing*. In Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on, pages 201–206, 2013. (Cited on pages 48, 59 and 111.)

[Sudderth 2005] Erik B. Sudderth, Antonio Torralba, William T. Freeman and Alan S. Willsky. *Learning Hierarchical Models of Scenes, Objects, and Parts*. In Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2, ICCV '05, pages 1331–1338, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on page 17.)

[Tanase 2013] Claudiu Tanase and Bernard Merialdo. *Introducing motion information in dense feature classifiers*. In Image Analysis for Multimedia Interactive Services (WIAMIS), 2013 14th International Workshop on, pages 1–4, 2013. (Cited on page 22.)

[Tang 2008] Zhiyuan Tang and Keiji Yanai. *UEC at TRECVID 2008 High Level Feature Task*. In Paul Over, George Awad, R. Travis Rose, Jonathan G. Fiscus, Wessel Kraaij and Alan F. Smeaton, editeurs, TRECVID. National Institute of Standards and Technology (NIST), 2008. (Cited on pages 28, 100 and 103.)

[Tran 2012] K.N. Tran, I.A. Kakadiaris and S.K. Shah. *Part-based motion descriptor image for human action recognition*. Pattern Recognition, vol. 45, no. 7, pages 2562–2572, 2012. (Cited on page 22.)

[Turner 1986] M R Turner. *Texture discrimination by Gabor functions*. Biol. Cybern., vol. 55, no. 2-3, pages 71–82, November 1986. (Cited on pages 15 and 132.)

[Tuytelaars 2008] Tinne Tuytelaars and Krystian Mikolajczyk. Foundations and trends in computer graphics and vision, volume 3, chapitre Local Invariant Feature Detectors: A Survey, page 177–280. 2008. (Cited on page 17.)

[Tuytelaars 2010] Tinne Tuytelaars. *Dense interest points*. In CVPR, pages 2281–2288. IEEE, 2010. (Cited on page 19.)

[Usman Niaz 2011] Usman Niaz, Miriam Redi, Claudiu Tanase, Bernard Merialdo, Giovanna Farinella and Qian Li. *EURECOM at TrecVid 2011: The light semantic indexing task*. In TRECVid'2011, 15th International Workshop on Video Retrieval Evaluation, 2011, National Institute of Standards and Technology, Gaithersburg, USA, Gaithersburg, UNITED STATES, 11 2011. (Cited on pages 40, 41 and 52.)

[van de Sande 2010] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek. *Evaluating Color Descriptors for Object and Scene Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pages 1582–1596, 2010. (Cited on page 111.)

[VanRullen 2002] Rufin VanRullen and Simon J Thorpe. *Surfing a spike wave down the ventral stream*. Vision Research, vol. 42, no. 23, pages 2593–2615, 2002. (Cited on page 153.)

[Vieux 2012] Rémi Vieux, Jenny Benois-Pineau and Jean-Philippe Domenger. *Content based image retrieval using bag-of-regions*. In Proceedings of the 18th international conference on Advances in Multimedia Modeling, MMM'12, pages 507–517, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on page 16.)

[Vig 2012] Eleonora Vig, Michael Dorr and David Cox. *Space-variant Descriptor Sampling for Action Recognition Based on Saliency and Eye Movements*. In Proceedings of the 12th European Conference on Computer Vision - Volume Part VII, ECCV'12, pages 84–97, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 40 and 43.)

[Viola 2004] Paul Viola and Michael J. Jones. *Robust Real-Time Face Detection*. Int. J. Comput. Vision, vol. 57, no. 2, pages 137–154, May 2004. (Cited on page 28.)

[Vrigkas 2013] Michalis Vrigkas, Vasileios Karavasilis, Christophoros Nikou and Ioannis Kakadiaris. *Action recognition by matching clustered trajectories of motion vectors*. In Proc. 8th International Conference on Computer Vision Theory and Application, pages 112–117, Barcelona, Spain, February 2013. (Cited on page 25.)

[Walther] Dirk Walther. *Interactions of Visual Attention and Object Recognition: Computational Modeling, Algorithms, and Psychophysics*. PhD thesis. (Cited on page 153.)

[Wang 1999] R. Wang and T. Huang. *Fast camera motion analysis in MPEG domain*. In Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, volume 3, pages 691–694 vol.3, 1999. (Cited on page 71.)

[Wang 2011] Heng Wang, Alexander Kläser, Cordelia Schmid and Liu Cheng-Lin. *Action Recognition by Dense Trajectories*. In IEEE Conference on Computer Vision & Pattern Recognition, pages 3169–3176, Colorado Springs, United States, June 2011. (Cited on pages 26, 27, 31, 66, 67, 69, 76, 77, 82 and 134.)

[Wu 2003] L. Wu, Y. Guo, X. Qiu, Z. Feng, J. Rong, W. Jin, D. Zhou, R. Wang and M. Jin. *Fudan University at TRECVID 2003*. In Notebook of TRECVID, 2003. (Cited on pages 28, 100, 103 and 104.)

[Wu 2011] Shandong Wu, Omar Oreifej and Mubarak Shah. *Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories*. In Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, pages 1419–1426, Washington, DC, USA, 2011. IEEE Computer Society. (Cited on page 25.)

