ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2014-ENST-0021

EDITE - ED 130

# Doctorat ParisTech

# T H È S E

**pour obtenir le grade de docteur délivré par**

## TELECOM ParisTech

### Spécialité « Signal et Images »

*présentée et soutenue publiquement par*

**Claudiu TĂNASE**

le 24 Avril 2014

# Vers l'analyse spatio-temporelle efficace pour la recherche de vidéo par le contenu

Directeur de thèse : **Bernard MERIALDO**

**Jury**

| | |
|---|---|
| **M. Vincent CHARVILLAT**, Prof., ENSEEIHT, Toulouse – France | Rapporteur |
| **M. Patrick LAMBERT**, Prof., LISTIC, Annecy le Vieux – France | Rapporteur |
| **M. Frédéric PRECIOSO**, Prof., Polytech'Nice-Sophia, Sophia Antipolis – France | Examinateur |
| **M. Stéphane AYACHE**, Prof., Ecole Supérieure d'Ingénieur de Luminy, Marseille – France | Examinateur |

**TELECOM ParisTech**
école de l'Institut Télécom - membre de ParisTech

T H È S E

# Contents

# List of Figures

# List of Tables

# Acronyms

Here are the main acronyms used in this document.

| | |
|---|---|
| BOMT | Bag of Motion Templates |
| BOW | Bag of Words |
| CBIR | Content Based Image Retrieval |
| CBVR | Content Based Video Retrieval |
| CV | Cross Validation |
| DT | Dense Trajectories |
| HOG | Histogram of Oriented Gradients |
| HOF | Histogram of Flow |
| LDA | Linear Discriminant Analysis |
| MAP | Mean Average Precision |
| SIFT | Scale Invariant Feature Transform |
| SIN | Semantic Indexing |
| ST | Spatio-Temporal |
| STIP | Spatio-Temporal Interest Point |
| SVM | Support Vector Machine |

# Chapter 1

# Resumé

## 1.1 Introduction

La quantité de données vidéo numériques disponibles a dépassé notre capacité à les organiser. Depuis le début de la dernière décennie les appareils photo numériques et les téléphones cellulaires capables d'enregistrer des vidéos ont vu leur popularité exploser. Aujourd'hui, l'utilisation du matériel d'enregistrement vidéo tels que les ordinateurs portables, les smartphones et les appareils photo est très répandue. Par conséquent, les consommateurs enregistrent et téléchargent des vidéos en ligne sur des sites de partage vidéo comme YouTube, qui à son tour augmente en taille et en popularité. En 2006, il y avait 6 millions de vidéos sur Youtube, en 2008 il y en avait 60 millions et un an plus tard 120 millions. En 2013, on estime qu'il y a 3 milliards de vidéos, avec 100 heures mises en ligne chaque minute et 6 milliards d'heures de vidéo regardées chaque mois [1]. Cette croissance exponentielle est l'une des nombreuses observations pratiques correspondant à la loi de Moore [1] qui indique que le nombre de transistors dans les circuits intégrés double tous les deux ans. Cela se traduit par de plus en plus de puissance de l'unité centrale disponible pour le même coût. Des observations similaires peuvent être faites sur la capacité du disque dur et des résolutions des capteurs d'images. La combinaison de ces effets a comme résultat d'énormes quantités de données vidéo qui doivent être efficacement comprimées, stock-

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

ées, indexées et recherchées.

Malheureusement, la technologie pour la recherche de vidéo dans de grandes collections n'a pas réussi à suivre le rythme. La taille des nouvelles collections rend le calcul de la recherche intelligente extrêmement intensif. Les solutions pratiques actuelles permettent de récupérer de la vidéo en se basant sur la description texte, des mots-clés ou des méta-données. Les mots-clés sont généralement extraits des sous-titrages, de l'analyse du discours, de l'annotation manuelle et de la reconnaissance optique de caractères. Alors que la recherche à base de texte est très efficace, annoter des vidéos nécessite de main-d'œuvre. Recherches par mots clés sont limitées par des annotations existants et donc on ne peut pas rechercher des événements non marqués. Enfin, l'annotation manuelle ne s'adapte pas à de grandes quantités de données et il est difficile de le faire en temps réel. Beaucoup des systèmes de recherche vidéo ne traitent que des *images clés* dans la vidéo. En particulier, il peut être observé dans TRECVID, une campagne de recherche dédié à l'analyse automatique de contenu vidéo, que l'extraction des descripteurs basés sur une image clé est encore la stratégie dominante. Il est de notre conviction que le contenu vidéo à l'extérieur de la portée de l'image-clé - le contenu *spatio-temporel* - peut être extrait et utilisé avec succès dans la recherche de vidéo en tant que information complémentaire supplémentaire aux descripteurs d'images clés. Notre accent est mis sur l'exploitation du contenu spatio-temporel bas niveau, en développant ainsi des *features spatio-temporelles* afin d'effectuer l'indexation sémantique sur TRECVID SIN.

Le contenu spatio-temporel d'une vidéo est cependant un sujet bien étudié dans les autres communautés scientifiques. Par exemple, la reconnaissance de l'action [2] a longtemps été réalisée en utilisant les fonctions spatio-temporelles. Les quelques caractéristiques spatio-temporelles existantes qui sont actuellement utilisés dans la pratique TRECVID (par exemple STIP) ont d'abord été proposé pour reconnaître des actions humaines. Le problème est que ces features ne sont pas facilement extensible au niveau de la récupération de la vidéo à cause de deux raisons principales: la complexité de calcul et la perte de généralité (en raison de leur application à des concepts généraux). Dans cette thèse, nous proposons de nouvelles fonctionnalités alternatives qui s'adaptent à la détection de concept et fonctionnent bien en fusion avec des features classiques d'images clés.

### 1.1.1 TRECVID SIN

TRECVID [3–5] est une campagne annuelle internationale sponsorisée par NIST[2] qui encourage la recherche dans l'analyse vidéo. Les équipes participantes des chercheurs testent leurs systèmes de recherche de vidéo sur une collection standardisée et comparent les résultats obtenus par des méthodes de notation bien établies. Sur les plusieurs tâches, notre équipe participe à la tâche "Semantic Indexing" (SIN, anciennement "high level feature extraction"), qui est essentiellement une tache de détection des concepts. Pour l'edition 2012 de SIN la collection était composée de 800 heures de vidéo de l'Archive Internet crées par des utilisateurs amateurs sur n'importe quel sujet. La détection automatique des bords de plan relève en total env. 545,923 shots dans 28,123 vidéos. Il y a deux versions de la tache SIN selon le nombre de concepts détectés: la tache "full" fait 346 concepts (dont un échantillon de 80 est évalué) et la tache "light" fait 50 concepts (dont 20 évalués). Les annotations binaires sont disponibles sur les données d'apprentissage et pour l'évaluation des classifieurs. Ces annotations éparses sont crées par un effort collaboratif d'annotation humaine des vidéos. La mesure de performance choisi est une variante échantillonnée de la précision moyenne ("mean average precision" ou MAP) appelée *xinfAP* ("mean extended inferred average precision").

### 1.1.2 Motivation

La particularité de notre recherche est de classifier la vidéo en utilisant dynamique, le mouvement et les descripteurs de contenu spatio-temporelles. En raison des progrès réalisés dans CBIR ou de la complexité de calcul plus élevée des approches spatio-temporelles, l'état de l'art actuel dans l'extraction vidéo se fonde essentiellement sur l'analyse de l'image clé. À quelques exceptions près, les systèmes de détection concept TRECVID sont essentiellement classifieurs d'images qui approximent une vidéo par une ou quelques images.

Les features spatio-temporelles existantes sont lourds en temps de calcul. Puisque TRECVID est un effort de recherche orienté vers les résultats, la faisabilité du calcul d'un descripteur est un facteur important lorsque on se confronte avec des dates limites de compétition. On peut calculer que, avec une vitesse d'extraction de 5 heures pour les 50 minutes du corpus vidéo KTH (ayant une résolution plus petite que les vidéos TRECVID), le temps d'extraction des features sur la collection de 400 heures de TRECVID 2010

[2]National Institute of Standards and Technology
`http://www.nist.gov/`

prendrait plus de 14 semaines de temps CPU. Ce calcul nécessiterait une puissance de calcul et parallélisation au-delà de nos ressources.

### 1.1.3 Énoncé du problème

Compte tenu du coût de calcul relativement léger des descripteurs image et de leur très bonne performance dans la classification d'image, les features spatio-temporelles existantes sont relativement lents, lourds et peu performants dans la détection des concepts.

1. Puisque les features existantes sont soit démesurément lentes ou pas suffisamment performantes, une nouvelle génération de *caractéristiques spatio-temporelles adaptées pour la détection des concepts* est nécessaire.

2. Au lieu de se concentrer uniquement sur la précision dans les plus petits datasets en ignorant totalement le temps processeur, on doit chercher des descripteurs dynamiques qui s'adaptent bien à l'échelle des données TRECVID. Dans le contexte d'une applications du monde réel comme TRECVID, un *compromis entre vitesse et précision* est demandée.

3. Des bonnes caractéristiques spatio-temporelles ne sont pas nécessairement exactes au point de remplacer les approches image-clé, mais sont plutôt *complémentaires* entre eux, de sorte que l'amélioration est obtenue par *fusion*. Nous ne sommes pas à la recherche des caractéristiques spatio-temporelles qui fonctionnent bien par eux mêmes, mais lorsqu'ils sont ajoutés à un pool de fusion apportent une amélioration significative.

### 1.1.4 Contributions de la thèse

En réponse à l'énoncé du problème, nous pouvons résumer les principales contributions présentes dans cette thèse dans les points suivants:

1. Nous expérimentons avec le descripteur HOG3D sur TRECVID et confirmons que la performance est insuffisante pour justifier son utilisation comme une base dynamique standard. Nous arrivons à la même conclusion avec une variante multi-image-clé de SIFT, réitérant ainsi la nécessité de nouvelles fonctions spatio-temporelles. Les solutions simplistes telles que la classification basé sur le type de mouvement caméra prédominant sont également supprimées.

2. Nous proposons un descripteur global qui généralise la fonction d'histogramme des bords MPEG-7 par des moyennes tout simplement et des écarts types de la série de temps résultant de l'extraction de l'histogramme de bords régulièrement à travers le temps. Nous montrons expérimentalement que l'amélioration significative de la MAP peut être obtenue par fusion avec même une feature spatio-temporelle simpliste avec un classificateur SIFT traditionnel.

3. La méthode de "Bags of Motion Templates" s'étend sur l'idée des motion templates de Bradski [6] en extrayant les parties du mouvement, puis de les utiliser comme mots visuels dans un modele Sac-des-mots. Lorsqu'il est appliqué sur la base KTH de reconnaissance d'actions, BOMT obtient environ 88% precision par rapport à 92,1% de STIP mais est calculé quatre fois plus rapide, répondant ainsi au critère de "vitesse vs précision". Encore une fois, l'amélioration par la fusion linéaire de BOMT avec d'autres descripteurs sur les données TRECVID est environ de 5%.

4. L'idée d'enrichir une feature Dense SIFT Sac-de-mots avec des informations de mouvement est présentée. Tout d'abord, une simple séparation entre les parcelles statiques et dynamiques DSIFT menant à deux histogrammes sac de mots est montré pour améliorer les performances sur les concepts contenant mouvement. Apres, une séparation en 4 directions créant des "Z* features" permet la MAP du système de lever de 5,5% par rapport à une base forte de TRECVID avec l'apprentissage SVM correct et la bonne technique de fusion.

## 1.2 Etat de l'art

**L'analyse de contenu** est un terme générique représentant les méthodes d'analyse et de compréhension des collections de médias. Historiquement, l'analyse de contenu a développé des techniques pour décrire texte, des décennies avant le premier moteur de recherche web.

Les progrès en informatique conduisent à l'inclusion ultérieure des nouvelles modalités dans Information Retrieval. L'objectif de **récupération des images basé sur le contenu** (Content based image retrieval - CBIR) consiste à rechercher des images numériques dans les grandes bases de données. CBIR exige que la base de données soit indexée avec des informations provenant du contenu de l'image, par opposition à d'autres formes de récupération de l'image (par exemple, le meta search récupère des images

basées sur les métadonnées). Cette information est généralement sous la forme de couleur, de texture ou de forme descripteurs qui fournissent un bon cadre pour définir une similitude au niveau contenu.

De même, l'idée derrière la récupération de la vidéo, parfois notée **content based video retrieval** (CBVR) consiste à rechercher et indexer une grande collection de vidéo, afin de récupérer une vidéo ou un segment vidéo avec un contenu pertinent à la requête. La campagne TRECVID [3] encourage la recherche sur CBVR et de nombreux chercheurs, compris nous, testent et comparent leurs méthodes sur les données et l'évaluation TRECVID. La piste de SIN de TRECVID est la référence pour tous les descripteurs proposés dans cette thèse. Semantic Indexing dans TRECVID est centrée sur l'apprentissage des classificateurs de séquences vidéo annotées qui peuvent identifier les occurrences des *concepts sémantiques* dans une collection des vidéos test.

### 1.2.1 Pratique courante dans l'indexation sémantique

L'approche standard dans la détection de concept (figure 3.2) est de représenter le document comme un ensemble de features. Les features fournissent une représentation numérique des différentes modalités du contenu des images, tels que la couleur (correlograme de couleur [7]), la distribution des bords ou de la texture (filtres Gabor [8], LBP [9]).

Ces features sont regroupées dans un vecteur feature spécifique au document. Les vecteurs de features sont ensuite utilisées comme entrée pour des classifieurs de concept, qui appliquent des méthodes d'apprentissage automatique pour apprendre à partir des données de entraînement de donner des prévisions sur les données de test. Les prédictions sont ensuite utilisés pour récupérer les vidéos de test qui contiennent probablement le concept appris. A ce stade, généralement la fusion entre les différents classifieurs basées sur des différents descripteurs est effectuée. Les performances d'un tel système sont mesurées en terme de précision moyenne MAP ou courbe de précision-rappel PRC.

Dans les sections suivantes, nous présentons quelques exemples de méthodes populaires utilisés dans l'indexation sémantique correspondant aux différentes étapes visibles dans la figure 3.2

#### Descripteurs couramment utilisés

En théorie les descripteurs sont des représentations du contenu à faible bande passante (compacts, basse dimensionnalité), invariants au variations par-

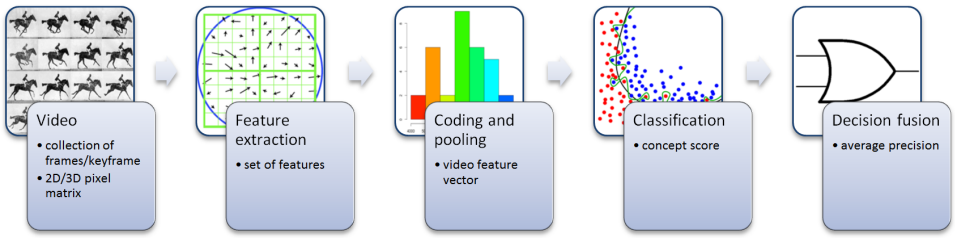


Figure 1.1: Vue d'ensemble d'un système d'indexation sémantique.

asites (par exemple changements de l'éclairage, l'échelle, position dans le cadre, rotation, perspective, contexte etc) et haute discriminabilité (c'est à dire capable de bien distinguer entre des documents ayant un contenu différent).

Un des moyens les plus anciens d'indexer le contenu visuel était basé sur la couleur. Par exemple, dans le système IBM SPEQ [10], une histogramme des couleurs à 64 bins dans l'espace L*a*b, indexé dans un tableau de 4096 couleurs est utilisé pour indexer et récupérer des images dans des collections d'images et de vidéos. Prenant note en outre l'idée d'index de couleur, la corrélogramme de couleurs [7] est un tableau de paires de couleurs indexées où la $d$-ème entrée de ligne $(i, j)$ spécifie la probabilité de trouver un pixel de couleur $j$ à une distance de $d$ pixels d'un pixel de couleur $i$ dans l'image.

L'utilisation de Mel-Frequency Cepstral Coefficients (MFCC [11]) est le fondement du traitement de la parole moderne. La forme vectorisée du spectre MFC est utilisé par certains systèmes [12,13] pour indexer le contenu de du canal audio d'une vidéo. Dans cette thèse, nous ne nous concentrons pas sur les caractéristiques audio.

Les textures ont prouvé expérimentalement au cours des années d'être un moyen efficace pour indexer les images, principalement en raison de la grande variabilité des textures par rapport aux autres formes de contenu visuel. Quelques représentations globales de texture sont devenus populaires en raison de leur coût de calcul bas. Un ensemble de filtres de Gabor sur différentes fréquences et orientations peut effectivement décrire les distributions des bords dans une image [8]. La transformée en ondelettes discrète (DWT) représente une image comme une somme de fonctions ondelettes à différents endroits et échelles. Des coefficients d'un banc de filtres =ondelettes sont utiles dans la compression d'image et peuvent être utilisés en tant que caractéristiques pour l'indexation. Les Local Binary Patterns [9] sont des opérateurs de texture très efficaces qui décrivent la texture locale

par seuillage rapide du voisinage du pixel. Améliorations sur l'idée originale comme le Color Orthogonal LBP [14] ont été testés avec succès dans TRECVID [12].

### Descripteurs locaux d'image

Un sous-ensemble important des descripteurs est représenté par des features 2D locales. Ces approches sont composés de deux phases: la phase de détection indique où dans l'image doivent être extraites des feautes locaux (détection des points d'intérêt) et le descripteur définit comment représenter le voisinage autour du point d'intérêt.

Voici quelques exemples de détecteurs pertinents pour notre travail.

- Le détecteur de coins Harris [15] est un détecteur invariant au rotation. Améliorations plus récentes [16] ont ajouté la détection automatique de l'échelle (Harris-Laplace) et l'invariance à déformation affine (Harris-Affine).

- La matrice 2 par 2 de Hesse est obtenue à partir des premiers termes de développement de la série Taylor de l'intensité de l'image. Basé sur des maxima locaux de la trace de cette matrice i.e. le Laplacien $\nabla$, des structures en forme de goutte peuvent être détectés. Analogue au détecteur de coin Harris il y a des détecteurs Hesse-Laplace et de Hesse-Affine qui évaluent l'échelle automatiquement et sont invariants au transformations affines, respectivement. Détecteurs Hessiens approximés par différences de gaussiens sont couramment utilisés comme faisant partie des détecteurs de points-clés multi-échelle pour SIFT [17].

- Les Régions extrémaux maximum stables (MSER [18]) sont les composantes connexes qui définissent une région constamment plus clair/plus sombre que les pixels de sa limite extérieure.

- L'extraction dense n'utilise aucune méthode de détection. Les features denses sont simplement extraits dans une grille régulière à partir de l'image source ou de la vidéo. Dans certains scénarios l'extraction dense a prouvé à surpasser la détection basé sur points d'intérêt à la fois dans la classification d'images [19] et de la reconnaissance de l'action [20]. SIFT dense est une caractéristique régulière trouvé dans de nombreux systèmes de TRECVID [12, 13]

Un descripteur local est calculé dans le voisinage du point d'intérêt et doit décrire le contenu visuel de la pièce le plus de l'invariabilité au perturbations possible. Nous présentons maintenant quelques descripteurs notables:

- SIFT [17] est le plus couramment utilisé descripteur dans la classification des images. La version originale proposée par Lowe est un histogramme dépendante de la position des directions de gradient local extrait d'une grille autour du point d'intérêt. Il existe de nombreuses extensions de SIFT qui prennent en considération des informations de couleur: OpponentSIFT [21] est un exemple qui a été utilisé dans l'indexation [12]. Dans OpponentSIFT, un descripteur TIFS standard de dimension 128 est calculé sur chacun des trois canaux de couleur opposants et les trois vecteurs sont concaténés

- Le descripteur SURF [22] part quelques similitudes avec SIFT, la différence étant dans la mesure qui est quantifié sur la grille: au lieu du gradient, SURF quantifie les réponses horizontales et verticales aux ondelettes Haar.

- Les Histogrammes de Gradient Orienté (HOG [23]) ont été d'abord proposé pour reconnaître les piétons dans des images statiques. HOG est un descripteur très général et configurable. La fonction d'histogramme de bord de MPEG-7 [24, 25] peut être considéré comme un descripteur HOG avec $4 \times 4$ cellules et 5 bacs par cellule. En outre, certains chercheurs considèrent SIFT à être un descripteur de type HOG.

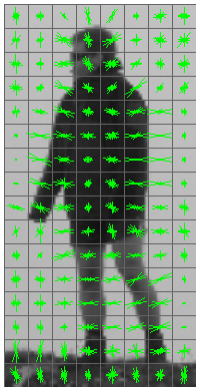

Figure 1.2: HOG superposé sur un image d'une personne. Source:`http://www.juergenwiki.de/work/wiki/doku.php?id=public%3ahog_descriptor_computation_and_visualization`

- Bien que LBP a d'abord été proposé comme un descripteur de texture (global), il a été utilisé [26] avec extraction dense comme un descripteur

local dans l'indexation vidéo.

### Codage et "pooling"

Dans l'indexation les représentations intermédiaires collectent des features bas niveau à partir du document et calculent une forme du document en les agrégant dans un vecteur discriminant qui peut être en suite classifié.

Un Sac de Mots (Bag of Words) est une histogramme des comptes d'occurrences d'un vocabulaire de features locales de l'image. Dans la vision par ordinateur et CBIR, le modèle est connu sous le nom de Sac de Mots Visuels (Bag of Visual Words), et est une technique très courante pour représenter des descripteurs sans modéliser la structure géométrique.

Une mise à jour notamment populaire sur BOW est l'inclusion des pyramides spatiales par Lazebnik et. al. [27], qui apporte une mesure de la structure spatiale à BOW en divisant progressivement l'image en sous-images. La performance est plus élevé, mais en revanche ça augmente de façon exponentielle la taille de la représentation.

Plus récemment, des vecteurs Fisher [28] ont gagné en popularité dans TRECVID. Cette technique construit le vocabulaire visuel où chaque mot visuel est une distribution gaussienne à la place d'un centroïde du cluster. Chaque échantillon est affecté doux à toutes les gaussiennes, ce qui conduit à un vecteur de caractéristiques dense (plutôt que de rares pour BOW). La dimension de la représentation est $k * N$ qui rend cette approche difficile à l'échelle de TRECVID.

Une autre évolution de BOW vue dans TRECVID est super-vecteurs [29]. Cette approche généralise l'attribution douce à l'ensemble du dictionnaire et élargit la représentation vectorielle à la dimension $k * (N + 1)$, avec un poids constant et $N$ distances pour chaque mot visuel.

Testé par [30], ces méthodes ont montré d'améliorer la précision de la classification d'image au coût de dimensionalités plus élevés. Leur application dans TRECVID est en train de devenir courante, mais la complexité de temps est un obstacle reconnu.

### Classification

Dans le cadre de ce travail, la classification concept est un problème d'apprentissage supervisé, dans lequel pour chaque concept il ya des étiquettes d'apprentissage pour une partie des vidéos d'entraînement. Les étiquettes sont binaires et dans la pratique sont faites par annotation humaine.

### Machines à vecteurs supports

SVMs, tel que proposé initialement par Vapnik et. al. [31] sont des classificateurs supervisés qui trouvent un hyperplan de séparation entre les échantillons positifs et négatifs et maximisent la marge entre l'hyperplan et les caractéristiques. L'apprentissage SVM est le standard *de facto* de classification d'image à grande échelle, ce qui a porté sur plus de la récupération de la vidéo dans TRECVID. Au lieu d'utiliser le produit scalaire comme mesure de la distance entre les échantillons, les techniques ultérieures ont adopté à l'utilisation de noyaux non-linéaires comme Gaussian RBF, $\chi^2$ exponentiel et le noyau intersection histogramme.

### Fusion

La combinaison des classificateurs est une composante importante d'un système d'indexation multi-caractéristiques. Une bonne stratégie de fusion peut exploiter la complémentarité des features. La fusion de plusieurs classificateurs peut être effectuée en plusieurs endroits sur le pipeline de de classification. La *fusion précoce* ou «fusion niveau caractéristiques» est effectuée après la phase d'extraction et avant la classification. En général, il s'agit de la concaténation des vecteurs de caractéristiques dans un grand vecteur unique. La *fusion tardive*, ou «fusion décisionnelle» ou «fusion des scores» utilise les décisions de classificateurs individuels pour donner une décision définitive pour un échantillon de test.

Concaténation des vecteurs de caractéristiques de différentes modalités dans un supervecteur est une stratégie de fusion simple qui a été utilisé dans la combinaison des caractéristiques audio et vidéo [32]. Snoek et. al. [33] a montré que la fusion précoce fonctionne mieux que la fusion tardive dans 6 concepts sur 20.

La fusion effectuée après classification est appelé fusion tardive. Pour chaque modalité un classifieur distinct est formé. Les décisions des classificateurs (scores/ valeurs de confiance/probabilités) sont utilisés pour donner une décision finale. Ce type de fusion est beaucoup plus populaire auprès des participants TRECVID.

## 1.2.2 Reconnaissance d'actions basé sur le spatio-temporel

La plupart des études sur la description de la vidéo spatio-temporelle a été initialement conçu pour reconnaître les mouvements humains (pour une étude complète sur la reconnaissance de l'action voir [2] et [34]). Analogue avec les features 2D, les features spatio-temporelles sont soit globales:

MHI [35], champs de flot optique [36], ou des templates spatio-temporelles d'action [37–41] , soit locales: STIP [42], cuboids [43], trajectoires denses [20], HOG3D [44] ou des extensions 3D de SIFT [45, 46]

### 1.2.3 Caractéristiques spatio-temporelles dans la détection de concepts à grande échelle

Dans les années 1990 et au début des années 2000, parallèlement à l'augmentation des vidéos Internet, un grand nombre de techniques de recherche vidéo ont été proposées. Un article de l'enquête par Ren et al. [47] tente d'organiser et classer un grand nombre de ces techniques. Bien qu'elles sont intéressantes, la plupart de ces approches n'ont pas continués à susciter l'intérêt de la recherche plus tard. À quelques exceptions près (la plupart présentés ci-dessous), les caractéristiques spatio-temporelles ont été très rarement mis en œuvre dans les systèmes pratiques de détection de concepts et presque jamais sur TRECVID. Une des causes est évidemment le calcul lourd. Au lieu de cela, la tendance dans TRECVID SIN [3–5] a été l'utilisation des plusieurs images clés par plan, à partir desquelles les descripteurs d'images sont extraits. Cependant, les approches hybrides sont parfois réussies: MediaMill échantillonne plusieurs images-clés par plan et obtient une performance supérieure. Cependant, combien est due à la stratégie multi-images-clés est inconnu.

En 2012, on voit dans TRECVID SIN quelques exemples de descripteurs dynamiques:

- Université de Kobe réalise les meilleures performances dans la tâche «Light»; leur run comprend les trajectoires denses [48] de Wang utilisés pour définir un déplacement de trajectoire en 30 dimensions et un descripteur HOG autour de la trajectoire. La représentation se fait d'abord en réduisant la dimensionnalité par PCA, puis codage par supervecteurs GMM. L'apprentissage se fait par un SVM noyau RBF et la fusion est obtenue par combinaison linéaire pondérée.

- Informedia extrait MoSIFT [46], ainsi que SIFT classique et couleur SIFT.

- IRIM [12] et Quaero extraient des STIP et une nouvelle feature apellée 'faceTracks'

- UEC emploie un descripteur spatio-temporelle dérivé de SURF [49]

- Florida International University et Université de Miami [50] utilisent le HOOF [51]
- L'équipe GIM [52] utilise leur propre feature de mouvement basé sur la direction du mouvement dans 5 régions de l'image (4 coins plus centre).
- Des features 2D sont extraites de plusieurs images par plan par IBM [53] (à 0.25fps), PicSOM et MediaMill. Egalement, ITI-CERTH extraite des features 2D des tranches du volume spatio-temporel apellées des "tomographes".

Pour résumer, sur les 25 équipes participantes en 2012 SIN, seulement 7 utilisent des vraies caractéristiques spatio-temporelles, par rapport à au moins 23 pour SIFT.

## 1.3 Approches globales

Les approches globales dans le Image Retrieval sont à la fois plus légers en calcul et moins précises que les features locales. Dans certains cas, comme pour les histogrammes des bords [24] définis dans les norme MPEG-7, le compromis vitesse / précision semble avantageux, en particulier pour les grands ensembles de données [25]. Par rapport aux approches de niveau local, on trouve très peu d'exemples d'extensions spatio-temporelles globales [54–56] avec des preuves expérimentales très limité. La vitesse précitée et l'absence de caractéristiques globales en 3D constituent la motivation de notre approche dans ce chapitre.

### 1.3.1 HOG3D comme baseline

Il n'existe aucune méthode de description du contenu spatiotemporel reconnue par la communauté comme référence. Fait intéressant, dans CBIR une telle référence existe certainement: la combinaison d'un descripteur d'image local et un modèle de représentation intermédiaire. Dans la pratique, c'est la plupart du temps SIFT et BOW.

Le descripteur HOG est un descripteur visuel similaire au SIFT qui a été appliquée avec succès à la reconnaissance de l'action humaine [23]. En suivant la tendance générale de transformer un descripteur d'images (2D) en descripteur ST, Kläser, Marszalek et Schmid [44] ont développé une version 3D du descripteur HOG. Ce descripteur local représente la région autour du point d'intérêt par un vecteur de caractéristiques, alors que l'ensemble du

volume d'analyse vidéo est représenté comme un ensemble de caractéristiques calculées à des échelles et positions différentes.

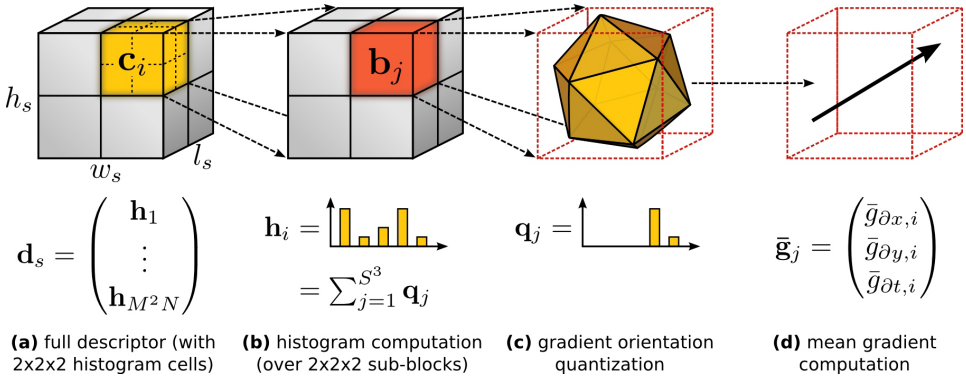


Figure 1.3: Vue d'ensemble du descripteur HOG3D [44] Source: `http://lear.inrialpes.fr/people/klaeser/research_hog3d`

La détection des points d'intérêt sur l'ensemble du plan est un processus coûteux en calcul (voir les temps de calcul pour STIP dans la figure **??**). Pour cette raison, nous avons décidé de convertir des points d'intérêt 2D des images clés de SIFT déjà existants à des points d'intérêt spatiotemporels. Ces points ont déjà été calculé pour la classification SIFT sur l'image-clé centrale du plan (keyframe). Nous avons utilisé le toolkit populaire Lipvireo [57] avec un détecteur Hesse-Laplace pour trouver $<$500 points-clés dans chaque image-clé. Le vecteur de features donné par HOG est une concaténation à niveau de sous-blocs spatiotemporels, entraînant une dimension de $4 * 4 * 3 * 20 = 960$. Le code pour le descripteur HOG3D est disponible sur le site de l'équipe LEAR [44]. Pour obtenir notre vecteur BOW, nous prenons un sous-échantillon aléatoire de 50,000 vecteurs de caractéristiques HOG3D et nous effectuons du regroupement k-means avec $k = 500$ clusters. Le vecteur des occurrences de taille 500 est en suite utilisé comme vecteur caractéristique de plan pour un classifieur SVM. Nous expérimentons avec HOG3D sur les données TRECVID 2007 SIN. Cette collection contient 36,262 plans obtenus à partir d'environ 100 heures de vidéo de magazines d'information, des reportages, des documentaires, des programmes éducatifs et des vidéos d'archives en format MPEG-1. Il y a 32 concepts à évaluer. Finalement, on calcule le gain de précision obtenu par fusion linéaire entre HOG3D et un classifieur SIFT. Le resultat de cette fusion est présenté dans la figure 4.2. Notez la dernière colonne qui représente le MAP de la fusion:

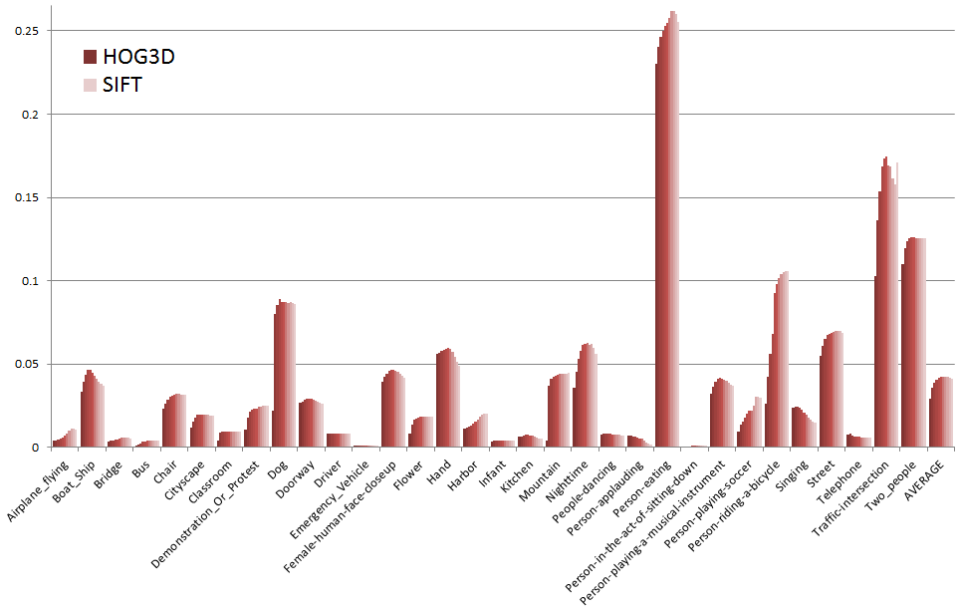


Figure 1.4: Fusion tardive entre HOG3D et SIFT

le MAP maximal du système est approximativement à la même hauteur que le classifieur SIFT.

### 1.3.2 SIFT multi-keyframe

Pour bien paramétrer leurs systèmes des détection concept, quelques [58] participants TRECVID traitent plusieurs cadres par plan. On essaye d'implémenter cette technique en utilisant un échantillonnage temporel à taux de 1/5. Une stratégie de pooling est aussi nécessaire pour estimer le score du plan à partir des $N$ scores des cadres-clés du plan. La moyenne et le maximum sont des candidats évidents pour le pooling, et on expérimente avec une troisième stratégie dont on calcule le score SVM de la moyenne les histogrammes BOW. En utilisant des classifieurs entraînes sur des features SIFT sur TRECVID 2010a (une moitié de l'ensemble de développement 2010) avec BOW de taille 500, on réplique le même processus sur l'ensemble de test avec une seule différence: en lieu d'extraire un seul cadre par plan on échantillonne un grand nombre N de cadres du plan (1/5 cadres), on calcule les caractéristiques SIFT et on crée une seule histogramme $h_{kf}$ par keyframe $kf$. Ainsi on obtient $N$ histogrammes de taille 500 qui peuvent être traitées par le classifieur pour

obtenir $N$ scores $score_{SVM}$.

Les résultats de mkfSIFT par rapport au baseline SIFT sont présentées dans le tableau suivant:

Table 1.1: Précision moyenne du mkfSIFT sur l'ensemble TRECVID 2010b

| concept | mkfSIFT1 max-pool | mkSIFT2 avg-pool | mkSIFT3 avg-bow | SIFT |
|---|---|---|---|---|
| Airplane_Flying | 0.006 | **0.025** | 0.018 | 0.017 |
| Boat_Ship | 0.019 | 0.018 | **0.024** | 0.012 |
| Bus | 0.003 | **0.006** | 0.005 | 0.005 |
| Cityscape | **0.178** | 0.154 | 0.137 | 0.168 |
| Classroom | 0.008 | 0.005 | **0.009** | 0.007 |
| Demonstration_Or_Protest | 0.038 | 0.033 | **0.060** | 0.044 |
| Hand | 0.003 | 0.008 | 0.006 | **0.025** |
| Nighttime | 0.028 | **0.063** | 0.040 | 0.059 |
| Singing | 0.060 | 0.076 | 0.072 | **0.140** |
| Telephones | **0.010** | 0.003 | 0.003 | 0.005 |
| MAP | 0.035 | 0.039 | 0.037 | **0.048** |

Selon les résultats il est un peu surprenant de découvrir que malgré avoir plus d'information dans le multi-keyframe, le MAP est pire que dans le SIFT traditionnel. Une cause possible est l'effet de bords: moyenner beaucoup des scores pour un plan diminue l'influence des grands scores qui correspondent au cadres relevants.

### 1.3.3 ST-MP7EH

Dans ce chapitre on propose un descripteur dynamique capable de contourner les problèmes des approches spatio-temporelles classiques: bruit éccessif, mouvement caméra intense, mauvaise segmentation en plans. Notre descripteur ST-MP7EH est basé sur le feature "histogramme des bords" (Edge Histogram), qui fait partie du standard MPEG-7 [24] et marche essentiellement en analysant l'évolution temporelle des bords dans le plan. En souséchantillonnant des cadres jusqu'à un taux raisonnablement bas nous pouvons réduire le temps de calcul, ce qui est essentiel vu nos jeux de données.

ST-MP7EH à été testé sur les données TRECVID, en ayant comme but son intégration dans un mechanisme de fusion tardive, et nos expériences dans section 4.3.4 essayent de le montrer. Les résultats montrent le fait que le statistiques temporelles améliorent le MAP et que la fusion avec autres

descripteurs augmente la performance. ST-MP7EH a fait l'objet d'une publication [59].

La figure 4.6 montre le résultat des différents mélanges entre ST-MP7EH et SIFT. Le gain de précision moyen du à la fusion tardive est de 22%, correspondant à un MAP de 0.0587.

### 1.3.4 Etude sur les mouvements caméra

Dans cette section on présente un étude sur la corrélation entre les différents types de ego-mouvement (mouvement de l'observateur) dans une vidéo et les concepts sémantiques présents dedans. En utilisant la structure de stabilisation vidéo présente dans tous nos expériences, on peut déterminer pour toute transition la transformation caméra affine. Selon [60], on peut exprimer les 6 paramétrés en terme de divergence, torsion et deux termes hyperboliques, qui après nous permettent de catégoriser le mouvement en 4 catégories principales: $P$ – "panning/tilt/traveling", $R$ – rotation autour de l'axe optique, $Z$ – zoom ou déplacement avant/arrière et $C$ - mouvement non-modelisable par homographie (e.g. parallaxe causé par le mouvement latéral). En calculant la présence de chaque catégorie de mouvement sur chaque plan, on peut calculer la corrélation mouvement/concept, montrée ici en figure 4.6. Malheureusement, aucun concept est même pas faiblement corrélé avec un type de mouvement caméra.

## 1.4 Méthodes BOW inspirées par la détection des actions

Lors de l'étude de la littérature sur la détection de l'action humaine on observe des points communs avec la détection de concepts. Dans les deux cas on extrait de l'information de mouvement et on détecte des motifs fortement variables dans des vidéos peu contraintes. Des amples revues de l'état de l'art [2] montre qu'il y a beaucoup plus intérêt dans la détection actions que classification concept. Par conséquence on trouve une gamme plus variée de collections des actions humaines: KTH, Weizmann, Youtube, Hollywood2. Ces datasets varient en taille, qualité et complexité de l'action mais elles sont tous plus petites que TRECVID. Notre contribution dans ce chapitre est un nouvel descripteur spatio-temporel appelé BOMT qui est basée sur la technique de Motion History Image et qui est adapté aux expériences de grande-échelle de TRECVID.

Le Motion History Image [35] est une représentation 2D du mouvement

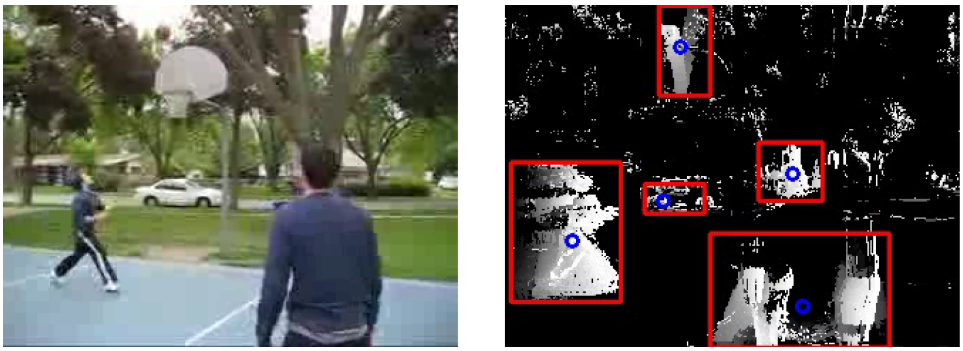

Figure 1.5: Exemple de Motion Templates extraites d'une séquence du dataset Youtube Action.

récent (voir figure 5.1) qui à chaque moment est calculé en mettant à jour les pixels en mouvement d'intérêt (en forme de masque $\Psi(x, y, t)$) et faner les autres pixels par un taux paramètre $\delta$. Le MHI est défini dans [61] comme:

$$H(x,y,t) = \begin{cases} 1 & \text{if } \Psi(x,y,t) = 1 \\ max(0, H(x,y,t-1) - \delta) & \text{otherwise} \end{cases} \tag{1.1}$$

### 1.4.1 Motion Templates

On suive la technique décrite dans [6] et implémenté dans OpenCV [62] pour la segmentation du MHI dans des régions de mouvement. La méthode calcule l'uniformité locale du gradient du MHI. Le processus filtre la plupart du bruit provenant de la masque de mouvement. Les régions restantes ont un gradient uniforme qui est indicatif de la direction et la vitesse du mouvement. Les frontières de ces régions appelées désormais des Motion Templates, sont facilement obtenues par segmentation en composantes connexes.

Pour représenter les MT dans l'espace des features on se sert d'un descripteur similaire à HOG [23]. D'abord on redimensionne le MT jusqu'à une taille carrée de $256 \times 256$. On divise le MT en 16 régions sur une grille $4 \times 4$ (voir figure 5.3) et dans chaque bloc on quantifie les orientations du gradient dans 8 bacs directionnels + 1 pour les gradients petits ou zéro. Le vecteur résultant à une taille de 144.

Le processus d'extraction est répété pour toutes les cadres du plan et après accumulées dans un "sac de mots" (d'où le nom BOMT - Bag of Motion Templates). Le dictionnaire est construit par le clustering k-means d'un

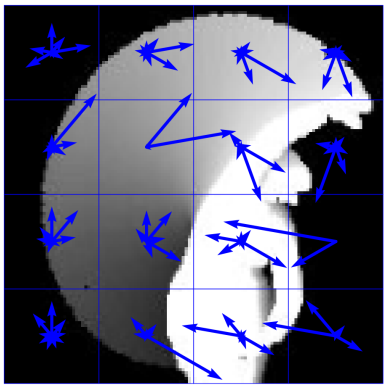

Figure 1.6: Un Motion Template avec ces histogrammes orientation

échantillon de 100,000 vecteurs BOMT. On peut observer dans figure 5.4 les différentes parts d'action correspondant au codewords BOMT, par exemple pour l'action "handwaving" sur le deuxième rang, quatrième colonne.

On apprend les concepts/actions avec des classifieurs SVM. Pour chaque action on entraîne une classifieur SVM binaire un-versus-reste utilisant le noyau $\chi^2$ exponentiel. Les paramètres d'apprentissage et de noyau sont optimisés avec la validation croisée. On estime les probabilités de chaque classe avec la formule de Platt implémenté dans LibSVM [63]. Dans le cas de classification multi-classe, on choisi la classe avec la plus probable.

### 1.4.2 Expériences

On a testé BOMT sur les 4 grands datasets de la reconnaissance action disponible: KTH, Youtube, UCF Actions, Hollywood2 et sur TRECVID 2010. Les caractéristiques et les formules d'évaluation pour ces collections sont présentés dans la section 3.2.3. Pour le paramétrage on a empiriquement choisi une valeur de $\delta = 1/30$ pour le "temps de vie" de la MHI, ce qui veut dire qu'un pixel mis à jour se estompe complètement en 30 cadres. La taille du dictionnaire est fixée à $k = 500$ sur KTH, UCF et Hollywood2 et $k = 5000$ pour TRECVID; sur Youtube on a fait 2 expériences pour les deux valeurs. Les résultats sont présentes dans les tableaux suivants.

Les résultats montre que BOMT ne dépasse pas l'état de l'art, mais que pour des datasets contenant des videos web (Youtube et TRECVID) la performance est comparable avec STIP. Ceci est un fait intéressant car l'extraction de BOMT est quelques fois plus rapide que STIP. Performance

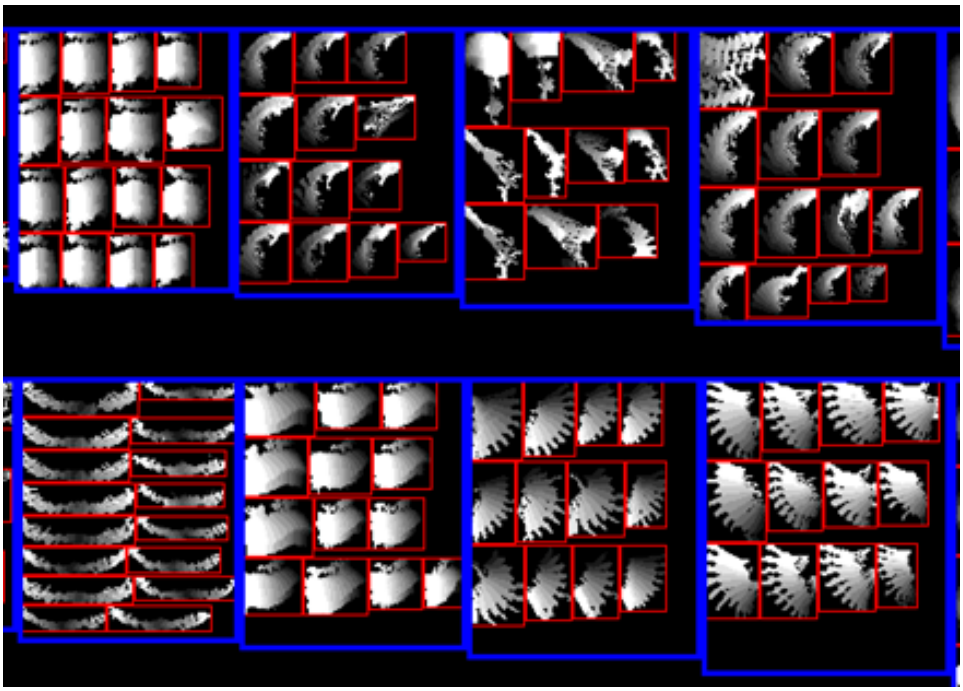

Figure 1.7: Détail d'un mosaïque des Motion Templates (bords rouges) groupés dans des codewords (bords bleus) extraites de KTH

| dataset<br>classes<br>metric | KTH<br>5 actions<br>avg. acc. | Youtube<br>11 actions<br>avg. acc. | UCF<br>10 actions<br>avg. acc. | Hollywood2<br>12 actions<br>mAP | TV2010<br>50 concepts<br>mAP |
|---|---|---|---|---|---|
| BOMT | 88.97% | 64.36% | 56.92% | 19.96% | 1.51% |
| STIP [20] | 92.1% | 59.1% [64] | 82.6% | 47.4% | 0.99%[1] |
| DT [48] | 94.2% | 84.2% | 88.2% | 58.3% | |
| MoSIFT [46] | 95.8% | 63.1% | | | 2.27% |
| chance | 20.0% | 9.09% | 10.0% | 9.63% | 1.03% |

Table 1.2: Comparaison de performance entre BOMT et l'état de l'art sur 5 datasets.

[1] résultat non-officiel obtenu directement d'un document interne d'IRIM [12]

| | boxing | handclapping | handwaving | jogging | running | walking |
|---|---|---|---|---|---|---|
| boxing | 97 | 1 | 0 | 1 | 0 | 1 |
| handclapping | 1 | 97 | 1 | 0 | 0 | 0 |
| handwaving | 0 | 5 | 94 | 0 | 1 | 0 |
| jogging | 1 | 0 | 0 | 79 | 13 | 7 |
| running | 3 | 0 | 0 | 23 | 74 | 0 |
| walking | 1 | 0 | 0 | 6 | 1 | 92 |

Figure 1.8: Matrice de confusion de BOMT sur KTH

sur UCF et Hollywood2 est faible probablement à cause de la variabilité des MT causé par une meilleure résolution vidéo corroboré avec le petit nombre de vidéos dans la collection.