[Xu 2012] Jingxin Xu, Simon Deman, Sridha Sridharan and Clinton B. Fookes. *SAIVT-QUT@TRECVid 2012 : interactive surveillance event detection*. In Paul Over, editeur, TRECVid 2012 Workshop, NIST, Gaithersburg, USA, November 2012. (Cited on page 26.)

[Yilmaz 2006] Emine Yilmaz and Javed A. Aslam. *Estimating average precision with incomplete and imperfect judgments*. In Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06, pages 102–111, New York, NY, USA, 2006. ACM. (Cited on pages 33, 50, 84, 110, 129 and 142.)

[Yilmaz 2008] Emine Yilmaz, Evangelos Kanoulas and Javed A. Aslam. *A simple and efficient sampling method for estimating AP and NDCG*. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '08, pages 603–610, New York, NY, USA, 2008. ACM. (Cited on pages 33, 50, 51, 56, 84, 110, 129 and 142.)

[Zhang 1999] Tong Zhang and C. Tomasi. *Fast, robust, and consistent camera motion estimation*. In Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., volume 1, pages –170 Vol. 1, 1999. (Cited on page 71.)

[Zhang 2011] L. Bao L. Takahashi S. Li Y. Hauptmann A. Zhang L. Jiang. *Informedia@TRECVID 2011: Surveillance Event Detection*. TRECVID Video Retrieval Evaluation Workshop, Gaitherburg, USA, 2011. (Cited on page 27.)

[Zhang 2013] Longfei Zhang, Ziyu Guan and Alexander Hauptmann. *The Co-attention Model for Tiny Activity Analysis*. Neurocomput., vol. 105, pages 51–60, April 2013. (Cited on page 123.)

[Zhu 2011] Chao Zhu, Charles-Edmond Bichot and Liming Chen. *Color orthogonal local binary patterns combination for image region description*. Rapport technique RR-LIRIS-2011-012, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/Ecole Centrale de Lyon, July 2011. (Cited on pages 15 and 132.)

[Zhu 2013] Chao Zhu, Charles-Edmond Bichot and Liming Chen. *Image region description using orthogonal combination of local binary patterns enhanced with color*

information. Pattern Recogn., vol. 46, no. 7, pages 1949–1963, July 2013. (Cited on page 111.)

Analysis and interpretation of visual scenes through collaborative approaches

During the last years, we have witnessed a great increase in the size of digital video collections. Efficient searching and browsing through such collections requires an indexing according to various meaningful terms, bringing us to the focus of this thesis, the *automatic semantic indexing of videos*.

Within this topic, the Bag of Words (BoW) model, often employing SIFT or SURF features, has shown good performance especially on static images. As our first contribution, we propose to improve the results of SIFT/SURF BoW descriptors on videos by *pre-processing the videos with a model of the human retina*, thereby making these descriptors more robust to video degradations and sensitivite to spatio-temporal information.

Our second contribution is a set of *BoW descriptors based on trajectories*. These give additional motion information, leading to a richer description of the video.

Our third contribution, motivated by the availability of complementary descriptors, is a *late fusion* approach that automatically determines how to combine a large set of descriptors, giving a high increase in the average precision of detected concepts.

All the proposed approaches are validated on the TRECVid challenge datasets which focus on visual concept detection in very large and uncontrolled multimedia content.

Keywords: semantic indexing, video, Bag of Words, SIFT, SURF, retina, spatio-temporal, trajectories, late fusion

Analyse et interprétation de scènes visuelles par approches collaboratives

Résumé : Les dernières années, la taille des collections vidéo a connu une forte augmentation. La recherche et la navigation efficaces dans des telles collections demande une indexation avec des termes pertinents, ce qui nous amène au sujet de cette thèse, *l'indexation sémantique des vidéos*.

Dans ce contexte, le modèle Sac de Mots (BoW), utilisant souvent des caractéristiques SIFT ou SURF, donne de bons résultats sur les images statiques. Notre première contribution est d'améliorer les résultats des descripteurs SIFT/SURF BoW sur les vidéos en *pré-traitant les vidéos avec un modèle de rétine humaine*, ce qui rend les descripteurs SIFT/SURF BoW plus robustes aux dégradations vidéo et qui leurs donne une sensitivité à l'information spatio-temporelle.

Notre deuxième contribution est un ensemble de *descripteurs BoW basés sur les trajectoires*. Ceux-ci apportent une information de mouvement et contribuent vers une description plus riche des vidéos.

Notre troisième contribution, motivée par la disponibilité de descripteurs complémentaires, est une *fusion tardive* qui détermine automatiquement comment combiner un grand ensemble de descripteurs et améliore significativement la précision moyenne des concepts détectés.

Toutes ces approches sont validées sur les bases vidéo du challenge TRECVid, dont le but est la détection de concepts sémantiques visuels dans un contenu multimédia très riche et non contrôlé.

Mots-clés : indexation sémantique, vidéo, Sac de Mots, SIFT, SURF, rétine, spatio-temporel, trajectoires, fusion tardive