Pour mieux illustrer le temps de calcul, on compare le temps CPU nécessaire pour l'extraction dans KTH (500 séquences, temps total de rendu vidéo 50 minutes à 25 fps) pour notre descripteur BOMT et STIP [42], Dense Trajectories [48] et MoSIFT [46].

| descripteur | time | fps | KTH acc. |
|---|---|---|---|
| BOMT | $1h08m55s$ | 18.12 | 88.97% |
| STIP | $5h11m45s$ | 4.01 | 92.10% |
| Dense Trajectories | $5h54m20s$ | 3.52 | 94.20% |
| MoSIFT | $6h45m50s$ | 3.08 | 95.00% |
| temps réel | $49m57s$ | 25.00 | |

Table 1.3: Comparaison du temps d'extraction des features pour les séquences KTH.

On peut observer dans le tableau 5.2 que BOMT fait à peu près 10% moins de précision mais il est calculé 4.5 fois plus vite.

Dans cette expérience on applique la fusion tardive sur TRECVID SIN 2010 entre BOMT, Dense SIFT et MoSIFT. A cause de la manque des données de validation sur l'ensemble test, la fusion dans cette expérience est de type optimiste (voir annexe B pour des détails). Le résultat est présenté dans le tableau.

Encore les résultats confirment l'amélioration obtenue par fusion des caractéristiques complémentaires. Si on considère DSIFT comme baseline,

| DSIFT | BOMT | MoSIFT | fusion | gain |
|---|---|---|---|---|
| | 0.0151 | 0.0227 | **0.0234** | 3.029% |
| 0.0862 | | 0.0227 | **0.0897** | 3.974% |
| 0.0862 | 0.0151 | | **0.0848** | 1.911% |
| 0.0862 | 0.0151 | 0.0227 | **0.0910** | 5.545% |

Table 1.4: Fusion linéaire de DSIFT, BOMT et MoSIFT pour des données TRECVID 2010

BOMT n'améliore que de environ 2%, MoSIFT de 4% mais les deux ensemble de 5.5%. Ce résultat confirme l'intuition qu'il existe de l'information redondante entre BOMT et MoSIFT.

## 1.5 Enrichir DSIFT-BOW avec mouvement

On a précédemment vu que la détection concept dans les grandes collection est habituellement faite par l'analyse des cadres-clés (keyframe). Un choix populaire pour la représentation du contenu visuel d'une keyframe est basée sur le pooling par bag of words des descripteurs locaux comme Dense SIFT [17]. Par ailleurs, des caractéristiques simples comme le flou optique peuvent être facilement extraites dans le positions correspondants au points DSIFT. Dans ce chapitre on propose des méthodes d'intégration des informations de mouvement dans DSIFT-BOW.

Dans un premier temps, nous voudrons exploiter l'information locale sur la présence/absence du mouvement. Le

Le cœur de notre approche est la séparation entre les patches statiques et dynamiques. Nous faisons cette séparation basée sur la valeur correspondante de flux optique compensée, que nous obtenons plus tôt pour le masque de mouvement. Figure 6.2 montre comment cette séparation influence le feature. Nous avons expérimenté avec différentes variantes de séparation.

Les trois stratégies de séparation qu'on essaie sont les suivantes:

- Dans un premier temps, on considère comme dynamiques les DSIFT qui correspond à un flou optique supérieur à un seuil $\theta_1$ (choisi comme la médiane de toutes les magnitudes des flous optiques extraits), et statiques les autres.

- La seconde stratégie utilise la même règle mais avec un seuil minimal. On baisse le seuil pour permettre à plus des patches à être considérés

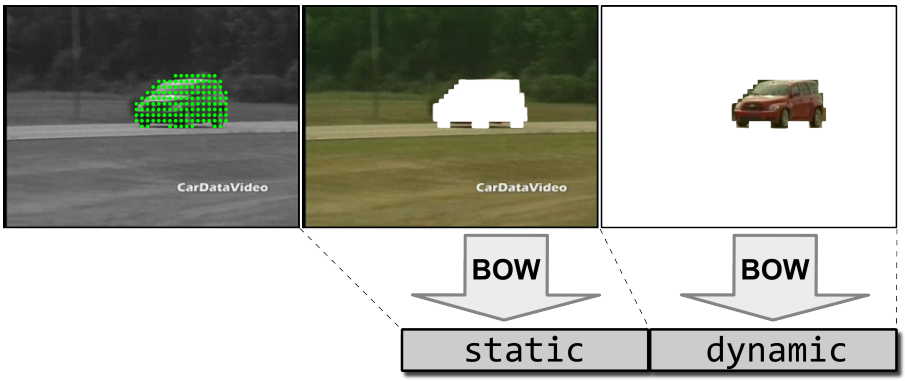


Figure 1.9: Séparation entre les DSIFTs statiques et dynamiques, suivie par le modèle BOW appliqué séparément sur chaque des 2 ensembles de patches résultants. Pour le feature final les vecteurs statique et dynamique sont concaténées.

comme dynamiques. Ce choix est motivé par la dominance des features statique résultants de la première stratégie.

- La troisième stratégie consiste affectation douce. Utilisation d'une valeur de seuil fixe signifie que les patches avec une vitesse proche du niveau de seuil peuvent tomber soit sur la partie statique ou dynamique, ce qui crée du bruit. Cela peut facilement être évité en faisant une affectation douce au lieu d'un choix statique / dynamique binaire. Nous y parvenons en attribuant chaque patch un *poids statique* $w_s$ et un *poids dynamique* $w_d$ avec $w_s + w_d = 1$. La vitesse d'écoulement $v$ est utilisé pour déterminer directement le poids dynamique en utilisant une fonction de rampe coupée (voir la figure 6.3). La valeur du paramètre $\alpha$ est fixé de manière empirique.

Puisque le principe de notre méthode est de séparer une histogramme BOW en deux selon un critère arbitraire, le nombre des patch de chaque coté sera pas constant. Ainsi une technique spéciale de normalisation est nécessaire pour empêcher un coté de l'histogramme à dominer le feature. On compare 3 stratégies de normalisation:

- La normalisation L1 par défaut ramène la somme du entier feature à 1.

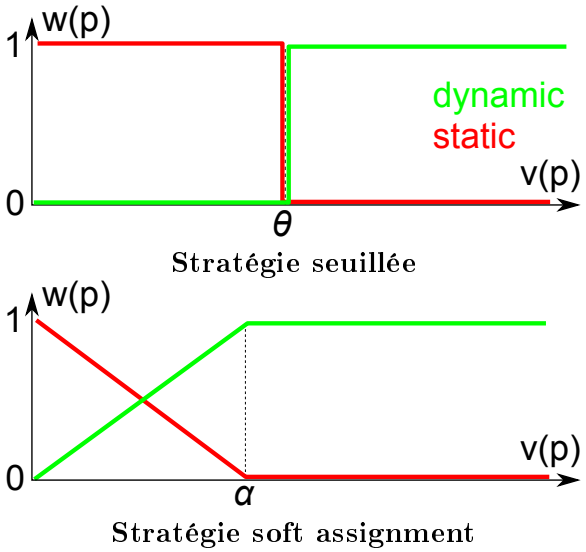


Figure 1.10: Les fonctions qui décrivent les poids statiques et dynamiques en fonction de la vélocité du patch

- Dans la deuxième stratégie, les histogrammes statiques et dynamiques sont normalisées avec norme L1 séparément
- La troisième stratégie est basée par composante, non par vecteur, ce qu'on appelle "svmscale" selon LibSVM [63]. Cette méthode redimensionne chaque composante en min-max et en suite normalise la composante L1.

Le protocole expérimental reste pareil que dans les expériences précédentes sur TRECVID 2010. On utilise LibSVM pour entrainer des classifieurs concept. On optimise les paramètres d'apprentissage $C$ et $\gamma$ par recherche de force brute. Si on compte l'approche DSIFT de base, il y a 11 combinaisons valides de séparation et normalisation. On exemplifie le résultat de classification sur 2 concepts: "Mountain" - prédominant statique et "Running" - prédominant dynamique. Les nombres sont présentes dans le tableau 6.1:

Ces résultats confirment que la meilleure combinaison est entre le 'soft assignment' et la normalisation 'svmscale'. L'amélioration en précision par rapport au baseline DSIFT-BOW confirme l'efficacité de la méthode et permet de continuer la recherche avec des représentations encore plus fins du mouvement.

| *Mountain* | default | trhesh=$\theta_1$ | thresh=$\theta_2$ | soft |
|---|---|---|---|---|
| globalL1 | 0.13164 | 0.0297 | 0.0581 | 0.0316 |
| sepL1 | N/A | 0.0045 | 0.0173 | 0.0003 |
| svmscale | 0.2699 | 0.2866 | 0.1808 | 0.2605 |
| *Running* | default | trhesh=$\theta_1$ | thresh=$\theta_2$ | soft |
| globalL1 | 0.02351 | 0.001 | 0.0235 | 0.0134 |
| sepL1 | N/A | 0.001 | 0.0235 | 0.0009 |
| svmscale | 0.01811 | 0.03843 | 0.0198 | 0.0474 |

Table 1.5: Précision moyenne des différents stratégies de séparation et normalisation pour 2 concepts

### 1.5.1 Z* features

Dans la section précédente on a montré que la séparation statique/dynamique des DSIFT amène de l'information significative au classifieur. Dans cette section on exploite non seulement la presence/absence de mouvement mais aussi l'information de direction en produisant des nouvelles features qu'on appelle *Z* features*. En utilisant une technique simple de binning, on crée 3 features nomées ZN, ZHV et ZLRUD.

La figure 6.6 présente une vue d'ensemble pour la méthode: des patches DSIFT sont assignés (a) au codewords (rouge et vert). Le DSIFT-BOW (b) est construit en comptant les occurrences de chaque codeword. Dans notre approche (c) les vecteurs de mouvement sont groupés par codeword et quantifiés (d) dans des histogrames 2D. Les lettres qui décrivent chaque bac sont des initiales pour Horizontal, Vertical, Left, Right, Up, Down et Zero. Le H est l'union de L et R, V est l'union de U et D. Les 3 features qu'on propose sont:

1. ZN qui décrit *si* le patch bouge
2. ZHV qui discrimine entre des patch en mouvement horizontal versus vertical, donc décrit *orientation*, et
3. ZLRUD qui décrit les 4 point cardinales, donc la *direction*

Le montage expérimental suit de près l'évaluation TRECVID 2012 SIN. Nous évaluons 50 concepts, avec des annotations d'entraînement éparses disponibles sur un ensemble de développement contenant 400,289 séquences, et appliqué sur un ensemble de 145,634 séquences de test. Pour bénéficier de

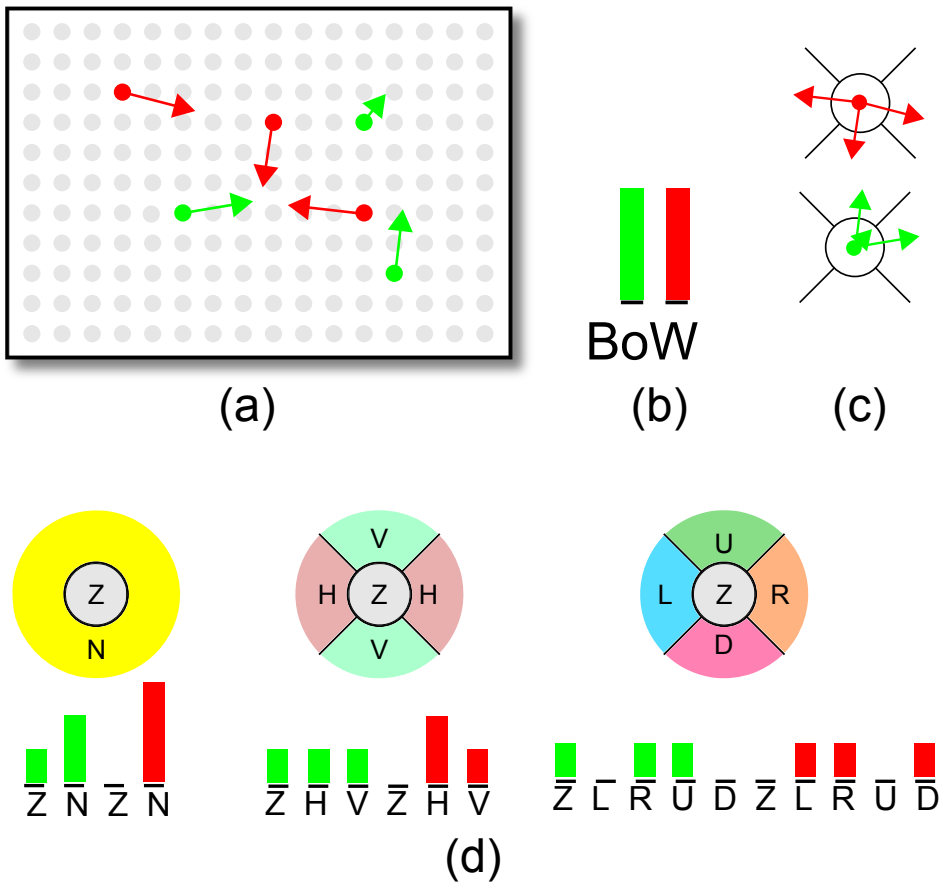


Figure 1.11: Construction des Z* features

la puissance de classification des noyaux non-linéaires, on emploi les approximations décrites en [65]. On applique le mapping utilisant le Homogeneous Kernel Map d'ordre $N = 3$ implémenté dans la librairie Scikit-learn [66] ce qui donne une représentation de $7\times$ la taille originale. Entrainer ce vecteur avec un SVM linéaire donne une approximation de la classification avec noyau $\chi 2$ additif mais dans une fraction du temps de calcul. Pour cela on utilise Liblinear [67] sur une moitié de l'ensemble de développement (197,185 séquences) et on se sert de l'autre moitié pour optimiser le paramètre $C$ de l'SVM.

Le tableau 6.3 montre le résultat de fusion pour les classifieurs DSIFT,

ZN, ZHV, ZLRUD. En dépit de son contenu enrichi d'information mouvement, les Z* features ont un MAP un peu plus faible que DSIFT. Pourtant en fusion on trouve quand même une amélioration signifiante.

| baseline | DSIFT 500 | ZN 1000 | ZHV 1500 | ZLRUD 2500 | fusion | gain |
|---|---|---|---|---|---|---|
| 0.1798 | 0.0985 | 0.0926 | 0.0844 | 0.0779 | 0.1898 | 5.55% |
| 0.1798 | | 0.0926 | 0.0844 | 0.0779 | 0.1882 | 4.63% |
| | 0.0985 | 0.0926 | 0.0844 | 0.0779 | 0.1008 | 2.37% |
| | | 0.0926 | 0.0844 | 0.0779 | 0.0917 | -0.95% |

Table 1.6: MAP pour les différentes fusions de DSIFT et Z* sur TRECVID 2012

L'avantage des Z* reste dans le coût réduit de calcul: si on dispose des vecteurs BOW DSIFT on peut facilement extraire un flou optique éparse sur le keyframe et obtenir 3 nouvelles features qui améliore le baseline en fusion. Les tests de validation de NIST confirme l'amélioration via fusion avec confiance 0.05

### 1.5.2 Sommaire sur les temps de calcul

Dans cette section on présente une diagramme (figure 6.9) qui montre le temps moyen nécessaire pour le traitement complet d'un plan: extraction, coding, classification. On a mis tous les descripteurs proposés dans cette thèse et aussi les plus connus descripteurs spatio-temporels dans la literature: STIP, Dense Trajectories et MoSIFT. La classification pour DT, MoSIFT et STIP ne figure pas car elle n'a pas été faite (en raison du temps trop long de calcul). Il est évident que les features proposés sont plus rapides. Aussi, les Z* features sauve encore de temps par la réutilisation des données DSIFT-BOW, qui en pratique sont présentes dans la plupart des systèmes. A noter que l'échelle diffère entre les catégories.

## 1.6 Conclusions

Dans cette section nous présentons les conclusions générales de la thèse:

1. Concernant la création des features spatio-temporelles spécifiques à la détection des concepts, on a proposé **3 nouvelles types de feature**

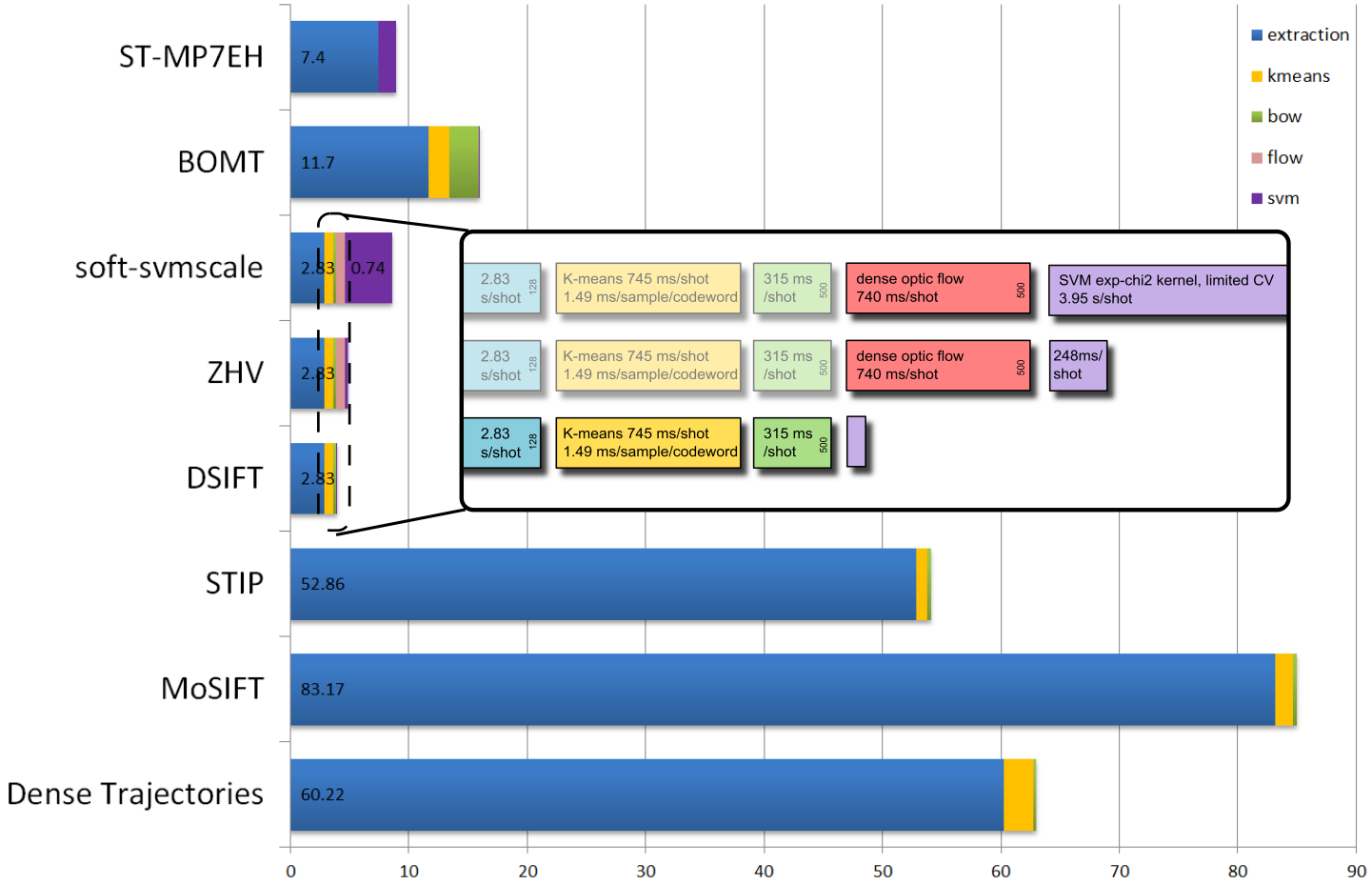


Figure 1.12: Timeline du temps moyen de calcul pour chaque étape du traitement d'un plan pour les features proposés, DSIFT et des features ST état de l'art. Extraction est en bleu, construction codebook en jaune, allocation BOW en vert, classification en violet, extraction flou optique en rouge. Les premières 3 phases de DSIFT peuvent être réutilisés par Z*; cela réduit le temps total à moins d'une seconde par plan. Temps de classification pour les derniers 3 features est inconnu.

**spatio-temporelle** et on a expérimenté avec les 3 dans le contexte pratique de la détection. Le prêt direct des techniques de détection d'action humaine, les extensions au spatio-temporel des méthodes image et des techniques simplistes de corrélation ont été évalués et rejetés dans le chapitre 4. Des nouvelles features proposées dans les chapitres 4.3, 5 et 6 montre des améliorations dans TRECVID via fusion ainsi que des résultats intéressants sur des autres bases.

2. En discutant le coût de calcul des nouvelles features, figure 6.9 montre le temps moyen de calcul pour chaque étape de calcul pour un plan, de l'extraction jusqu'à la classification. Visiblement les méthodes proposées sont au moins **quatre fois plus rapides que l'état de l'art**. Également on a montré que BOMT atteint une précision de 88.97% sur KTH, par rapport au 92.10% du STIP, mais il est extrait dans un temps 4.5 fois plus court. On trouve remarquable le fait d'avoir une méthode simpliste et très rapide qui est plus précise que l'état de l'art à aussi récemment que 2008 [38,43,68,69]. Sur TRECVID on dispose pas des données expérimentales pour la comparaison avec STIP ou Dense Trajectories, mais il est raisonnable d'estimer que BOMT donne toujours un peu moins de performance que STIP.

3. Tous les features proposées **améliore le MAP du système en fusion sur TRECVID**: pour 10 concepts dans TRECVID 2011, ST-MP7EH et SIFT produisent une amélioration de 22%, BOMT améliore par 5% lors de la fusion avec DSIFT et MoSIFT sur TRECVID 2010, les Z* features emportent ensemble à un système pour TRECVID 2012 contenant beaucoup des caractéristiques une amélioration du MAP de 0.179 à 0.189. Ces améliorations sont validées par des tests standard de TRECVID qui confirment la signifiance statistique pour un niveau de confiance de 0.05. Dans des certains cas, le gain obtenu par fusion est surprenant à cause du MAP plutôt bas du descripteur seul.

### 1.6.1 Considerations sur TRECVID

Quelques considérations sur l'utilisation des caractéristiques spatio-temporelles dans la récupération vidéo, en particulier dans TRECVID. Au moment de la rédaction de ce document, les informations sur la contribution du STIP ou MoSIFT au TRECVID est insuffisante. Cela rend impossible une comparaison des features proposées avec un éventuel ''état de l'art dans la récupération vidéo basée sur le spatio-temporel". Cette manque est probablement due au temps de calcul excessif. Effectivement, en se basant sur les données

présentés dans la figure 6.9, le temps total d'extraction pour STIP sur les 266,473 plans de TRECVID 2010 prendrait à peu prés 185 jours de temps CPU, sans compter les étapes BOW ou SVM qui suivent l'extraction.

### 1.6.2 L'avenir du domaine

Les contributions présentées dans cette thèse dépendent dans leur phases initiales sur des méthodes de **vision par ordinateur**. ST-MP7EH utilise le détecteur de bords Sobel. BOMT extrait le mouvement du première plan en calculant la différence entre cadres successifs. Les features Z* font la quantification de la direction du mouvement selon le flou optique de Lukas-Kanade. Dans tous les features proposées on utilise la stabilisation caméra basée sur RANSAC. Tout bruit provenant des étapes de vision bas-niveau a des grandes répercussions sur l'entière chaine de traitement et par conséquence baisse la performance du système. On rappelle que par exemple dans BOMT (section 5.6) l'usage des boite englobantes manuellement annotées augmente la précision par 37%. Des meilleurs méthodes de **séparation avant/arrière plan** aideront beaucoup le calcul du BOMT. Un possible candidat pour ce rôle est la segmentation spatio-temporelle de Brox [70], mais pour le moment le coût de calcul de quelques secondes par paire de cadres rend infaisable l'approche sur TRECVID. Des méthodes encore plus avancées comme la méthode MRF de Liu [71] pourraient être utilisées pour segmenter l'avant-plan mais aussi à très grand coût de calcul. A noter que l'élimination des vidéos avec tremblement de caméra augmente la précision du BOMT par 15%. Ceci est un signe que nos méthodes de compensation de mouvement caméra sont pas assez robustes. Des approches à multiples homographies comme [72] donnent des résultats plus précises mais elles sont toujours pas scalables.

Les Z* features déterminent le mouvement de chaque patch DSIFT en utilisant le flou optique de Lukas-Kanade. Il existe trop d'implémentations de flou optique éparse pour pouvoir les compter (dans nos expériences on utilise celui de OpenCV [62]), mais le choix des paramètres optimaux sur une collection comme TRECVID reste pas claire. Des mesurements empiriques sont difficile à obtenir car un changement de configuration dans l'étape d'extraction demande que l'expérience entière soit refaite plusieurs fois. Peut-être qu'une estimation automatique des paramètres du flou optique selon les données mêmes pourrait réduire ces effets.

Bien que beaucoup des chercheurs utilisent encore les SVM avec noyau RBF avec des résultats quasi-état de l'art, quelques méthodes de modélisation innovantes ont été développées. Dans l'étape de pooling, les **vecteurs**

Figure 1.13: Examples des plans récupérés par des features spatio-temporelles Z* dans TRECVID 2010 sur 50 concepts.

**de Fischer** [28] mélangent les approches génératives et discriminatives en modelant une GMM selon la distribution des features. Des bons résultats sur TRECVID SIN et PASCAL VOC recommandent cette approches pour un remplaçant du BOW, malgré le coût de calcul.

L'état de l'art en classification pour la récupération est en cours de tourner des SVM vers des **réseaux de neurones profondes**. Le système proposé par Krizhevsky et. al. sur ImageNet [73] à que récemment été appliqué au TRECVID, avec des résultats excellents. A noter que dans le deep learning, les features bas-niveau ne sont plus crée à main (par exemple [74] utilise un volume spatio-temporel aplati de taille $15 \times 15 \times 7$ du gradient spatio-temporel). Plutôt que des caractéristiques conçues pour la détection, en deep learning les features sont apprises en même temps que les modèles, sous forme de filtres de convolution.

# Chapter 2

# Introduction

The quantity of available digital video data has outgrown our ability to organize it. Since the beginning of the last decade digital cameras and cell phones capable of recording video have exploded in popularity. Today the use of video recording hardware such as notebooks, smartphones and cameras is widespread. Consequently, consumers record and upload videos online on video sharing sites like Youtube, which in turn grows in size and popularity. In 2006, there were 6 million videos on Youtube, in 2008 there were 60 million and one year later 120 million. In 2013 there is an estimated 3 billion videos, with 100 hours uploaded every minute and 6 billion hours of video are watched every month[1]. This exponential growth is one of the many practical observations matching Moore's law [1] which states that the number of transistors in integrated circuits doubles every two years. This translates into more and more CPU power being available for the same cost. Similar observations can be made on hard disk capacity and image sensor resolutions. The combination of these effects results in tremendous amounts of video data that needs to be efficiently compressed, stored, indexed and searched.

Unfortunately, the technology for retrieval in large video collections has failed to keep pace. The size of the new collections makes intelligent search extremely computationally intensive. Current practical solutions can retrieve video based on text summarized from keywords and metadata. The key-

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

words are typically extracted from closed captioning, speech analysis, manual annotation and optical character recognition. While text-based search is very efficient, annotating the videos is labor-intensive. Keyword searches are limited by existing annotations and thus one can not search for unlabeled events. Finally, manual annotation does not scale to large amounts of data and it is difficult to do in real time. Many research video retrieval systems that analyze visual information only process the *keyframes* in the video. In particular it can be observed in TRECVID, a research campaign dedicated to video content analysis, that keyframe-based descriptor extraction is still the dominating strategy. It is our belief that video content outside of the scope of the keyframe - the *spatio-temporal* content - can be successfully extracted and used in video retrieval as additional, complementary information to keyframe descriptors. Our focus is on harvesting spatio-temporal content at low level, thus developing *spatio-temporal features* in order to perform semantic indexing.

The spatio-temporal content of a video is however a well studied topic in other scientific communities. For example, action recognition [2] has long been performed by using spatio-temporal features. The few existing spatio-temporal features that are currently used in practice in TRECVID (e.g. STIP) have first been proposed to recognize human actions. The problem is that these features are not easily scalable to video retrieval level because of two main reasons: the computational complexity and the loss of generality due to applying them to general concepts. In this thesis, we provide new alternative features that scale well to concept detection and work well in fusion with classical keyframe features.

## 2.1 TRECVID Semantic Indexing and VideoSense

TRECVID [3–5] is an annual NIST[2] sponsored international benchmarking activity that encourages research in video retrieval. Participating teams of researchers test their video retrieval systems on a standardized collection and compare results obtained through well established scoring methods. Out of the several tasks, our team participates in the "Semantic Indexing" task (SIN, formerly "high level feature extraction") which is essentially a concept detection challenge.

The SIN video collection almost doubles in size with every edition. In 2012, the collection consisted of roughly 800 hours of Internet Archive videos

[2]National Institute of Standards and Technology
`http://www.nist.gov/`

with Creative Commons licenses in MPEG-4/H.264 with duration between 10 seconds and 3.5 minutes. An automatic shot boundary detection on the videos is provided, which indicates a total of approx. 545,923 shots in 28,123 videos. The dataset is split into a development set of 400,289 shots and a test set of 145,634 shots. By today's standards, the videos are rather low resolution (in average $240 \times 335$, see table 3.1 for reference). The video content is completely unconstrained due to the nature of the web collections. A comparison of the different editions is available in table 2.1.

| SIN edition | 2007 | 2010 | 2011 | 2012 | 2013 |
|---|---|---|---|---|---|
| total playback | 100 hours | 400 hours | 600 hours | 800 hours | 1400 hours |
| video count | 47 shows | 11,644 | 19,860 | 28,123 | 30,543 |
| shot count | 18k | 266k | 403k | 546k | 883k |
| concept count light / full | 10/39 | 10/39 | 50/346 | 50/346 | 50/346 |
| source | TV | IACC | IACC | IACC | IACC |

Table 2.1: Video collections for TRECVID Semantic Indexing 2007, 2010-2013

There are two versions of the task depending on how many concepts are considered: the "Full" task is 346 concepts and the "Light" task is 50 concepts. After submission only 80 concepts in the Full task and 20 in the Light task are evaluated by human assessors. We can divide the concepts by semantic category (see figure 2.1). The majority of concepts are *visually* identifiable: objects ("Airplane") or scenes ("Cityscape"). In contrast, other concepts are *time* relevant and describe actions ("Running") or events ("Explosion"). A number of concepts are *abstract* ("Science/Technology") or *composite* concepts (e.g. concept "Weather" does not have a single visual representation but may contain instances of sun, clouds, thunder, snowflakes, etc.). A few concepts portray emotions ("Disgust"). Some concepts are *intersections* of several concepts ("People marching" is both objective "People" and action "Marching"). Lastly, some concepts describe specific *instances* of other concepts ("George Bush" as an instance of "Politicians"). Some "is a" relations between concepts are modeled into an ontology that is used to propagate annotations.

Annotations are binary values assigned to shots that tell if the concept is present or not present in the shot. TRECVID annotations are the result of a collaborative effort to manually assign labels by visually inspecting keyframes or video snippets of the shot [75]. As a result of human label-


Figure 2.1: Examples of TRECVID concepts.

ing the annotations are sparse, with many shots skipped because of time constraints and some inconclusive shots intentionally skipped.

For each semantic concept, participants must submit a ranked list of 2000 shots from the test set that most likely contain the concept. The submitted shots are selectively evaluated so that an estimate of the retrieval performance can be calculated. It is important to note that non-evaluated shots are counted as non-relevant. The evaluation metric is the *xinfAP* (mean extended inferred average precision) [76], which is a sample-based estimate of the mean average precision (MAP). Average precision is the average of precisions at relevant document recall levels. It is used in retrieval to measure how close are the relevant documents to the head of the list. The MAP is simply the mean across concepts of average precisions. Since for the 2013 edition the development data coincides with the entire 2012 dev+test data, annotations for 2013 can be used to give a better estimate of the MAP than the xinfAP because of the new 2013 annotations. This is why in our experiments we always use for evaluation purposes the largest body of annotation available.

The research carried out for this thesis is founded by the French National Research Agency[3] (ANR), under the project VideoSense[4]. The VideoSense project has a similar framework to TRECVID. Video data is provided by

[3] `http://www.agence-nationale-recherche.fr/`

[4] `http://www.videosense.org/`, project code ANR-09-CORD-026

the project partner Ghanni[5]. Additionally, the VideoSense consortium has made submission to TRECVID SIN in the 2011, 2012 and 2013 editions.

## 2.2 Motivation

The particularity of our research is to classify video using dynamic, motion and spatio-temporal content descriptors. Because of either the apparent progress achieved in CBIR or the higher computational complexity of spatio-temporal approaches, current state-of-the-art in video retrieval relies mostly on keyframe analysis. With very few exceptions, TRECVID concept detection systems are essentially image concept classifiers that approximate a video by one or a few frames. The following table is the result of a survey on the use of dynamic descriptors compared to static descriptors in TRECVID 2012. A feature is counted as one for a participant, regardless of how many variants of it uses in that paper. Out of 57 teams with submissions in the different tasks, 36 mention space-time features. However, if we restrict the search to Semantic Indexing, only 7 papers (if we exclude Eurecom's submission that runs the ST-MP7EH described in section 4.3) out of 25 contain dynamic descriptors, yet SIFT is present in almost all of them.

| | SIN | all |
|---|---|---|
| STIP [77] | 2 | 14 |
| DT [48] | 1 | 4 |
| MoSIFT [46] | 1 | 5 |
| MHI [35] | 0 | 2 |
| HOG3D [44] | 0 | 1 |
| ISA [78] | 0 | 2 |
| Other motion feature | 3 | 8 |
| SIFT [17]/SURF | 23 | 50+ |
| participating teams | 25 | 57 |

Table 2.2: Counting dynamic descriptors vs. SIFT in TRECVID 2012 publications

We can safely conclude that although TRECVID is a video retrieval themed campaign, most of the work is done by reducing videos to keyframes. Intuitively, this leads to several questions: what if the concept is present in

[5] http://www.ghanni.com/

the shot but not in the selected keyframe? How do we differentiate between (hypothetical) concepts *Airplane landing* and *Airplane taking off* if we only look at one frame in the shot?

Content based video retrieval is still an open research problem [79]. In a very comprehensive survey paper from 2009, Ren et. al. [47] reference 171 research papers on spatio-temporal video retrieval but one single practical example of such a system[6]. A more recent survey paper by Hu et. al. [80] lists as the first item on the list of topics requiring further research "distinguishing between background and foreground motion, detecting moving objects and events, combining static and motion features and constructing motion-based indices".

The existing spatio-temporal features are computationally heavy. Since TRECVID is a results-oriented research endeavor, the computational feasibility of a descriptor is an important factor when met with deadlines. In table 5.2 and figure 6.9 proposed descriptors are compared against some dynamic descriptors in literature. We can calculate that with an extraction speed of 5 hours for the 50 minute KTH video corpus (with smaller resolution than TRECVID videos), the feature extraction time on the 400 hour TRECVID 2010 collection would take more than 14 weeks of CPU time. Such computation would require parallelization and hardware resources way beyond our (and many other researchers) resources.

## 2.3 Problem Statement

The reality of the video retrieval problem is that at this moment keyframe methods are an acceptable compromise. Considering the relatively low computational cost of image descriptors and the very good performance of image classifiers, the existing spatio-temporal features are slow, heavy and perform poorly in concept detection. Relying on the much simpler image retrieval problem is reasonable because of the success of CBIR itself.

1. Since the existing features are either impractically slow or not sufficiently performing, a new breed of *spatio-temporal features tailored for concept detection* is needed.

2. Instead of focusing on accuracy on smaller datasets and completely ignoring CPU time, dynamic descriptors that scale well to the big data

[6]The ARTEMIS-UBIMEDIA mentioned in the survey paper has since the year of publication (2009) participated in TRECVID Instance Search task with a system based on keyframe information only

multi-label retrieval problem are needed. In the context of real-world applications like TRECVID, an optimum in *balance between speed and precision* is sought. Figure 2.2 illustrates this idea.

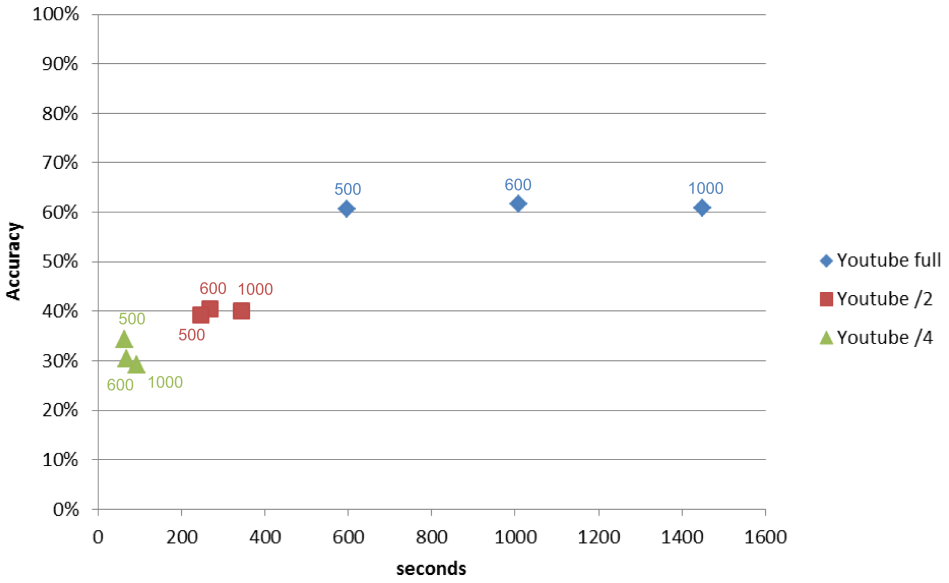


Figure 2.2: Average accuracy of MoSIFT features on the Youtube dataset vs. average classification time. Colors represent the downsampling rate: full-size videos in blue, video resolution sampled at 50% in red and at 25% in green. The 3 points correspond to different sizes of the visual vocabulary: 500, 600 and 1000. Is it worth downsampling videos by factor 2 in order to double speed but lose 10% accuracy? Is it worth increasing the BOW size from 500 to 600, thus increasing training time by 66% in order to go from 0.605 to 0.617 accuracy?

3. Good spatio-temporal features are not necessarily accurate to the point of replacing keyframe approaches, but rather *complementary* to them, so that the improvement is obtained by *feature fusion*. We are not searching for spatio-temporal features that work well on their own, but features that when added to a fusion pool bring significant improvement.

## 2.4 Challenges

The size of the TRECVID video collection makes spatio-temporal retrieval a challenge. Table 3.1 on page 76 in the State of the Art chapter, gives an overview of quantitative measurements across different action recognition and image classification datasets, along with known performance values for several of the state-of-the art descriptors. These datasets are presented in detail in chapter 3. From the data in table 3.1 we can see that as a classification problem TRECVID is somewhat comparable in size with image classification benchmarks PASCAL VOC and ImageNet, but the descriptor extraction, which depends on the quantity of raw data, is much higher than for all of the action datasets (combined). If we extrapolate the CPU time required to extract STIP on the KTH dataset to the 800 hours of TRECVID 2012 we get 200 days. Although not impossible, such a computation is highly impractical and requires some form of parallel computing. The considerable number of concepts demands a generic approach; employing a special method for each concept is impractical. Variability of the concepts themselves makes the development of such a generic approach difficult.

*Computational considerations* are probably the main reason why researchers are reluctant to adopt of spatio-temporal features in concept detection. Since the main available ST features like HOG3D, STIP, Dense Trajectories and MoSIFT have been applied first on action recognition smaller datasets (see table 3.1), they are not built to scale up. On small datasets like KTH, slowdowns incurred by CPU-intensive computations such as pixel-dense optic flow are not noticeable. This allows fine-grained, advanced visual analysis methods to be tested. Indeed, while action features detect local motion patterns regardless of whether the underlying motion is human, there is no theoretical reason to prevent the extension to concept detection. However, the size of the TRECVID collection makes the total computation time too long to permit experimental tuning of the method. This is an important practical constraint that prevents most researchers from studying space-time features on TRECVID. Also the storage cost of the descriptor vector for some features is too high, to the point that some researchers [48] extract and assign a BOW vector directly in one pass without storing the feature data.

Additionally, there are difficulties with the video data itself. More than 75% of all TRECVID shots contain important *camera motion*. This is not an issue for keyframe methods because the keyframes are usually chosen to minimize motion, but for spatio-temporal analysis camera motion is problematic. Trajectories are subject to the camera movement, which must be correctly modeled. Motion blur disrupts most computer vision functions.

Tracking is unreliable as objects may be quickly moving in and out of the frame. Camera motion stabilization methods exist and help alleviate some of these problems, but are either slow or not robust enough.

Another important issue is the video quality. It can be seen in table 3.1 that TRECVID has higher values in all criteria except video resolution. Compression artifacts and small size of objects add noise to feature extraction. A common problem with web videos is that some lower-quality recording devices attempt to "fake" a standard 25fps framerate by recording at lower rates and duplicating frames. This results in artificially jagged movements.

Some videos contain graphics and text superimposed on footage. While some may argue that these elements are an integral part of the video semantic content and should be indiscriminately extracted and classified (e.g. for the concept "News Studio" the logo of the studio and the channel animations are a good indication of the concept), some technical issues interfere. Static or animated text can "fool" the camera stabilization which may consider the text as background and the rest of the frame as foreground. Training on shots containing too much text can in fact lead to classifiers that simply detect shots with text. Picture-in-picture situations are hard to classify because they show content from two very different scenes.

There are significant challenges in the classification stage as well. Classifiers have to deal with very high *intra-class variation* caused by several possible phenomena. Natural variations such as illumination, occlusions, scale change are issues that carry over from CBIR. Low resolution and camera movement contribute heavily to detection noise, which causes parasite variation in features. Finally, some concept simply have genuine variability, e.g. concept "Weather" can mean a sunny sky, a thunderstorm or an image of a snowflake. Figure 2.3 exemplifies these facts.

Another problem in classification is the heavy imbalance between negative and positive learning classes. For the 50 concepts of the 2012 Lite SIN task, the average ratio of positives to negatives is 9.7%, with some concepts showing very few positives in the test set, e.g. concept 'Helicopter Hovering' has only 18 positive instances. The impact of class imbalance is mostly related to noise. Imbalance in the training set means that classifiers will learn that most of the samples are negative with very few exceptions, which means that when classifying a test sample the predicted label will always be negative. To compensate for this effect the positive class samples can be assigned a different higher weight such as in LibSVM [63], at the cost of higher noise. Imbalance in the test set means that we are in a "needle in a haystack" situation: small noise in the classification scores of the few


Figure 2.3: Image representatives for some classes of the Pascal VOC '07 Challenge. Intra-class variability is evident. Source:`http://raweb.inria.fr/rapportsactivite/RA2007/lear/uid64.html`

positives induce high variations in the average precision. Finding the needle becomes more of a game of chance than of systematic searching.

Not all shots in the TRECVID dataset are annotated. In average annotations for a concept cover about 10% of the data. Missing annotations in the test set negatively affect the quality of the MAP estimation, because they are by default considered nonrelevant yet in reality they may be unannotated relevant shots, correctly retrieved by the system. Another problem is that a different set of shots is annotated for every concept, and their intersection is practically empty. Consequently average precision for different concepts are not comparable and also multi-class strategies are not applicable. Annotation is made by collaborative human effort and although there are double-checking mechanisms in the annotation process, situations when duplicate shots have contradicting annotations exist.

Motion and spatio-temporal information is helpful in scene understanding for humans, but much to a lesser degree than static visual perception. Image features have higher AP than motion features. A common situation found at fusion level in our experiments happens when attempting score fusion between a strong classifier (SIFT) and a weak spatio-temporal classifier. It is not uncommon when fusing motion and image descriptors that fusion strategies result in diminishing the contribution of motion features to the

point of completely ignoring them, as a result of poor stand-alone precision.

## 2.5 Thesis Contributions

In response to the Problem Statement, we can summarize the main contributions present in this thesis in the following points:

1. We experiment with the existing action recognition feature HOG3D on TRECVID in chapter 4.1 and confirm that the performance is insufficient to justify its use as a standard dynamic baseline. We reach the same conclusion in chapter 4.2 with a multi-keyframe SIFT classifier variant, thus reiterating the need for new spatio-temporal features. Simplistic solutions such as classifying concepts based on what camera motion type is predominant are also discarded.

2. In chapter 4.3 we propose a global descriptor that generalizes the MPEG-7 edge histogram feature by simply computing averages and standard deviations of the time series resulting from the extraction of the edge histogram regularly through time. We prove experimentally that significant improvement in MAP can be achieved by fusing even a simplistic spatio-temporal feature based classifier with a traditional SIFT classifier.

3. The Bag of Motion Templates method in chapter 5 extends on Bradski's [6] motion template idea by extracting parts of motion and then using them as visual words in a Bag of Words-like scheme. When applied on the KTH action recognition dataset, BOMT performs at about 88% compared to 92.1% of STIP but is computed 4 times faster, thus fulfilling the "speed vs accuracy" criteria in section 2.3. Again, improvement through linear fusion of BOMT with other descriptors on TRECVID data is shown to be around 5%.

4. In chapter 6 the idea of enriching a Dense SIFT Bag of Words feature with motion information is presented. First, a simple separation between static and dynamic DSIFT patches leading to two Bag of Words histograms is shown to improve performance on concepts containing movement. Further separation into 4 directions creating so-called Z* features permits the MAP of the system to raise by 5.5% w.r.t. a strong TRECVID keyframe baseline with the proper SVM learning and fusion technique.

# Chapter 3

# State of the Art

**Content analysis** is an umbrella term representing methods for analysing and understanding collections of media. Historically, content analysis developed techniques for describing text, decades before the first Internet search service. One important topic in content based analysis is **Information Retrieval**, which deals with obtaining documents relevant to a query from a large database. In 1992, the US Department of Defense along with the National Institute of Standards and Technology (NIST), cosponsored the Text Retrieval Conference (TREC). The aim of this was to look into the information retrieval community by supplying the infrastructure that was needed for evaluation of text retrieval methodologies on a very large text collection. This catalyzed research on methods that scale to huge corpora. The introduction of web search engines has boosted the need for very large scale retrieval systems even further.

Advances in computing lead to the later inclusion of new modalities in Information Retrieval. The goal of **Content-based Image Retrieval** (CBIR) is to search for digital images in large databases. CBIR requires that the database be indexed based on information derived from the content of the image as opposed to other forms of Image Retrieval (e.g. Image meta search retrieves images based on metadata). This information is generally in the form of color, texture or shape descriptors that provide a good framework for defining content similarity. Today image search is becoming a standard

feature of web search engines, with systems like Google Image Search[1] and Tineye[2] crawling and indexing millions of online images. Active benchmarks for research in CBIR include ImageNET [81] and LabelMe [82].

Similarly, the idea behind retrieving video, sometimes denoted **Content-based Video Retrieval** (CBVR) is to search and index a large collection of video, in order to retrieve a video or a segment of video with content relevant to the query. The TRECVID [3] campaign encourages research on video retrieval and many researchers, including us, test and compare our methods on TRECVID data and evaluation. The SIN track of TRECVID is the benchmark for all the proposed descriptors in this thesis. Semantic Indexing in TRECVID is centered on learning classifiers from annotated video shots that can identify occurrences of *semantic concepts* in a test collection of videos.

However the transition from CBIR to CBVR is not straightforward. While both image and video content indexing retrieval rely on computer vision methods to analyze the low level data, video is more challenging for two reasons. First, a video collection is considerably larger than an image image collection with the same number of documents, which leads to steeper computational requirements in all the stages of the indexing process. Secondly, compared to image classification or object detection, where several techniques are so accurate that they have become *de facto* standards, there is no real effective content modeling method for video retrieval. Instead the tendency of SIN participants is to fall back to CBIR by keyframing videos.

Video content analysis is a large research theme, with many theoretical and technical aspects. In this chapter we will mainly concentrate on two views: semantic indexing and space-time methods in action recognition. In the first section we will present semantic indexing systems as a whole, with an emphasis on some of the most popular methods seen in TRECVID SIN. These methods are predominantly based on keyframe extraction. In the second section we present dynamic feature approaches established in the action recognition community that are general enough to be applicable to concepts and their associated datasets. In the final chapter we look at how the two domains intersect by giving an overview of practical implementations of space-time descriptors in semantic indexing and look at practical considerations for existing descriptors such as STIP.

Throughout this chapter it is we will often refer to table 3.1 on page 76 when presenting datasets and performances of ST features.

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

[2] `http://www.tineye.com/`


Figure 3.1: Example of CBIR engine: imgSeek can search for images that match a rough user painted sketch. Source: `http://www.imgseek.net/`

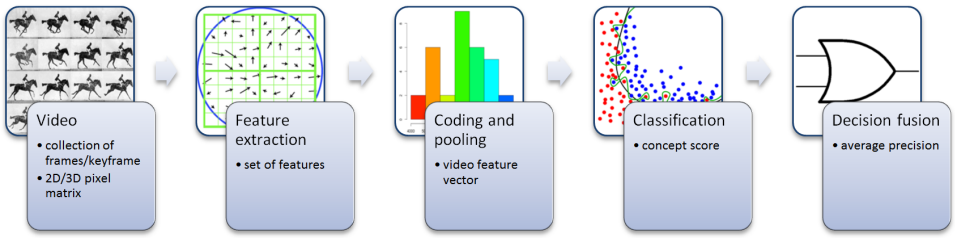


Figure 3.2: Overview of a Semantic Indexing system.

## 3.1 Standard practice in Semantic Indexing

The standard approach (figure 3.2) in concept detection is to represent the document as a collection of features. Features provide a numerical representation of different modalities of the content, such as color, edge distribution or texture for images. These features are aggregated into a document-specific feature vector. Feature vectors are then used as input to concept classifiers, which apply machine learning methods to learn from training data and give predictions on test data. The predictions are then used to retrieve the test videos that most likely contain the learned concept. At this point typically fusion between different classifiers is performed. The performance of such a system is measured in terms of mean average precision or precision-recall curve.

In the following sections we present some examples of popular methods used in Semantic Indexing corresponding to the different stages from figure 3.2

### 3.1.1 Commonly used descriptors

In theory descriptors are representations of content with low bandwidth (compact, low dimensionality), invariant to parasite variations (e.g. illumination, scale, position in frame, rotation, perspective, context etc.) and high discriminability (i.e. able to robustly tell apart documents containing different content).

#### Color descriptors

One of the oldest ways to index visual content was based on color. For example in the IBM QBIC [10] system, a 64 bin color histogram in L*a*b space, indexed into a 4096 element color table is used to index and retrieve images

in collections of images and video. The histogram is computed on small tiled subimages so that spatial color layout searches are possible. An example of this technique is the retrieval of paintings in the Hermitage Museum based on dominant colors and spatial arrangement[3].

Another classic image indexation feature is the color correlogram [7]. Early image compression algorithms made use of color index "palettes" by providing a reduced list of colors and a list of pixel regions that used that color. Taking the color index idea further, the color correlogram is a table of indexed color pairs where the $d$-th entry for row $(i, j)$ specifies the probability of finding a pixel of color $j$ at a distance $d$ from a pixel of color $i$ in the image.

### Audio descriptors

The use of Mel-frequency cepstral coefficients [11] (MFCC) is the foundation of modern speech processing. The vectorized form of the MFC spectrum is used by some systems [12, 13] to index the content of the audio channel in a video. In this thesis we do not focus on audio features.

### Texture descriptors

Experimentally textures have proven during the years to be an efficient way to index images, mostly because of the higher variability of textures compared to other forms of visual content. A few global texture representations have become popular because of their lower computational cost.

Gabor filters are parametric linear filters that respond well to edge features. It has been discovered that some of the receptive cells in the visual cortex of mammalian brains can be modeled by Gabor functions. A set of Gabor filters with different frequencies and orientations can effectively describe the edge distributions in an image. One of the early uses was to detect textures [8].

Wavelet representations give information about the variations in the image at different scales. The Discrete Wavelet Transform (DWT) represents an image as a sum of wavelet functions with different locations and scales. Coefficients of a filter bank of wavelet transforms are useful in image compression and can be used as features for retrieval. One example of wavelet used in image indexing is the Haar wavelet.

Local Binary Patterns [9] are very efficient texture operators that describe local texture by quickly thresholding the neighborhood of the pixel

[3] `http://www.hermitagemuseum.org/fcgi-bin/db2www/qbicSearch.mac/qbic?selLang=English`

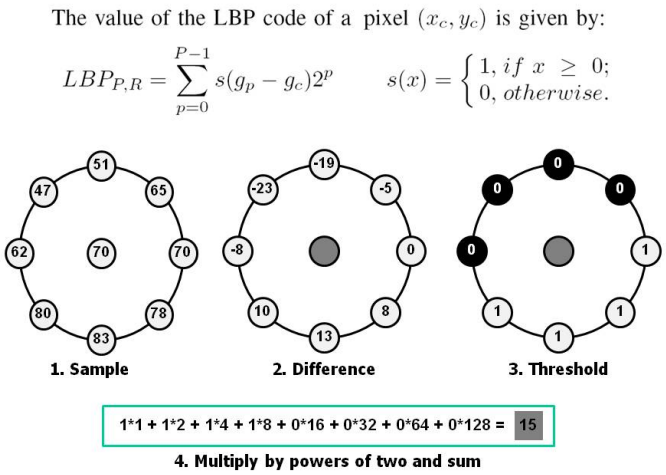


Figure 3.3: Computing LBP. The sign of the comparison corresponds to a bit in the final binary histogram. Image source: `http://www.scholarpedia.org/article/Local_Binary_Patterns`

(see figure 3.3). The comparison between the central pixel gray value and the tested pixel from the neighborhood is binarized so that the final representation is a small binary signature. Improvements on the original idea such as Color Orthogonal LBP combination [14] have been successfully tested in TRECVID [12].

### Local image patch features

These approaches are composed of two phases: the detector phase tells *where* in the image should local patches be extracted, and the descriptor defines *how* to represent the local patch around the interest point. The two stages are completely separable; basically any detector method is compatible with any local descriptor, provided a trivial interfacing is respected. For a review of local image feature detectors, check the survey by Tuytelaars [83]. Here are some examples of detectors relevant to our task:

- the Harris [15] corner detector is a rotation invariant detector. The principle is that if the eigenvalues of the structure tensor of the local intensity are two large positive values, that means the local patch shows variations in two directions, therefore it is a corner. Further improvements [16] have added scale autodetection (Harris-Laplace) and

invariance to affine deformation (Harris-Affine). Notably, the Harris operator has been extended to 3D by Laptev et. al. [42, 77] producing space-time interest points (STIP), described in detail in the next section of the chapter.

- the 2×2 Hessian matrix is obtained from first terms of Taylor expansion of the image intensity function. Based on local maxima of the trace of this matrix i.e. the Laplacian $\nabla$, blob-like structures in an image can be detected. The Laplacian can be approximated efficiently with a Difference of Gaussians (DoG) filter. Just as the Harris corner detector (and theorized by the same researchers [16]) there are Hessian-Laplace and Hessian-Affine detectors that estimate scale automatically and are invariant to affine transformation, respectively. Hessian detectors are commonly used as part of keypoint detectors for SIFT at multiple scales. In fact, the SIFT detector [17] is a multi-scale DoG detector that finds extrema points in 2D+scale space. The SURF [22] detector efficiently approximates the Gaussian second-order derivative by applying box filters; this computation is very fast thanks to an integral image representation.

- Maximally Stable Extremal Regions [18] (MSER) are connected components that define a region consistently brighter/darker than the pixels of its outer boundary. The region is defined by choosing a threshold value around which the regional area change is minimal. Consequently, MSERs are extremely robust to changes in global illumination and affine transformation. The MSER detector is used by some teams in TRECVID, for example the LEAR team in Multimedia Event Detection [84].

- Dense extraction does not use any kind of detection method. Dense features are simply extracted in a regular grid from the source image or video. In some scenarios dense extraction has proven to outperform interest point based detection in both image classification [19] and action recognition [20]. Dense SIFT is a regular feature found in many TRECVID systems [12, 13]

The output of detectors will be referred to as 'interest points', regardless of whether they originate from dense sampling or a feature detector per se. A local descriptor is computed in the local neighborhood of the interest point and must describe the visual content of the patch with as much invariability to perturbations as possible. We now present some notable descriptors:

- Scale Invariant Feature Transform [17] is the most common image descriptor used in image classification. The original version proposed by Lowe is a position-dependent histogram of local gradient directions forming a grid around the interest point. In order to obtain rotation invariance a dominant orientation is determined from the local gradients. The grid is positioned based on the dominant orientation and scaled according to the scale parameter from the detector. A rectangular $4 \times 4$ grid is laid out in the image domain, centered at the interest point, with its orientation determined by the main peak(s) in the histogram and with the spacing proportional to the detection scale of the interest point. For each point on this grid an orientation histogram of local gradient in computed. The histogram has 8 discrete orientation bins, and each point contributes a value proportional to the gradient magnitude (see figure 3.4). Additionally, the histogram contributions are weighted by a Gaussian window function centered at the interest point and with its size proportional to the detection scale, so that gradient near the interest point has more contribution. The dimensionality of the final feature is $4 * 4 * 8 = 128$. There are many extensions of SIFT that take into consideration color information: OpponentSIFT [21] is an example that has been used in indexing [12]. In Opponent SIFT, a standard 128 dimension SIFT descriptor is computed on each of the three opponent color spaces and the 3 vectors are concatenated.

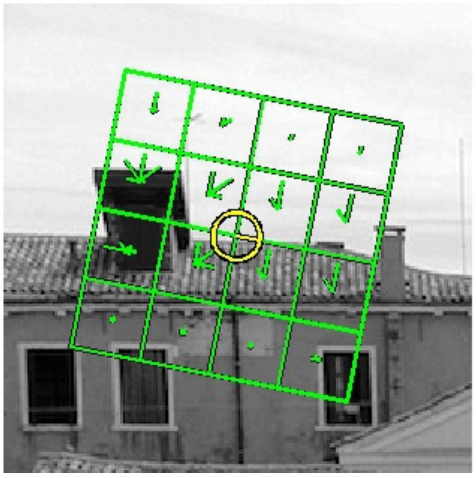

Figure 3.4: An example of SIFT grid with local orientation histograms displayed. Image source: `http://www.vlfeat.org/overview/sift.html`

- The SURF [22] descriptor shares some similarities with the SIFT detector, in that an oriented quadratic grid with $4 \times 4$ subregions is centered on the interest point (see figure 3.5). The difference is in the quantized measure: instead of gradient, horizontal and vertical responses to the Haar wavelet $d_x, d_y$ are characterized by 4 sums: $\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|$. The key to SURF's efficiency is the fast computation of the wavelet responses by using integral images. SURF produces a vector of dimension $4 * 4 * 4 = 64$.

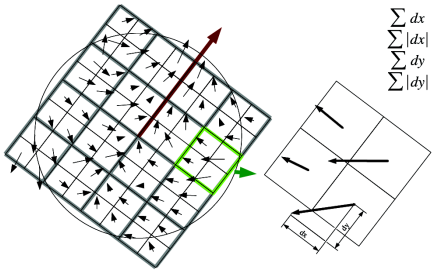


Figure 3.5: The SURF descriptor. In every $2 \times 2$ cell (highlighted in green), the sums of the $d_x$ and $d_y$ Haar wavelet responses are represented by sums of $d_x, |d_x|, d_y, |d_y|$ Image source: [22]

- Histograms of Oriented Gradients (HOG) [23] have been first proposed to recognize pedestrians in static images. In the original scenario HOG features were extracted densely from cells with some overlap. The HOG feature is computed as follows. First, a gradient map of the image is created by filtering with gradient filters $[-1, 0, 1]$ in the $x$ and $y$ directions, without any smoothing. Using the gradient values, an orientation map of the unsigned gradient is computed (with values from 0 to 180 degrees), and binned into a 9 bin histogram. The weight contribution on each bin is usually the gradient magnitude. In the typical R-HOG implementation, the orientation histogram is computed on 6x6 pixel cells that form 3x3 cell blocks. There is a vector normalization step that happens at block level by a clipped (to a maximal 0.2 value) L2-norm. The size of the descriptor is equal to *number of blocks* $\times$ *cells per block* $\times$ *bins per cell*, which in the standard approach is 3780 for a $64 \times 128$ detection window. There is no scale or rotation notion in this descriptor. HOG is a very general and configurable descriptor. The edge histogram feature from MPEG-7 [24, 25] can be seen as a

HOG-like descriptor with $4 \times 4$ blocks/cells with 5 bins per cell. Also some researchers consider SIFT to be a HOG-like descriptor.

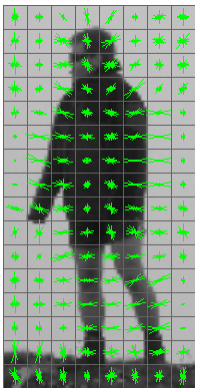

Figure 3.6: HOG descriptor orientation histograms overlaid over a person image. Image source:`http://www.juergenwiki.de/work/wiki/doku.php?id=public%3ahog_descriptor_computation_and_visualization`

- although LBP was initially proposed as a texture descriptor (global), it has been used [26] with dense extraction as a local descriptor in retrieval.

### 3.1.2 Coding and pooling schemes

In retrieval intermediary representations collect low-level feature data from the document and compute an expression of the document by aggregating the low level features into a discriminative vector that can be later classified.

A Bag of Words is a histogram of occurrence counts of a vocabulary of local image features. In computer vision and CBIR, the model is known by the name Bag of Visual Words, and is a very common technique to represent feature descriptors without modeling geometric structure. The technique is rather simple. Let's assume the feature size is $N$. By randomly sampling a sufficiently large number of features $M$ from the feature space, one can construct a "visual vocabulary" (or "codebook") of size $k$, which contains $k$ representative features ($k$ vectors of size $N$). In practice, K-means clustering on the $N \times M$ feature matrix is used for this purpose, with a value of $k$ chosen anywhere between hundreds to hundreds of thousands (depending on available time and memory). The K-means++ [85] seeding technique is

often used in practice as it leads to better codebook representation. The visual words will be the resulting $k$ cluster centroids. When representing a document with BOW, every feature extracted from the document is checked against the codebook and is assigned to the closest codeword. Many distance measures have been used but the best performing remains the simple Euclidean distance. The final feature vector for the document is the $k$-sized occurrence count histogram of every codeword.

The quantization step in Bag of Words is fast but the quality of the representation has some weaknesses. For instance a feature that is very close to the centroid will have the same effect as one that is almost at the edge of the quantization zone (that is a Voronoi cell), which in terms of representation is an information loss. Also, a feature that is at the frontier of 2 quantization zones (i.e. halfway between two neighboring codewords) will switch assignment between two codewords under the effect of noise. These issues are easily solved with soft assignment. [86] Instead of having the feature contribute with a weight of one to the nearest codeword and zero to the others, in soft assignment the weight contribution is a function of distance: close centroids have high weight, far centroids have low weight. Not all codewords need be assigned; soft assignment can be performed on the nearest $k$ centroids as well.

One particular popular update on BOW is the inclusion of spatial pyramids by Lazebnik et. al. [27], which brings a degree of spatial structure to BOW by dividing progressively the image in subimages. Performance is reportedly higher but it exponentially increases feature size (from $N$ to $N * \frac{4^{k+1}-1}{3}$ for $k$ pyramid levels).

More recently, Fisher vectors [28] have gained popularity in TRECVID. This technique constructs the visual vocabulary where each visual word is a Gaussian distribution instead of a cluster centroid. Each sample is soft-assigned to all gaussians, which leads to a dense feature vector (rather than sparse for BOW). The dimensionality of the representation is $k * N$ which makes this approach difficult to scale to TRECVID.

Another evolution of BOW seen in TRECVID is the super-vectors [29]. This approach generalizes the soft assignment to the entire codebook and expands the vector representation to dimensionality $k * (N + 1)$, with one weight constant and $N$ distances for each codeword.

Tested by [30], these methods have shown to improve precision in image classification at the cost of higher dimensionalities. Their application in TRECVID is slowly becoming commonplace, yet the time complexity is an acknowledged hurdle. While it is confirmed they improve results by some

measure, the simplicity and speed of BOW along with our somewhat lackluster hardware forced us to stick to BOW in our experiments.

### 3.1.3 Classification

In the context of this work, concept classification is a supervised learning problem, wherein for every concept there are learning labels for some of the training videos. The labels are binary and in practice are made through annotation by humans. For instance, in TRECVID, concept labels are made by a collective human annotation effort, where annotators (usually volunteers participating in TRECVID themselves) inspect a selection of frames representing shots and manually assess the presence of the annotated concept.

#### Support Vector Machines

Support Vector Machines, as proposed initially by Vapnik et. al. [31] are supervised classifiers that find a separation hyperplane between positive and negative samples and maximize the margin between the hyperplane and the features. SVM learning is the *de facto* standard in large scale image classification, a fact which has carried on over to video retrieval in TRECVID.

Instead of using dot product $k(x,y) = \sum_i x_i y_i$ as distance measure between samples, later techniques adopted the use of non-linear kernels. Some of the most popular SVM kernels in Semantic Indexing are:

- Gaussian RBF kernel: $k(x,y) = exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$ or $k(x,y) = exp\left(-\gamma||x-y||^2\right)$
- Exponential $\chi^2$ kernel: $k(x,y) = 1 - \sum_i \frac{(x_i-y_i)^2}{\frac{1}{2}(x_i+y_i)}$
- Histogram intersection kernel: $k(x,y) = \sum_i min(x_i, y_i)$

One particular shortcoming of SVM classification is in choosing the SVM and kernel parameters. There are some rules-of-thumb: the $\gamma$ parameter of the exponential kernels can be set to 1/A, where A is the average of the distance matrix of the training set, $C$ generally produces decent results when set to 1. Many researchers (including us) still resort to grid-searching for parameter combinations (where parameters are searched for in logarithmic increments), which further slows down the learning process but significantly improves precision over hand-picked values.

A technique worth mentioning is the Homogeneous Kernel Map [65] that allows a mapping of the features to a higher dimensional space, such that

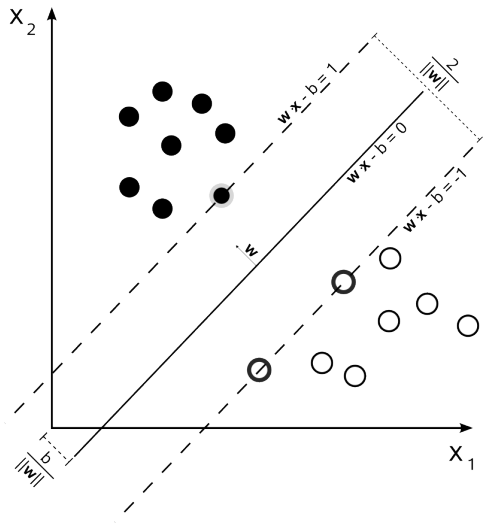


Figure 3.7: Maximum-margin separating hyperplane. In this (famous) 2D example representation, the classes are white and black dots and the separating hyperplane is the solid line. The two white and one black highlighted points along the class boundaries are the support vectos for the linear SVM. Note that in this case the constraints are perfectly satisfied, giving a perfect solution (the points are separable)

classifying the new features with a *linear* kernel (i.e. much faster) approximates the results of classifying with a non-linear kernel (higher accuracy). We have successfully used this technique on our Z* features in chapter 6.2.

There are plenty of SVM implementations available online. A popular choice among TRECVID participants is LibSVM [63], which uses the SMO algorithm as solver. LibSVM is integrated in larger machine learning packages, which we have used in our experiments: Shogun [87] and Scikit-learn [66]. For the linear case there is Liblinear [67] included in the previous packages, and the OCAS [88] fast solver implemented in Shogun.

### KNN classifier

Nearest neighbor classifiers are sometimes used in situations when the classification problem is too big to allow a cost-effective classification with more complex methods. A KNN classifier simply searches for the K training samples nearest to the test sample. From the pool of K neighbors, the test label is determined by label voting. The nearest features are found according to

some distance measure, typically euclidian. For it's simplicity and speed some indexing systems use this classifier as backup.

### Gaussian Mixture Model classifiers

The expectation step from Expectation-Maximization used in GMM learning can be used to predict the label of a new sample based on the expected weights of each Gaussian in the mixture. This procedure can effectively turn GMM into a classifer. In some scenarios, where the underlying number of prior classes is known (such as in separating speech from non-speech in audio features) GMMs classification can be used effectively.

### Random Forest classifiers

Random forests are classifiers that construct many decision trees at training time. The decision trees are constructed from randomly drawn features, and the split threshold is chosen to maximize margin along a random subset of the features, usually by maximizing mean information gain. The output of the forest is determined by class voting of each tree.

### Deep Belief Networks

Convolutional Deep Belief Networks are hierarchical generative models based built by stacking layers of Restricted Boltzmann Machines. The 'convolutional' means that weights between hidden and visible layers are shared among all locations in the image. Between the RBM layers there are pooling layers that shrink the representation. The pooling is achieved with techniques that extend the idea of max-pooling in feedforward networks to both top-down and bottom-up learning. Unlike the previous methods presented, convolutional deep belief networks do not rely on manually designed features such as SIFT. Instead, low level convolutional filters are automatically learned by training the network with unlabeled data. An interesting aspect of DBNs is that the topmost layer performs a softmax function to map the concept distribution. This effectively means that the network learns all the concept at once, instead of requiring binary one-vs-all classifiers. Recently deep belief networks have topped in classification challenges such as ImageNET [73] LSVRC 2012 and TRECVID SIN 2013[4]

[4] `http://www-nlpir.nist.gov/projects/tvpubs/tv13.slides/mediamill.tv13.sin1.slides.pdf`

### 3.1.4 Fusion

Combining classifiers is an important component of a multi-feature indexing system. A good fusion strategy can exploit the complementarity of features. The fusion of several classifiers can be performed in several places on the classifier pipeline (displayed in figure 3.2). *Early fusion* or 'feature-level fusion' is performed after the feature extraction phase and before classification. Typically it involves concatenation of feature vectors into a single large vector for training. *Late fusion*, 'decision fusion' or 'score fusion' uses the decisions of individual classifiers to predict a final decision for a test sample. Arguably, there is also 'intermediate fusion' at the coding and pooling stage, but in the literature it is mostly considered early fusion. In this section we will refer to fusion participants as modalities, even though the base features do not necessarily originate from different modalities.

#### Early fusion

Concatenating feature vectors from different modalities into a supervector is a straightforward fusion strategy that has been employed in combining audio and video features [32]. Snoek et. al. [33] has shown that early fusion performs better than late fusion in 6 out of 20 concepts. Intuitively fusing at the feature level is exploits the correlation between modalities to the maximum. Some of the obvious advantages of feature level fusion are:

- There is only one classifier to be learned and optimized, as opposed to one classifier per feature in late fusion.
- In Bag of Words style classifiers the relations between different modalities is preserved, in contrast with intermediary level fusion where correspondence between modalities is lost in the bagging process.

Some notable disadvantages of early fusion:

- The increased dimensionality for the learner; concatenating features may result in a super-feature that is too big to efficiently classify
- The need for a robust normalization method in order to harmonize the different features; failure to do so can result in components of one modality dominating the superfeature, which would render the other modalities useless.
- The choice of classifier may not be straightforward; some classifiers tend to work better in learning visual data while others work better in

text, audio, etc. Also, a classification method that works well on the modalities does not necessarily work well on their concatenation.

### Late fusion

Fusion performed after classification is referred to as late fusion. For each modality a separate classifier is trained. The decisions of the classifiers (scores/confidence values/probabilities) are used to give a final decision. This type of fusion is much more popular with TRECVID participants for several reasons:

- The decision fusion per se is a much smaller computational problem than early fusion. Instead of having to classify $N_{samples} * N_{modalities} * dim_{modality}$ we only work with $N_{modalities}$ vectors of $N_{samples}$, given that $dim_{modality}$ can be in the tens of thousands. If the scores/decisions for each modality are given, even brute force search approaches are feasible. This is the idea behind the weighted linear fusion that we mostly employ in this thesis, described in detail in annex B.

- Classification of the modalities can be separated from the fusion, allowing better tuning for each modality classifier. Also, computation-wise training $N_{modalities}$ classifiers with features of size $dim_{modality}$ is generally faster than training one classifier with superfeatures of size $N_{modalities} * dim_{modality}$.

The shortcoming of late fusion is that correlation between modalities is lost, which makes fusion results harder to explain. Using multiple classifiers and then fusing can lead to cumulated effects of bad classification like overfitting.

There are several ways to perform score fusion: the most simple are the score pooling methods where a simple function (sum, max, average, geometric mean, etc.) is used on the modality scores in order to obtain the final score. Better results are achieved with weighted linear fusion, where the function is a linear combination of scores [13]. The weights of this combination are typically learned by score optimization through cross-validation. Others use a meta-classifier on the scores and train it on validation data e.g. Adams et. al. [89]. More complex methods such as rule-based decision have been developed. For example, Strat et. al. [90] use custom defined rules to combine decisions of dozens of multimodal classifiers for video concept detection. This is done by studying the effect of hierarchically clustering classifiers following different strategies, including manually grouping based on modality, descriptor, classifier type, etc.

#### Intermediate level fusion

In the particular context of BOW methods, fusion can be performed at the coding stage. In chapters 6.1 and 6.2 of this thesis new features constructed by BOW concatenation from different features are studied. Another type is the kernel-level fusion from Wang et. al. [48], where features are directly mixed as channels in the SVM kernel. A way to overcome the aforementioned shortcomings of early fusion is to combine different kernels, known as Multiple Kernel Learning (MKL) [91].

### 3.1.5 Indexing datasets

From a data perspective, Semantic Indexing in today's state-of-the-art is interesting to our theme in two instances: first in the much richer literature surrounding CBIR and second in TRECVID SIN. We will present in parallel several CBIR datasets and an example of well performing method.

#### Pascal VOC 2012

The latest (and last) edition of the PASCAL Visual Object Classes [92] showcases an 'Object classification' competition. The goal is to correctly predict the presence/absence of an example of object in the test image. Unlike the detection competition, the position of the object is not relevant. In this regard, we can see this problem as a subset of concept detection (remember that in section 2.1 we discussed about objects being possible semantic concepts). The dataset contains 11,530 images and 20 object classes. The second best performing system [93], with a mean average precision of 78.6% uses the following low level features: SIFT [17], LBP [9] and HOG [23]. Sampling of these features is done both densely and with interest point detection. Additionally patch level features based on the image segmentation are computed based on multiple segmentation methods with multiple parameters. Bag of Words with spatial pyramid [27] is used for achieving better localization for low level histogram features. Two kernels are used in classification: a $\chi^2$ distance based kernel for low level histogram features and the RBF kernel for patch level features. Classification models are learned with kernel SVM.

#### ImageNET

The ImageNET Large Scale Visual Recognition Challenge[5] follows the same philosophy as PASCAL VOC but with a considerably larger dataset. There

[5] `http://www.image-net.org/challenges/LSVRC/2013/index.php`

are 200 object classes, and there is a separate image classification challenge with 150,000 photographs and 1000 object categories. In both 2012 and 2013 editions the best performing systems that did not use external training data according to the results page[6] were a deep convolutional networks. The winning network in 2012 [73] consisted of five convolutional layers and three fully connected layers, counting 650,000 neurons and 60 million parameters (see figure 3.8). Input images where whitened by pixel value and downsampled to $256 \times 256$ pixels.

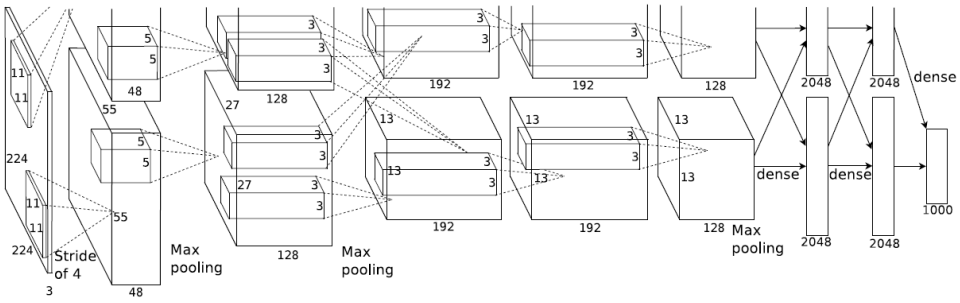


Figure 3.8: Overview of Krizhevsky et.al. [73] convolutional neural network used in ImageNet LSVRC contest.

### TRECVID SIN

The TRECVID Semantic Indexing dataset is the standard dataset for the experiments in this thesis. For computational reasons, the experiments on TRECVID are performed on the 2010 collection, with the exception of Z* features which use the 2012 edition data (see table 2.1 for a quantitative comparison). The data for the 2010 edition of the Semantic Indexing TRECVID contains 400 hours of user-submitted copyright free Internet videos. As a consequence, the dataset is extremely heterogeneous, with very high intra-class variability. There is also a high amount of camera motion, and uneven video resolution, compression quality and overlaid graphics. See sections 2.1 and 2.4 for discussions on this dataset.

Automated shot boundary detection reveals that this represents 119,685 development and 146,788 test shots. The evaluation campaign requires the submission of 346 concepts. However because of time and computation limitations the 50 concepts from the "Lite" SIN task [4] were evaluated in our

[6] `http://www.image-net.org/challenges/LSVRC/2012/results`

experiments. These shots however are incompletely annotated, which means that only a subset of variable size of the training set will actually be used in training, and that classification can only be verified on a subset of testing shots. Performance measure is the mean average precision across all concepts, but for evaluating challenge submissions the inferred mean average precision (infMAP) is used. Because the testing set is so large, mean average precision has to be estimated by manual evaluation of a sampling of a submitted list of 2000 shot candidates. This creates a pool of new labellings that acts as ground truth. Non-evaluated shots are assumed negative (possibly leading to MAP underestimation). We use the 2010 dataset because of the smaller size and because evaluations are more robust since subsequent editions use the 2010 collection as development data, therefore more densely annotated.

A representative example of a TRECVID SIN system is IRIM's submission for the 2012 edition [12]. The system is a fusion of 128 descriptors contributed by participants to the IRIM group. Out of the 128, we exemplify the following. In their fusion system there were at least 19 descriptor variants of SIFT [17] into 5 categories: a 3-level dense spatial pyramid, a standard Bag of Words, opponent SIFT generated by dense and interest points, color SIFT from superpixel segmentation and the biologically inspired Opponent SIFT BOW with retinal preprocessing [94]. There are also three appearances of HOG [23] extracted with varying window sizes, aggregated in a VLAT [95] scheme. There is one update of Color LBP [9] that produces more compact representations called OLBPC [14]. One dynamic feature named 'faceTracks' is described as "OpenCV + median temporal filtering, assembled in tracks, projected on keyframe with temporal and spatial weighting and quantized on image divided in $16 \times 16$ blocks", however no other details or references are available on this feature. There are also four variants of STIP [42], combinations of HOG and HOF as descriptor with Bag of Words of size 256 and 1000. All features are optimized by a power transformation (which normalizes histogram components across features) and PCA dimensionality reduction. Classification is either done by k-nearest neighbors or by multiple learner SVMs. The computed MAP of the faceTracks descriptor is 0.0191, considerably lower than for image descriptors (dense SIFT is in general $>0.1$). The performance of STIP on the test set is missing.

Another example is Informedia's 2011 system[7]. The submission for SIN contains three low-level features: SIFT, Color SIFT and MoSIFT [46]. It is

[7] http://www-nlpir.nist.gov/projects/tvpubs/tv11.papers/cmu.pdf

not uncommon for TRECVID systems to use exclusively SIFT features.

## 3.2 Spatio-temporal based Action Recognition

Most of the research on spatio-temporal video description has originally been designed for recognizing human motion (for a comprehensive survey on action recognition see [2] and [34]). This is motivated by the fact that human motion is more limited in range and variability (for example, the KTH Human action dataset - widely used as a testset in human action recognition - contains only six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping). For the same reason, training datasets can be much smaller than for the general case. On the other hand, action recognition systems may impose more constraints on the video data (e.g. extracted background, defining an ROI), yet some other systems may work on noisy surveillance videos. A particular feature that has been extensively exploited in recognition is the repeatability of human poses (for instance, in walking). Spatio-temporal descriptors have been successfully used in human action detection/recognition with applications in video surveillance, video indexing, human-computer interaction and others [2]. A significant portion of research in this field is specific to human body actions: gestures, facial expressions, full-body movement. However, many of the vision-based techniques built for- and tested on human action recognition are not limited to human motion, and therefore suitable for the task of video classification.

### 3.2.1 Spatio-Temporal Global Descriptors

One of the earliest spatial-based action representations, a.k.a. "holistic representations" are implicit global models. Their global nature implies the homogeneity of the representation, which is to say that every point in the XYT space is treated and processed in the same manner from the other. For this reason, image-based representations generally are computed in grid-based or dense manner, which is computationally efficient and highly parallelizable. Defining an ROI centered on the person is required by these techniques. Provided some pre-processing of the video such as foreground-background separation for silhouette or contour extraction, MHIs (Motion History Images) or MEIs (Motion Energy Images) [35] can be used to characterize simple human motion such as those in the KTH dataset. Silhouette extraction is subject to noise, so special solutions have been developed, such as the use of Chamfer distance [96] for silhouette matching and shape context

descriptors [97]. Motion History Image are the basis for the BOMT method we define in chapter 5.

Efros et. al. [36] compute optical flow densely in person-centered sequences and decompose flow direction into 4 directional channels. See figure 3.9 for an example.

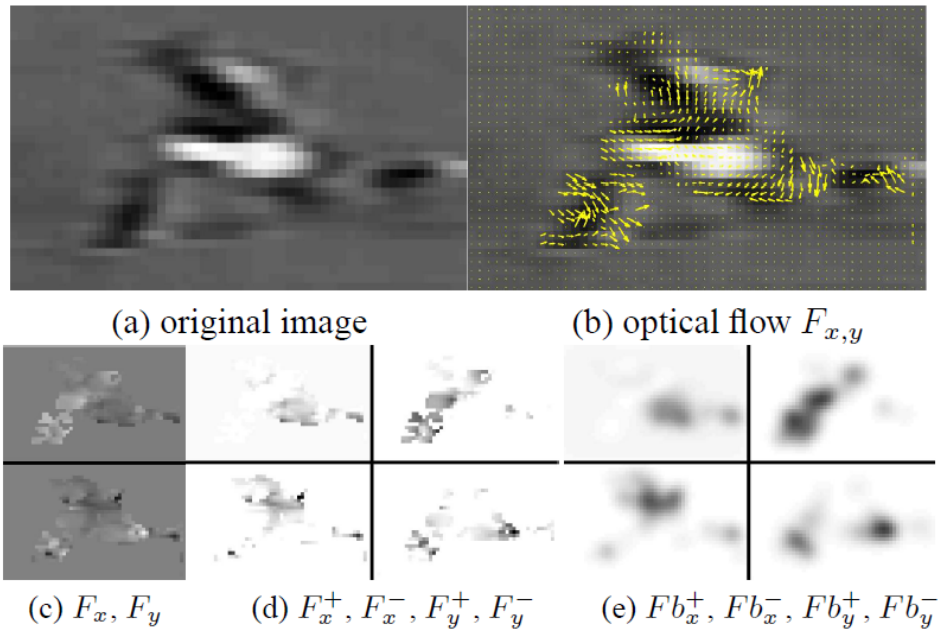


Figure 3.9: Efros et. al. [36] recognize the motion of a 30-pixel tall football player by separating horizontal and vertical optic flow components into positive and negative channels and blurring the resulting motion channels.

Other later techniques work directly with the evolution of the silhouette in the space-time volume. Blank et. al. [37] temporally stacks silhouettes into a 3D object then recognizes them using space-time saliency and orientation. Ke et. al. [38] compute volumetric features from dense flow and sample the horizontal and vertical components in space–time. Other similar space-time volume approaches are [39–41].

### 3.2.2 Spatio-Temporal Local Descriptors

The type of spatio-temporal feature particularly interesting to the concept detection task is the local variety. What these approaches share in common

is that similarly to the 2D local features described in 3.1.1 they consist of a *detector* that finds keypoints in the 3D volume, and a *descriptor* which characterizes the local spatio-temporal neighborhood of the keypoint. Systems using these features often inherit from CBIR the Bag of Words (called "Bag of Visual Words" in CBIR) and binary classifier learning framework. We will now present some space-time detectors and descriptors used in action recognition and relevant to our field. Some of the work in this section is based on Wang's evaluation paper [20]. For the performance of these descriptors on the action datasets, see table 3.1.

The following is a list of notable spatio-temporal detectors using in action recognition.

- Laptev and Lindenberg propsed in 2003 [42] the idea of space-time interest points (STIP) by extending to 3D the well known 2D Harris corner detector, effectively detecting space-time corners. The detector works by first smoothing independently in space and time using Gaussians. A second-moment matrix $\mu$ of the space-time gradient neighborhood of each point is computed. The interest points are given by local maxima of the function $det(\mu) - ktrace^3(\mu)$. The interest point detector was updated in [98] with invariance to background translational motion.

- Dollar et. al. [43] detects interest points by finding maximal responses of 2D Gaussian spatial $\times$ 1D Gabor temporal separable filters applied on the 3D volume. Initially tested on animal activity, the temporal Gabor filtering proved suitable for detecting repeatable motion.

- 3D Hessian extension was proposed by Willems et. al. [100]. This detector finds 3D blobs by maximizing the determinant of the $3 \times 3$ space-time Hessian matrix. Similar to the other methods, scale selection is done by computing the response over several octaves of scale and performing non-maximum suppression. This is however the only approach where the scale space search is done for both space scale and time scale jointly.

- Dense extraction. The whole idea of an interest point detector has been challenged in recent years by Wang et. al. [20]. Their experimental findings show that in the setting of realistic video dense extraction of features seems to outperform spatio-temporal keypoint extraction, much like in 2D image retrieval. Dense sampling does however bring

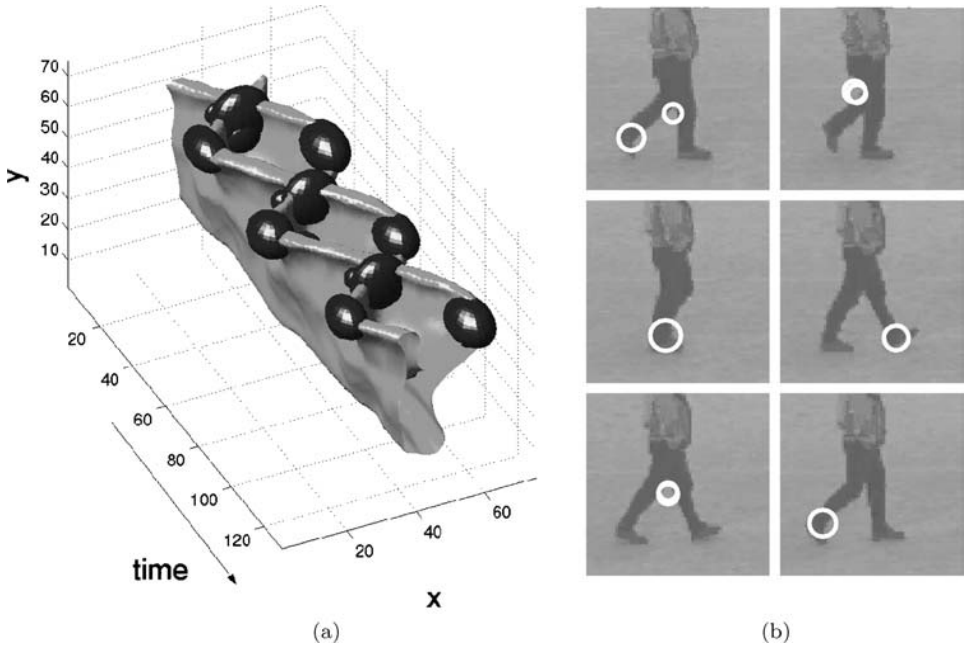


Figure 3.10: Example of space-time corners extracted from a walking sequence. (a) shows the evolution of the leg shape (upside down) and the detected extrema points as ellipsoids. (b) the interest points as they appear in their corresponding frames. Image source: [42, 99]

an increased feature count per shot, which manifests in heavier extraction computation. Dense trajectory based features were introduced by Wang et al. [48] as an alternative to XYT based interest point based detectors like STIP. Inspired by the improvement in results in image classification using dense sampling over sparse sampling, they proposed using densely sampled trajectories produced by KLT trackers as features rather than sparsely sampled trajectories.

The spatio-temporal descriptors represent the space-time neighborhood of the spatio-temporal interest point. Here are some important descriptors:

- The descriptors studied by Laptev et.al. [42, 98] are single- and multi-scale N-jets, histograms of optical flow, and histograms of gradients [77]. N-jets are spatio-temporal Gaussian derivatives of the cuboid up to the N-th order. Histograms of gradient and flow are computed akin to the cells in the classical HOG [23] with the difference that they

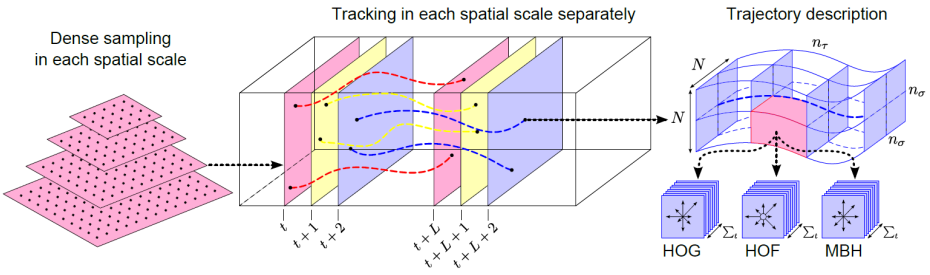


Figure 3.11: An overview of Wang et. al. [48] Dense Trajectories descriptor: from every spatial scale grid based tracking is performed for L frames. Descriptors are extracted from a local neighborhood centered on the trajectory and divided into a spatio-temporal grid of size $n_\sigma \times n_\sigma \times n_\tau$

take into consideration the 3D space-time neighborhood of the point. These histograms are built by subdividing the volume in cells (typically 3 spatial $\times$ 2 temporal cells) and computing a 4-bin HOG and a 5-bin HOF in each cell. A concatenation of the last two named HOGHOF has become the standard way of extracting STIP data.

- In Dollar's paper [43] three local descriptors are proposed in the paper: based on normalized pixel values, brightness gradients, and windowed optical flow, out of which the gradients give the best results when detecting actions on KTH.

- A 3D extension of the HOG descriptor was proposed by Kläser et. al. [44]. The idea is that instead of projecting the direction of 2D gradient on directional bins like in HOG, 3D gradient is projected on the N faces of a regular polyhedron, thus quantizing it into an N bin histogram. The detector is a 3D Harris similar to STIP. In section 4.1 we experiment with this descriptor on the TRECVID dataset.

- In Wang's [101] method HOG, HOF and MBH descriptors are extracted from cells in the trajectory tunnel and aggregated temporally. Motion Boundary Histograms (MBH), formalized in [102], are oriented histograms obtained from the two components of the gradient of optic flow. As the name implies these features capture high response zones along the frontiers of moving regions. While this approach outperformed the state-of-the-art on major datasets (Youtube, Hollywood2, UCF Sports), the computational cost remains very high.

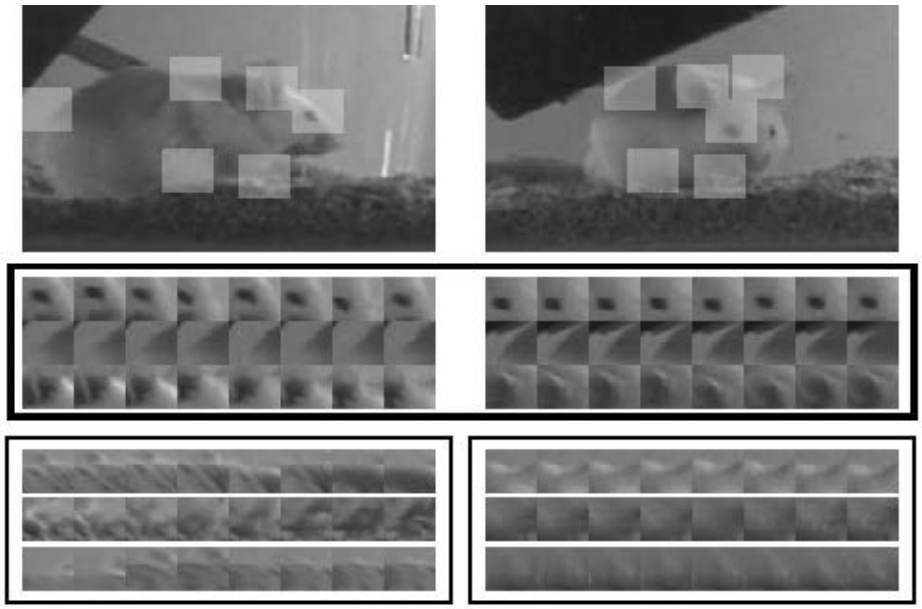

Figure 3.12: Examples of flattened space-time cuboids from Dollar's paper [43] extracted from two sequences displaying the same action. The first three cuboids (black outline) for each sequence are similar.

- Paul Scovanner et. al. [45] propose a 3D extension of the well known SIFT descriptor. Conceptually it is similar to the initial steps of HOG3D, but with gradient orientation quantized into a 2D histogram based on polar angles (effectively dividing the sphere into meridians and parallels).

- Another extension on SIFT is Chen and Hauptmann's MoSIFT [46] which applies the orientation quantization function of SIFT on the dense optic flow field (instead of gradient orientation). Resulting feature is a concatenation of the usual SIFT and the motion SIFT.

- Another example of extending a 2D feature to 3D is the Volume Local Binary Patterns [103], which is based on the well known Local Binary Patterns [9] by Ojala et. al. Basically, around an interest point, three orthogonal planes that cross the point are considered (XY, XT, YT), LBP is computed in each of the planes and the final feature is a

concatenation of the 3 histograms of patterns generated by LBP.

### 3.2.3 Action Recognition Standard Datasets

In this section 4 video datasets used extensively in this work are described. KTH, Youtube, UCF Sports and Hollywood2 are video collections built for recognizing human actions and some researchers [20,48] have evaluated their descriptors on all 4 datasets in parallel. Action recognition datasets are considerably smaller than TRECVID (as seen in table 3.1) and the number of classes is significantly lower. Also, the defined action classes have much less variability than TRECVID concepts.


Figure 3.13: Sample frames from the 4 action recognition datasets. From top to bottom: KTH, Youtube, Hollywood2, UCF Sports. Source: [101]

Using the camera motion compensation algorithm presented in annex A, we were able to compute some global statics concerning camera motion. The simplest of these statistics was the percentage of frame transitions without camera motion (transitions where the transformation homography is very close to identity), which we estimate at less than 24%. With the exception of KTH, we have found this percentage to be surprisingly consistent across the datasets.

### KTH

The KTH dataset [68] first mentioned in 2004 contains six classes of human actions: walking, jogging, running, boxing, hand waving and hand clapping. There are 25 subjects, each person performs the same action four times under four different scenarios: outdoors, outdoors at a different scale, outdoors with camera motion and indoors. Background is mostly static and homogeneous. There is a total of 599 sequences of $160 \times 120$ resolution. We follow [69] in performing leave-one-out cross validation to evaluate our approach. Leave-one-out cross-validation uses 24 subjects to build action models and then tests on the remaining subject. Performance is reported as the average accuracy of 25 runs. In spite of its age and simplicity, the KTH dataset is still used to test video analysis methods, as seen in table 3.1. It is only very recent work that finally reports perfect accuracy on KTH. [74]

### Youtube Actions Dataset

The YouTube dataset [104] contains 1600 videos into 11 action categories: basketball shooting, biking, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking a dog. This dataset contains real-life Youtube videos and is therefore consistently more challenging than KTH: it exhibits abundant camera motion, background clutter, extremely variable illumination and object size, compression artifacts. An evaluation method similar to the previous is used, with leave on out cross validation for a predefined set of 25 folds [104]. Average accuracy over all classes is reported as performance measure.

### UCF Sports Dataset

The UCF sport dataset [105] has 150 video samples containing ten human actions: swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking. Intra-class variability is high, but video resolution and quality are an improvement over Youtube. We follow [48] and augment each training set with horizontally flipped copies of the training videos. Classification is done by a leave-one-out strategy: on each fold, one video is used to test the 149 others plus their flipped versions are used as training.

### Hollywood2

The Hollywood2 dataset [106] has been collected from 69 different Hollywood movies. There are 12 action classes: answering the phone, driving car, eating, fighting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up. In contrast with the previous datasets, Hollywood2 contains high quality, professionally filmed video with long shots and regular camera movement. There is however more variability in each action because of closeups, framing. The higher resolution means that there are more scales where the action can happen. The dataset is divided between the training (823 sequences) and testing sets (884 sequences), which are shots extracted from movies. All shots from one movie are either one the training or the testing set, never across. The performance is evaluated by computing the average precision for each of the action classes and reporting the mean AP over all classes.

## 3.3 Space-time features in large scale concept detection

In the 1990s and early 2000s, along with the rise of Internet video, a large number of video retrieval techniques have been proposed. A survey paper by Ren et. al. [47] attempts to organize and classify many of these techniques. Although interesting, most of these approaches to retrieval have not generated consistent research interest later on. With some exceptions (most of them presented below), spatio-temporal features have been very rarely implemented in practical concept detection systems and almost never on TRECVID. One of the causes is obviously the heavier computation. Instead, the trend in TRECVID high-level feature extraction [3–5] has been the use of usually one, sometimes several keyframes per shot, from which image descriptors are extracted. However, hybrid approaches are sometimes successful: MediaMill [58] sample several keyframes in one shot and obtain top performance. However, how much is due to the multi-keyframing strategy is unknown.

In 2012 we have seen in TRECVID SIN a few instances of dynamic descriptors.

- Kobe university [107] achieves top performance in the Light task; their run includes Wang's dense trajectories [48] used to define 30 dimensional trajectory displacement and HOG around the trajectory. Feature representation is done by first reducing dimensionality with PCA,

then coding by GMM supervectors. Training is done with an RBF kernel SVM and fusion is achieved by weighted linear combination.

- Informedia extracts MoSIFT [46], along with classical SIFT and Color SIFT.
- IRIM [12] and Quaero extract STIP and 'faceTracks', as described at the end of section 3.1.
- UEC employ a SURF-derived spatio-temporal descriptor [49]
- Florida International University and University of Miami [50] use the HOOF from [51].
- GIM [52] team use their own motion feature based on motion direction in 5 regions of the frame (corners plus center).
- Image features are extracted from multiple frames in a shot by IBM [53] (at a rate of 0.25fps), PicSOM and MediaMill. Also, ITI-CERTH uses slices in the space-time volume called tomographs, used in a similar way with the keyframes.

To summarize, out of the 25 participating teams in 2012 SIN, only 7 use actual space-time features, compared to at least 23 for SIFT.

Recognition by local ST features is still very slow. The timeline in figure 6.9 and a quantitative study in section 5.5.1 compares STIP, MoSIFT and Dense Trajectories extraction time to one of our proposed TRECVID descriptors. Results are reported for the KTH dataset, as most of these features can only be efficiently computed on such smaller datasets. Of all the sparse features presented above, only these three have been reportedly computed on TRECVID SIN by teams disposing of considerable hardware resources. However for a team disposing of a reasonably sized cluster, the computational argument alone is sufficient to render methods like Dense Trajectories highly impractical for application on TRECVID SIN, lest some optimization is brought to the code. With today's hardware, the TRECVID concept detection problem requires more emphasis on speed than on precision, which is the opposite of the descriptors presented here.

| | Action Recognition | | | | CBIR | | |
|---|---|---|---|---|---|---|---|
| | KTH | UCF Sports | Youtube | Hollywood2 | PASCAL VOC 2012 | ImageNET ILSVRC 2013 | TRECVID 2012 |
| videos/sequences | 599 | 150 | 1600 | 60 | n/a | n/a | 28123 |
| samples/shots/keyframes | 599 | 150 | 1600 | 1600 | 11530 | 456182 | ∼546000 |
| average resolution | 160x120 | 720x480 | 320x240 | 336x610 | 469x387 | 482x415 | 240x335 |
| database size | 1.1 GB | 1.7 GB | 424 MB | 15 GB | ? | ? | 200 GB |
| total playback time | 50 min | 13.9 min | 2.81 hours | 2.93 hours | n/a | n/a | 800 hours |
| number of classes | 6 | 10 | 11 | 12 | 20 | 200 | 500 |
| average number of training samples per class | 96 | 298 | 1535 | 823 | 5717 | ? | 59525 |
| STIP [20] | 92.1% [20] | 82.6% [20] | 59.1% [64] | 47.4% [20] | n/a | n/a | ? |
| Dense Trajectories [48] | 94.2% [48] | 88.2% [48] | 84.2% [48] | 58.3% [48] | n/a | n/a | |
| MoSIFT [46] | 95.8% [46] | | 63.1% [46] | | n/a | n/a | 2.27% |
| HOG3D [44] | 92.7% [44] | 71.2% [44] | | 48.6% [44] | n/a | n/a | |
| ST-SIFT [108] | 90.7% [108] | 80.5% [108] | | | n/a | n/a | |

Table 3.1: Video and Image retrieval dataset comparison and performances of known features

# Chapter 4

# Baselines and Global Approaches

We have discussed in the previous chapter that one way to categorize content description features in the literature is by scope: either a feature is built by analyzing the entire document (e.g. color histogram of an image) and is named *global* or only a small well-defined subsection of it (e.g. SIFT [17] descriptor for a $16 \times 16$ pixel region around a keypoint detection), known as *local*. Local approaches are more general and more suitable for image to spatio-temporal extension, a fact testified by the state-of-the-art. Some examples of 2D to 3D extensions are: of SIFT [17] into 3D-SIFT [45] and MoSIFT [46], HOG [23] into HOG3D [44], LBP [9] into LBP-TOP [103].

In image retrieval global approaches are both computationally lighter and less accurate than local features. In some cases, such as for the edge histogram [24] defined in the MPEG-7 standard, the speed/accuracy tradeoff seems advantageous, especially for large datasets. [25] In comparison to the local approaches, very few examples of global spatio-temporal extensions can be found [54–56] with very limited experimental proof. The aforementioned speed and the lack of global features in 3D make up the motivation of our approach in this chapter.

The literature offers no standard baseline in video retrieval for the spatio-temporal. In section 4.1 we experiment with the HOG3D local descriptor on TRECVID data in order to produce a spatio-temporal baseline for our

classifiers. Since the HOG3D descriptor did not outperform the traditional TRECVID keyframe approach, we extended the SIFT-BOW classifier to multiple keyframes per shot in section 4.2, also in search for a suitable baseline. The main contribution of this chapter is in section 4.3. ST-MP7EH is a spatio-temporal extension of the global 2D edge histogram descriptor from MPEG-7. We conclude the chapter with a global analysis on shot motion. In section 4.4 the relation between concepts and characteristic camera movement detected in TRECVID videos is studied.

## 4.1 HOG3D as a baseline

Combining the idea that SIFT-BOW is a successful solution in image retrieval and HOG3D is the video counterpart of the SIFT-like HOG, the experiments in this section aim to build a HOG3D-BOW spatio-temporal baseline.

As stated earlier, there is no known spatio-temporal content description method recognized by the community as state-of-the-art baseline. Interestingly, in CBIR one such baseline definitely exists: the combination of a local image descriptor and an intermediate document representation model. In practice, this is most of the time SIFT and BOW. Such a baseline is present in many TRECVID concept detection systems that use keyframes, including EURECOM's. These systems are effectively performing image classification, since they train classifiers on keyframe images and test them against a keyframe extracted from the test shot. Of course, this approach is completely oblivious to any motion present in the shot, or any other spatio-temporal information.

The HOG descriptor is a SIFT-like visual descriptor that has successfully been applied to human action recognition [23]. The method is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. As with SIFT, the idea is that local image appearance and shape can be effectively characterized by the distribution of local intensity gradients or edge directions, which can be done without precise knowledge of the corresponding gradient or edge positions. This is implemented by dividing the image windows into small cells, and accumulating a local histogram of gradient directions over the pixels of the cell. The representation is given by the combined histogram entries, which is usually contrast-normalized for illumination/shadowing invariance.

Following the general trend of transforming a 2D image descriptor into a ST descriptor, Kläser, Marszalek and Schmid [44] have developed a 3D

version of the HOG descriptor. This local descriptor represents the region around the interest point by a feature vector, while the whole analyzed video volume is represented as a set of features computed at different scales and positions.

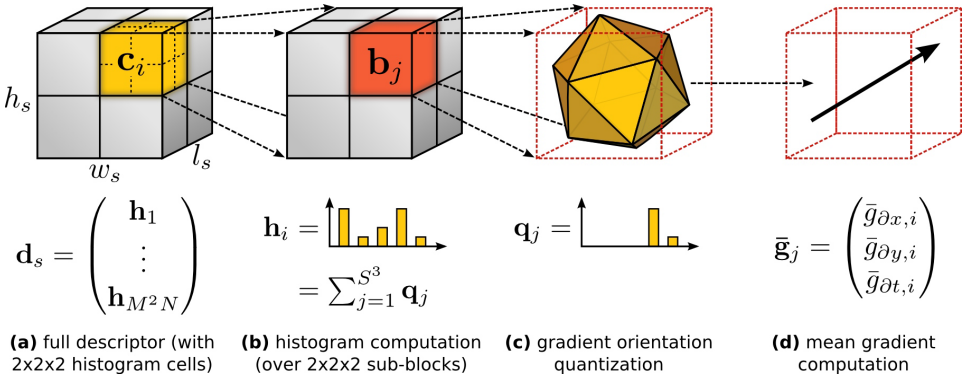


Figure 4.1: Overview of HOG3D descriptor. [44] Source: `http://lear.inrialpes.fr/people/klaeser/research_hog3d`

In order to compute the descriptor, the spatio-temporal volume is segmented into spatio-temporal pyramids that represent the region in multiple spatial and temporal scales. For each sub-block one gradient histogram is computed. This histogram is built by binning the average direction of the gradient by projecting it onto the faces of the quantization polyhedron. The polyhedral face where the projection falls marks the chosen bin. Finally, all histograms are concatenated into a feature vector.

HOG3D is a sparse descriptor, in that the process is supposed to have 2 steps. First, the spatio-temporal interest points must be identified, based on local intensity/saliency extrema or invariance to transformations, using one of the many interest point detectors. Secondly, the descriptor is calculated over the spatio-temporal volume surrounding the ST interest point. However, interest point detection over the entire shot is a computationally costly process (see STIP computation in figure 6.9). For that reason, we decided to convert the already existing 2D keyframe interest points from SIFT to spatio-temporal interest points. These points have been previously calculated for SIFT image classification on the central keyframe of the shot. We have used the popular Lip-vireo toolkit [57] with a Hessian-Laplacian detector to find at most 500 keypoints in every keyframe. Around every interest point, a space-time volume of size $32 \times 32 \times 18$ is subdivided into

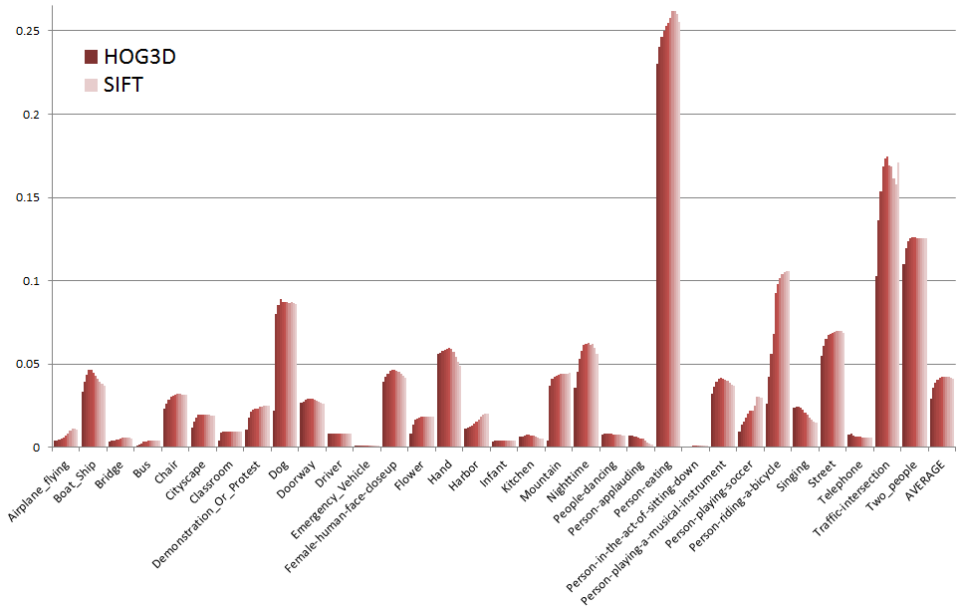


Figure 4.2: Late fusion between HOG3D and SIFT

$4 \times 4 \times 3$ blocks. Average gradient orientation in each block is quantized to the normals of a icosahedron (20 face regular polyhedron) resulting in a 20 bin histogram. The feature vector is the histogram concatenated from all blocks, resulting in a dimensionality of $4 * 4 * 3 * 20 = 960$. Code for the HOG3D descriptor is available from the LEAR team [44].

In order to obtain our BOW, we take a random subsample of 50,000 feature vectors and perform k-means clustering with $k = 500$ clusters. The centroids of these clusters make up the visual words, which correspond to the $k$ positions in the occurrence histogram. By performing a nearest-neighbor search in feature space between the analyzed feature vector and the visual words, we can count the occurrence of the visual word on the histogram corresponding to the analyzed shot. This histogram is then normalized and used as input feature vector representing the shot in SVM training.

We experiment with HOG3D on the TRECVID 2007 SIN dataset. This collection contains 36,262 shots obtained from approximately 100 hours of footage of news magazine, science news, news reports, documentaries, educational programming, and archival video in MPEG-1 format. We evaluate 32 concepts: Airplane flying, Boat Ship, Bridge, Bus, Chair, Cityscape,

Classroom, Demonstration Or Protest, Dog, Doorway, Driver, Emergency Vehicle, Female-human-face-closeup, Flower, Hand, Harbor, Infant, Kitchen, Mountain, Nighttime, People-dancing, Person-applauding, Person-eating, Person-in-the-act-of-sitting-down, Person-playing-a-musical-instrument, Person-playing-soccer, Person-riding-a-bicycle, Singing, Street, Telephone, Traffic-intersection, Two people. Just like with SIFT and ST-MP7EH (presented later in this chapter in section 4.3), we train one SVM for each concept by labeling our annotated training data with either $+1$ or $-1$ (positive or negative example). After testing the obtained models on the test data, we obtain a list of score pairs which can be seen as the probabilities that the concept appears or not in the shot. Finally, these scores are used to rank the shots for each concept in order to calculate the average precision.

In the next experiment, we took the HOG3D scores and combined them with SIFT scores in a weighted linear fusion experiment (see annex B for the detailed procedure). We search for the correct "mix" between the scores of the 2 descriptors by testing 10 equally spaced values for the weight $\alpha$ in the $[0, 1]$ interval. We found no significant increase in MAP from the fusion of the 2 descriptors, perhaps because both of them rely on the same interest points. The result of the HOG3D-SIFT latent fusion is shown in figure 4.2. On each concept, the dark-red column (the left-most column) represents HOG3D precision, the light-red (rightmost) represents SIFT, and the columns between represent classifiers mixing SIFT and HOG3D. Note the last column, which represents the MAP of the classifier: the maximal MAP of the system is at approximately the same height as SIFT.

The MAP values for the HOG3D experiment on the TRECVID 2007 dataset are presented in the following table.

| TV07 | SIFT | HOG3D | fusion |
|---|---|---|---|
| MAP | 0.041423344 | 0.028952188 | 0.042391781 |

## 4.2 SIFT multi-keyframe

In order to tune their concept detection systems some TRECVID participants [58] process several keyframes from the same shot instead of one. The idea is to increase the accuracy of the classifier by allowing the concept to appear more often and in more varied views, and having less chance to miss the concept occurrence if it is not present throughout the entire shot.

We attempted to implement this technique by using a dense temporal sampling of 1/5 (every 5 regular frames, one is selected for classifier prediction). A pooling strategy is also needed in order to calculate the score estimation for one shot from the $N$ scores of individual keyframes from the same shot. Average and maximum are obvious candidates for the pooling, but we also considered some other fusion techniques.

Using classifiers already trained on SIFT data from TRECVID 2010a (half of the 2010 development data) with BOW of size 500, we replicate the same process on the the test set, with one difference: instead of using one keyframe we sample a large number of keyframes from the shot (1/5 regular frames), compute SIFT features and create a single visual word occurrence histogram $h_{kf}$ per keyframe $kf$. This gives us $N$ histograms of size 500. These histograms can be processed by the classifiers in order to obtain $N$ individual keyframe scores $score_{SVM}$.

At this point the 3 variants of our descriptor branch:

1. *mkfSIFT1* the highest-scoring keyframe of the shot gives the shot score (a.k.a. max-pooling)

$$score_{mkfSIFT1}(shot) = max_{kf \in shot}(score_{SVM}(h_{kf})) \tag{4.1}$$

2. *mkfSIFT2* the shot score is the mean of the scores of the keyframes (a.k.a. average-pooling)

$$score_{mkfSIFT2}(shot) = average_{kf \in shot}(score_{SVM}(h_{kf})) \tag{4.2}$$

3. *mkfSIFT3* the shot score is the score of the average histogram of the shot (each bin is the average of the corresponding bin of the keyframe histograms)

$$score_{mkfSIFT3}(shot) = score_{SVM}(average_{kf \in shot}(h_{kf})) \tag{4.3}$$

The results of mkfSIFT compared with the initial SIFT keyframe method are shown in the following table:

From the results, it is somewhat surprising to discover that although there is more information contained in multi-keyframe, the MAP is lower than the traditional keyframe SIFT. As it is often the case in ranked retrieval, the difference in precision cannot be easily explained. One possible cause is a border effect: averaging many scores for one shot diminishes the influence of high scores on relevant keyframes. Conversely, scores that are erroneously high on non-relevant shots (e.g. due to classifiers learning the

Table 4.1: Mean average precision of mkfSIFT on the TRECVID 2010b dataset

| concept | mkfSIFT1 max-pool | mkSIFT2 avg-pool | mkSIFT3 avg-bow | SIFT |
|---|---|---|---|---|
| Airplane_Flying | 0.006 | **0.025** | 0.018 | 0.017 |
| Boat_Ship | 0.019 | 0.018 | **0.024** | 0.012 |
| Bus | 0.003 | **0.006** | 0.005 | 0.005 |
| Cityscape | **0.178** | 0.154 | 0.137 | 0.168 |
| Classroom | 0.008 | 0.005 | **0.009** | 0.007 |
| Demonstration_Or_Protest | 0.038 | 0.033 | **0.060** | 0.044 |
| Hand | 0.003 | 0.008 | 0.006 | **0.025** |
| Nighttime | 0.028 | **0.063** | 0.040 | 0.059 |
| Singing | 0.060 | 0.076 | 0.072 | **0.140** |
| Telephones | **0.010** | 0.003 | 0.003 | 0.005 |
| MAP | 0.035 | 0.039 | 0.037 | **0.048** |

backgrounds) are more likely to have maximal value in a shot, thus resulting in an abnormally high score.

In figure 4.3 for each of the three variants of mkfSIFT, $x$ coordinates of a point represents the score of a shot as classified by SIFT and the $y$ coordinate the score for the same shot according to mkfSIFT. The exemplified concept is 'Nighttime'. The color describes the annotation label for the shot (green=positive annotation, red=negative or missing annotation). Ideally, for positive samples mkfSIFT should produce higher scores than SIFT, which would mean green points should be higher than the red points or, at least above the diagonal of the graph. What we see instead is that the general trend of mkfSIFT is to uniformly boost the score of both positives and negatives indiscriminatingly. This effect coupled with the predominance of negative samples over the positive explains the lower MAP.

## 4.3 ST-MP7EH

Spatio-temporal descriptors have been used for various video detection tasks, most notably in human action recognition. These descriptors show very good discrimative features and are reliable in general, but have steeper computing requirements and sometimes need pre-processed video data [2, 47]. When used on general concept detection such as TRECVID, the noise, camera

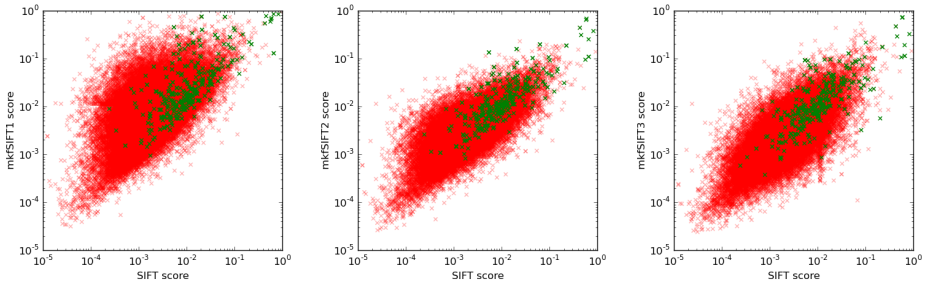


Figure 4.3: Comparison between SIFT and mkfSIFT scores for TRECVID concept 'Nighttime'

movement, and possibly bad shot segmentation can prevent spatio-temporal descriptors from efficiently recognizing video concepts. Generalizing from human motion to any dynamic feature in video is also a challenge for the ST descriptors. For these reasons, their adoption in real systems at TRECVID has been slow and with mediocre practical results [3]. The most likely cause is the comparatively high computational cost that comes with descriptors on *xyt* space, but also because of dataset problems, such as high intra-concept variability and very few positive instances of concepts.

We propose a spatio-temporal descriptor that can work around these problems. Our ST-MP7EH descriptor is based on the Edge Histogram image descriptor, part of the MPEG-7 standard [24], and is basically analyzing the temporal evolution of edges in video. Our descriptor works by computing an edge histogram in each frame, and then calculating means and standard deviations on the distribution in time of each bin in the histogram. By subsampling frames at a reasonably low rate we can decrease computation time, which is essential given our large video datasets.

ST-MP7EH has been tested on TRECVID data, with the goal of integrating it in a larger feature fusion scheme, and our experiments in section 4.3.4 are meant to highlight this idea. Results show that temporal statistics increase by almost 3 times the precision and that feature fusion with classic descriptors works well. Consequently ST-MP7EH was included in EURECOM's run in the 2011 edition of TRECVID SIN task. We have authored a publication on ST-MP7EH [59].

### 4.3.1 Previous Work

TRECVID systems seem to tend toward increasing the number of visual features and incorporating sophisticated fusion strategies while relying less on motion or edge information [3]. The most successful systems do incorporate spatio-temporal information by sampling multiple keyframes, however the number of frames extracted for one shot is generally extremely small (MediaMill uses up to 6 additional I-frames distributed around the middle key frame of each shot) [58].

However, several TRECVID participants, mostly in the new Multimedia Event Detection (MED) use edge features. [109] compute edge histograms on a local level and use the BoSW (Bag of Spatiotemporal Words) strategy to track features in the space-time volume. An interesting approach is the TGC (temporal gradient correlogram) by [54], which computes edge direction probabilities over 20 frames evenly distributed in the shot. Their work is similar in principle to ours, but the temporal aspect is represented by a mere concatenation of the 20 vectors resulting from the 20 sampled shots. In spite of some temporal information, this approach is highly dependent on the shot length and has a higher computational cost. The EOAC [55] (edge orientation autocorrelogram) is practically identical. Another example of edge-based descriptor is MEHI (motion edge history image) from [56], which evolves from previous MHI and MEI (heavily used in human action recognition), and is a suitable descriptor for human activity detection. However, its use in general concept-based retrieval is questionable because of camera motion, broader range of possible motions and inherent video quality problems that come with Internet archive videos. Moreover, the exhaustive manner of computation could prove impractical for the high-level feature task. In SIN (high level feature extraction), the MPEG-7 Edge Histogram has been used only in systems that work with the middle keyframe of the shot [110–112], thus without any spatio-temporal or motion information.

### 4.3.2 MPEG-7 Edge Histogram Descriptor

The MPEG-7 standard describes an Edge Histogram descriptor for images, which is meant to capture the spatial distribution of edges, as part of a general set of texture descriptors. As with all color and texture descriptors defined in the MPEG-7 standard [24], this descriptor is evaluated for its effectiveness in similarity retrieval [25], as well as extraction, storage and representation complexity. The distribution of edges is a good texture signature that is useful for image to image matching even when the underlying

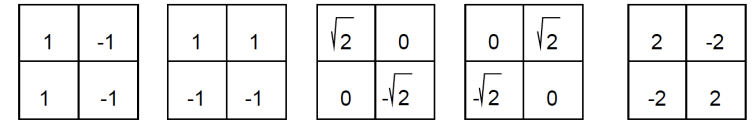


Figure 4.4: MPEG-7 directional filters [25]

texture is not homogeneous.

The exact method of computation for the MPEG-7 Edge Histogram descriptor can be found in [24] and [25]. The general idea is that the image is divided into $4 \times 4$ sub-images, and the local edge histograms are computed for each of the sub-images. There are 5 possible edge orientations that are considered: vertical, horizontal, 45° diagonal, 135° diagonal and isotropic (no orientation detected). For each sub-image and for each image type an edge intensity bin is computed, amounting to a total of $16\ \mathit{subimages} \times 5\ \mathit{edges} = 80$ bins.

Each sub-image is further divided into sub-blocks, which are down-sampled into a $2 \times 2$ pixel image by intensity averaging, and the edge-detector operators are applied using the 5 filters in the image below. The image blocks whose edge strengths exceed a threshold are marked as "edge blocks" and used in computing the histogram. These values are counted and normalized to $[0, 1]$ for each of the 80 bins. The value in each bin represents the "strength" of the corresponding edge type in that image block. According to its authors [24], this image descriptor is effective for representing natural images for image-to-image retrieval. It is not suited for object-based image retrieval. Moreover, the computation is efficient [25], and has low dimensionality and storage needs.

### 4.3.3 ST-MP7EH Spatio-Temporal Descriptor

In the context of video retrieval, visual descriptors that are traditionally used in CBIR are sometimes used to describe frame sequences instead of images, following a more or less elaborate extension process. A good example of a properly built 3D descriptor is the 3D extension [45] of SIFT or the extension [44] of HOG used initially in human action detection. However these cases are rare, as most systems tend to use keyframe-based approaches or compute 2D descriptors at salient points in the ST volume (detected by spatio-temporal interest points).

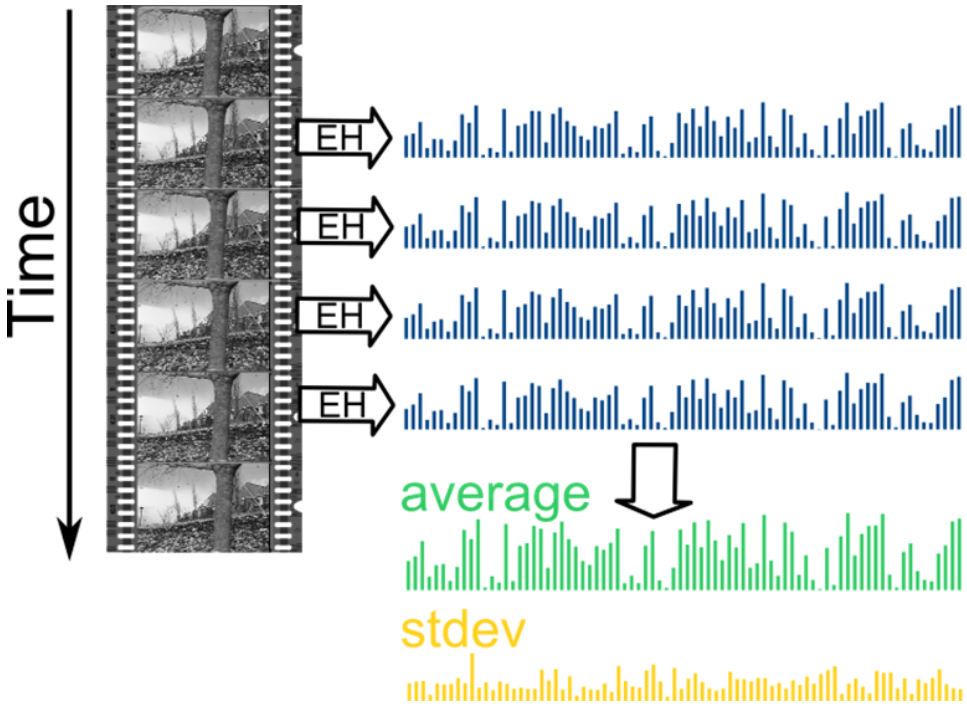


Figure 4.5: Overview of ST-MP7EH computation

Our opinion is that temporal and spatio-temporal features have a huge potential in content-based video retrieval, and at the same time keyframe approaches miss a great deal of information on two aspects: the feature we are looking for may not be present on the selected keyframe (but present on other frames in the shot), and the feature is easier to recognize by means of its dynamics throughout the shot rather than its instantaneous visual characteristics [36]. Naturally ST methods require far more processing power than keyframe approaches, so a compromise between performance and computational cost must always be made. For that reason, we decided on MPEG-7 Edge Histogram thanks to its low computational cost [25].

In the same idea as the seminal work of Nelson and Polana [113] we tried to create a global and general descriptor that can detect the evolution in time of visual texture. In our case, the texture is characterized by the predominance of an oriented edge on a region of the image. The ST descriptor is computed in a simple manner: for each frame t of the analyzed video, we compute the (2D) MPEG-7 edge histogram descriptor, which gives an

80 value feature vector $x_1^t$ to $x_{80}^t$. We compute this on every frame and put the data in an $N \times 80$ matrix, where $N$ is the number of analyzed frames. We consider each column $j$ of this matrix as a time series $x_j^1, x_j^2, ..., x_j^N$. This series represents the evolution in time of a certain feature of the image (e.g. the 19th element represents the strength of horizontal edges on the 3rd sub-image). The feature vector is made from the average $a_j$ and standard deviation $\sigma_j$ of each of these 80 series, namely $[a_1, a_2, ..., a_N, \sigma_1, \sigma_2, ..., \sigma_N]$ which gives it a fixed dimension of 160. The order of magnitude is conserved between the two descriptors by means of average and variance. The spatial information given by the edge distribution and the division into grids is inherited from the underlying edge histogram descriptor. Temporal information is present in the form of a shot-wide average edge histogram plus the temporal variance in each histogram bin. Formally, the mean represents the first moment of the discrete distribution, and the standard deviation is the square root of the second central moment (the variance). We use the standard deviation and not the variance in order to conserve the metric (to have the same measuring unit), which is essential to classification.

Similar temporal strategies have been used in space-time analysis before, most notably in human gesture recognition. Darrell and Pentland propose the use of Dynamic Time Warping [114] for gesture recognition, and other authors have proposed frequency domain [113] (Fourier analysis) and wavelets [115] to detect repetitive patterns in walking motion. However, in the context of general motion of an object in a video sequence, possibly combined with noise and camera motion, periodicity would obviously not prove as robust. The computational overhead would also be significant by comparison to state-of-the-art video retrieval systems. One thing our approach shares in common with frequency domain representations is that both methods store the mean of the signal: ST-MP7EH computes it explicitly and Fourier analysis computes the mean as the first Fourier coefficient.

In order to minimize computation time we temporally sub-sampled the frames of the shot by a 1/5 ratio. We found the subsampling appropriate for two reasons: firstly, because the functions we use should be unaffected by the subsampling of the dataset, and secondly because we assume the continuity of the MPEG-7 edge strengths in time (as they are calculated from a continuous shot). Intuitively this property should hold true for a sequence of frames forming a continuous shot: since the difference between any 2 consecutive shots is small, so should be the difference between 2 elements of the edge histogram for the corresponding edges.

### 4.3.4 Experiments

We have tested the proposed feature on our TRECVID 2010 platform. Eurecom's system uses a fusion of several visual classifiers (SIFT, GIST, color moments, wavelet features), of which SIFT is the earliest and still the most powerful, just like in most TRECVID systems.

#### ST-MP7EH Evaluation

We test the performance of our ST-MP7EH descriptor on TRECVID data. The chosen training and test sets are two subsets of the annotated TRECVID 2010 data. Our experiments were conducted using the 10 concepts from the TRECVID 2010 "light run" of the SIN task. The training set contains 59,800 shots and the test set 59,885 shots, which are designated '2010a' and '2010b' in TRECVID.

Given the fact that this is an "embarrassingly parallel" problem, splitting the workload into a manageable number of jobs is trivial. Computation time for a single shot depends on shot length and frame size, as well as hardware-dependent considerations. On average for 10% of the tv10.shorts.B corpus (137,327 shots) it takes approx. 28.23 hours, which makes for an average 7.4011 seconds per shot. This can also be approximated as 1.23× playback time. We estimate average memory usage at 5.747 kB per shot. The implementation is based on an unoptimized MATLAB edge extractor. Optimizing the approach could greatly improve speed.

Classification is performed with SVMs. We label our data using the available annotation. At this point the number of training examples becomes concept-dependent as the annotation is not complete over the entire dataset. We use a modified version of the SVM software available from LibSVM [63] that uses the exponential $\chi 2$ kernel. As with all other SVM training experiments, we use one binary SVM per concept. Given the disproportionate nature of the positive and negative examples (positive/negative ratio is $< 1\%$ for every concept), the label obtained in testing will always be negative. We use the probability estimates derived from Platt's formula implemented in LibSVM as concept scores. We sort by this probability in order to obtain our ranked list on which we compute the AP for the 10 concepts. The average of APs over all concepts is the MAP.

#### Comparison with Spatio-Temporal Baseline Descriptors

We compared ST-MP7EH with the only implemented spatio-temporal feature we had available at the time: the 3 variants of our SIFT multi-keyframe

Table 4.2: Comparison between ST-MP7EH and mkfSIFT on TRECVID 2010 test

| Descriptor | ST-MP7EH | mkfSIFT1 | mkfSIFT2 | mkfSIFT3 |
|---|---|---|---|---|
| Airplane_Flying | 0.00174 | 0.00589 | **0.02514** | 0.01793 |
| Boat_Ship | **0.02417** | 0.01880 | 0.01797 | 0.02367 |
| Bus | 0.00273 | 0.00348 | **0.00629** | 0.00514 |
| Cityscape | 0.17529 | **0.17755** | 0.15426 | 0.13681 |
| Classroom | **0.01240** | 0.00785 | 0.00465 | 0.00900 |
| Demonstration_Or_Protest | 0.02246 | 0.03806 | 0.03266 | **0.06042** |
| Hand | **0.01947** | 0.00335 | 0.00759 | 0.00572 |
| Nighttime | 0.02681 | 0.02773 | **0.06348** | 0.04030 |
| Singing | **0.15701** | 0.06039 | 0.07564 | 0.07210 |
| Telephones | 0.00132 | **0.00969** | 0.00346 | 0.00302 |
| MAP | **0.04434** | 0.03528 | 0.03911 | 0.03741 |

baseline described in 4.2. Table 4.2 shows the results of the experiment. Improvement is noticeable in the MAP and in the concepts Boat_Ship, Classroom, Hand and Telephones.

### Gain from spatial to spatiotemporal

We remind that the proposed descriptor is a concatenation of average and deviations of the evolution in time of Edge Histograms. In this experiment we compare the proposed approach with simply using one Edge Histogram. This histogram is extracted from the keyframe defined in every shot. The 80-value vector obtained from the MPEG-7 Edge Histogram is used in SVM classification in a similar way to ST-MP7EH. Results in table 4.3 clearly indicate the gain of using multiple keyframes per shot.

### ST-MP7EH - SIFT Late Fusion

Following the multiple descriptor fusion paradigm that seems to dominate current state-of-the-art systems, especially in TRECVID, we perform weighted linear fusion between the SIFT descriptor and ST-MP7EH. The idea was to show that whilst both descriptors provide good concept recognition separately, each one represents a different type of visual information: SIFT is a very accurate texture descriptor, while ST-MP7EH detects shapes through edge orientations. In our experiment we follow the technique described in

Table 4.3: Comparison Between ST-MP7EH and MPEG-7 Edge Histogram on TRECVID 2010b dataset

| Descriptor | ST-MP7EH | MPEG-7 edge |
|---|---|---|
| Airplane_Flying | 0.00174 | 0.00210 |
| Boat_Ship | 0.02417 | 0.00527 |
| Bus | 0.00273 | 0.00006 |
| Cityscape | 0.17529 | 0.02011 |
| Classroom | 0.01240 | 0.00374 |
| Demonstration_Or_Protest | 0.02246 | 0.00629 |
| Hand | 0.01947 | 0.01700 |
| Nighttime | 0.02681 | 0.05869 |
| Singing | 0.15701 | 0.05053 |
| Telephones | 0.00132 | 0.00155 |
| MAP | 0.04434 | 0.01653 |

annex B to prove that fusing two different descriptors in such a manner could significantly improve the MAP. Results are shown in table 4.4.

Figure 4.6 shows how different mixes between ST-MP7EH and SIFT perform. Each group of 10 columns corresponding to one concept represent the 10 values for the $\alpha$ parameter, from 0 to 1. Zero means the scores have zero weight on SIFT and full weight on ST-MP7EH, one meaning the opposite. The average gain in precision caused by late fusion is 22%, corresponding to a MAP of 0.0587.

### 4.3.5 Conclusions

Current large scale concept video retrieval systems show a slow adoption of dynamic features. In the generalistic concept classifiers seen in the Semantic Indexing Task, the vast majority of systems still use only one keyframe to describe the entire shot. Only two different approaches have proven successful: local features (either in space or space-time) computed around STIP [42] and the use of more than one keyframe in shot description. However these descriptors are part of complex systems where they participate in feature fusion. On all accounts, any method that analyzes the XYT volume is subject to a high processing and memory penalty. Our descriptor is invariant to changes in temporal scale, so that the frames of the shot can be subsampled up to a minimal rate. We can improve computation time by lowering the sampling rate with very little change in the feature vector.

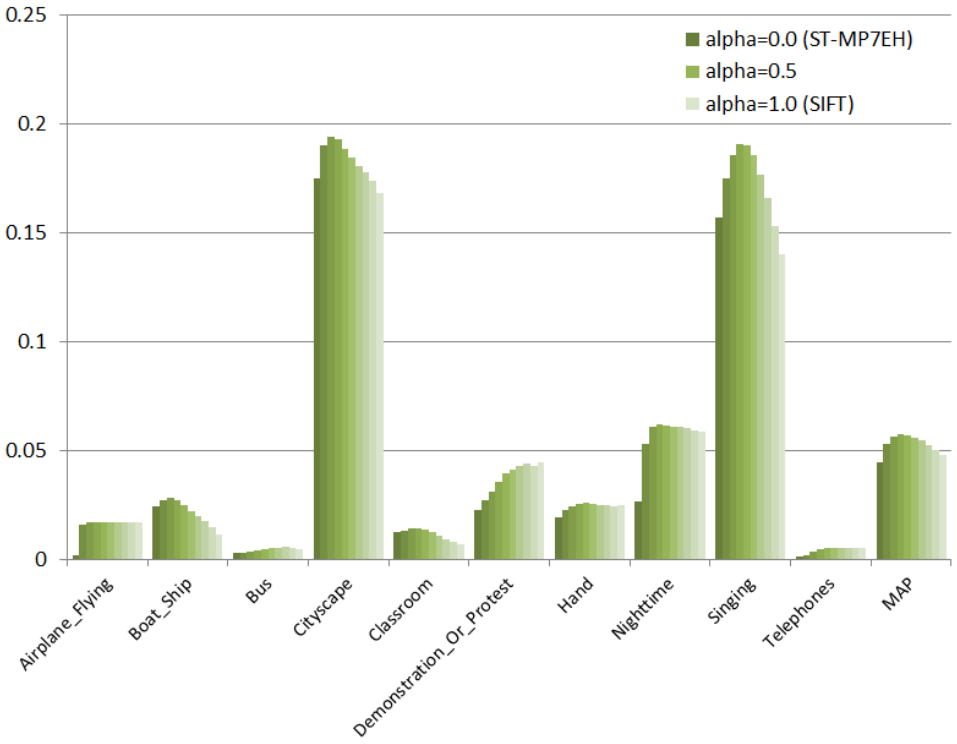


Figure 4.6: Average Precision for late fusion between ST-MP7EH and SIFT

Table 4.4: Late fusion between ST-MP7EH and SIFT

| | ST-MP7EH | SIFT | fusion | alpha |
|---|---|---|---|---|
| Airplane_Flying | 0.00174 | 0.0170 | 0.0172 | 0.3 |
| Boat_Ship | 0.02417 | 0.0116 | 0.0280 | 0.2 |
| Bus | 0.00273 | 0.0048 | 0.0056 | 0.7 |
| Cityscape | 0.17529 | 0.1681 | 0.1945 | 0.2 |
| Classroom | 0.01240 | 0.0067 | 0.0142 | 0.3 |
| Demonstration_Or_Protest | 0.02246 | 0.0444 | 0.0444 | 1.0 |
| Hand | 0.01947 | 0.0249 | 0.0259 | 0.4 |
| Nighttime | 0.02681 | 0.0586 | 0.0619 | 0.3 |
| Singing | 0.15701 | 0.1404 | 0.1906 | 0.3 |
| Telephones | 0.00132 | 0.0049 | 0.0051 | 0.4 |
| MAP | 0.04434 | 0.0481 | 0.0587 | - |

$$div = \frac{1}{2}(a_2 + a_6) \qquad rot = \frac{1}{2}(a_5 - a_3)$$
$$hyp_1 = \frac{1}{2}(a_2 - a_6) \qquad hyp_2 = \frac{1}{2}(a_5 + a_3)$$

Our experiment shows that MAP increases by almost 3 times when passing from the keyframe-based MPEG-7 Edge Histogram Descriptor to ST-MP7EH. This is remarkable since the two descriptors carry the same spatial information. The improvement is simply due to the extension in time.

## 4.4 Study on camera motion parameters

It is a common (mis)conception that the presence of movement or type of camera motion in a video may be a strong hint on the presence of certain concepts. For instance, many researchers have suggested that concept *Mountain* should contain little camera motion (since the scene is usually static) and the concept *Running* should display horizontal camera rotation. In this section we challenge this hypothesis by recognizing the amount of frame-level camera motion on all TRECVID 2010 videos and correlating them with the learning annotations for each concept. The goal of the study is to evaluate correlations between types of camera motion (e.g. panning, zooming, translation, etc.) and concepts.

Using the camera motion compensation framework used in later experiments, an affine camera transformation $A = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix}$ is computed using RANSAC from correspondence pairs between each frame transition. The displacement vector $\vec{d(p)}$ for a point $p = (x, y)$ is:

$$\vec{d(p)} = \begin{pmatrix} a_1 + a_2(x - x_c) + a_3(y - y_c) \\ a_4 + a_5(x - x_c) + a_6(y - y_c) \end{pmatrix} \tag{4.4}$$

where $(x_c, y_c)$ are the coordinates of the screen center. The 6 parameters can be expressed in a new basis in terms of divergence, curl and two hyperbolic terms:

These parameters, along with translational coordinates $L_{tr}$ can further be physically interpreted according to the following table:

According to the motion classes defined above, we categorize the types of motion that we can detect into 4 *motion labels*: $P$ – panning/tilt/traveling, $R$ – rotation of the camera around the optic axis, $Z$ – can mean zooming in/out or traveling forward/background and $C$ which illustrates a motion that cannot be modeled by homography (for instance, distortion caused by

Table 4.5: Camera motion classification based on hyper-parameters. Source: [60]

| $L_{lin}$ | $L_{tr}$ | Camera motion |
|---|---|---|
| $(0,0,0,0)$ | $(0,0)$ | Static camera |
| $(0,0,0,0)$ | $\neq (0,0)$ | Pan, tilt,<br>or sideways camera traveling<br>if the scene background is parallel to the image plane |
| $(div,0,0,0)$ | | Zoom (in/out) or forward/backward traveling |
| $(0,rot,0,0)$ | | Rotation around the optical axis |
| $(div,rot,0,0)$ | | Zoom (in/out) or forward/backward traveling<br>and rotation around the optical axis |
| $(div,rot,hyp_1,hyp_2)$ | | Sideways camera traveling,<br>(if the scene background is not approximately a plane parallel to the image)<br>or complex camera motion. |

parallax, which cannot by modeled by affine transformation). At shot level, each of these labels can arbitrarily appear at any frame transition. We compute a shot-level percentage of each motion label by normalizing the number of transitions containing the corresponding label in the shot to the total number of transitions in the shot (i.e. 11 panning transitions in a shot containing 23 frames is 50%, a shot containing 100% PAN movement consists of transitions that all contain panning). Transition labels are not exclusive. For every annotated shot for each of 50 concepts in TRECVID 2010 dev set we calculate the following measures: the correlation between the concept label (-1, +1) and the percentage of motion of type P, R, Z and C; the correlation between the concept label and the binarized motion of type P, R, Z and C, obtained by thresholding the percentage at zero; mutual information between concept label and the binarized motion value. Results are displayed in table 4.6, where the notations are: "corr/" = correlation with motion, "corr/L" = correlation with binarized motion label and "mi/" = mutual information. The first column "corr/F" described the correlation between the concept label and the number of frames.

The results clearly show that not a single concept is even weakly correlated to a type of motion (largest absolute value in the graph does not exceed 0.2), aside from some negative correlations of camera motion with the concept 'Indoor' and positive with 'Road' which make sense.

## 4.5 Conclusions

The fact that neither the established ST feature HOG3D, nor the multi keyframe modification of SIFT-BOW do not contribute to keyframe classifiers on the TRECVID dataset raises some questions over the validity of

| | corr/F | corr/P | corr/R | corr/Z | corr/C | corr/LP | corr/LR | corr/LZ | corr/LC | mi/P | mi/R | mi/Z | mi/C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 'Adult' | 0.032 | -0.071 | -0.038 | -0.041 | -0.047 | 0.004 | 0.000 | 0.018 | -0.014 | 1.00E-05 | 1.19E-07 | 2.30E-04 | 1.36E-04 |
| 'Airplane_Flying' | 0.001 | 0.004 | 0.007 | 0.007 | 0.018 | 0.008 | 0.001 | 0.011 | 0.013 | 5.07E-05 | 3.30E-07 | 8.76E-05 | 1.13E-04 |
| 'Animal' | 0.009 | -0.021 | -0.014 | -0.014 | -0.015 | 0.005 | 0.008 | 0.011 | 0.006 | 1.91E-05 | 5.20E-05 | 9.66E-05 | 2.61E-05 |
| 'Asian_People' | -0.006 | -0.004 | -0.005 | -0.008 | -0.007 | 0.004 | 0.014 | 0.007 | 0.014 | 9.57E-06 | 1.39E-04 | 3.25E-05 | 1.32E-04 |
| 'Bicycling' | -0.008 | 0.053 | 0.056 | 0.042 | 0.024 | 0.019 | 0.031 | 0.016 | 0.013 | 3.01E-04 | 7.12E-04 | 2.02E-04 | 1.23E-04 |
| 'Boat_Ship' | -0.005 | -0.013 | 0.035 | -0.001 | -0.007 | 0.006 | 0.031 | 0.021 | 0.038 | 2.88E-05 | 6.99E-04 | 3.36E-04 | 9.30E-04 |
| 'Building' | -0.060 | 0.056 | -0.001 | 0.021 | -0.008 | 0.000 | -0.038 | -0.035 | -0.058 | 6.17E-11 | 1.04E-03 | 8.66E-04 | 2.51E-03 |
| 'Bus' | 0.014 | -0.006 | 0.001 | 0.006 | -0.007 | -0.011 | -0.003 | -0.005 | -0.009 | 8.02E-05 | 5.30E-06 | 1.73E-05 | 6.45E-05 |
| 'Car' | -0.020 | 0.029 | 0.016 | 0.014 | 0.011 | 0.009 | 0.006 | -0.006 | 0.010 | 6.24E-05 | 2.66E-05 | 2.39E-05 | 6.54E-05 |
| 'Cheering' | -0.010 | -0.017 | -0.011 | 0.023 | -0.012 | -0.025 | -0.009 | -0.002 | -0.021 | 4.37E-04 | 5.71E-05 | 2.93E-06 | 3.64E-04 |
| 'Cityscape' | -0.028 | 0.036 | -0.006 | 0.065 | -0.011 | 0.028 | 0.004 | 0.027 | 0.006 | 6.02E-04 | 1.46E-05 | 5.44E-04 | 2.84E-05 |
| 'Classroom' | 0.023 | -0.011 | -0.015 | -0.013 | -0.010 | 0.006 | -0.018 | -0.010 | -0.019 | 3.08E-05 | 2.63E-04 | 6.46E-05 | 3.61E-04 |
| 'Computer_Or_Television_Screens' | 0.033 | -0.096 | -0.036 | -0.087 | -0.030 | 0.014 | 0.024 | 0.016 | 0.052 | 1.33E-04 | 4.13E-04 | 1.93E-04 | 1.93E-03 |
| 'Computers' | 0.032 | -0.085 | -0.065 | -0.069 | -0.046 | -0.085 | -0.073 | -0.090 | -0.054 | 5.03E-03 | 4.16E-03 | 5.81E-03 | 2.35E-03 |
| 'Dancing' | -0.016 | 0.000 | -0.004 | 0.000 | 0.001 | 0.002 | 0.006 | 0.016 | 0.004 | 1.81E-06 | 3.03E-05 | 1.97E-04 | 1.25E-05 |
| 'Dark-skinned_People' | -0.018 | -0.034 | -0.024 | -0.006 | -0.017 | -0.009 | -0.005 | 0.024 | -0.011 | 5.26E-05 | 1.95E-05 | 4.15E-04 | 9.30E-05 |
| 'Demonstration_Or_Protest' | -0.010 | 0.022 | -0.004 | 0.003 | -0.017 | 0.018 | 0.002 | -0.002 | -0.022 | 2.46E-04 | 3.36E-06 | 2.20E-06 | 3.88E-04 |
| 'Doorway' | -0.015 | -0.045 | -0.041 | -0.035 | -0.029 | -0.030 | -0.046 | -0.045 | -0.042 | 6.19E-04 | 1.55E-03 | 1.46E-03 | 1.34E-03 |
| 'Explosion_Fire' | -0.039 | 0.067 | 0.052 | 0.037 | 0.089 | -0.025 | -0.013 | -0.020 | 0.018 | 4.33E-04 | 1.14E-04 | 2.88E-04 | 2.26E-04 |
| 'Female-Human-Face-Closeup' | -0.001 | -0.014 | -0.001 | 0.001 | -0.005 | 0.012 | 0.037 | 0.033 | 0.016 | 9.89E-05 | 9.93E-04 | 8.38E-04 | 1.83E-04 |
| 'Female_Person' | 0.000 | -0.029 | -0.049 | -0.040 | -0.032 | -0.015 | -0.036 | -0.018 | -0.037 | 1.72E-04 | 9.19E-04 | 2.36E-04 | 1.04E-03 |
| 'Flowers' | -0.013 | -0.015 | 0.001 | -0.010 | -0.012 | -0.056 | -0.033 | -0.035 | -0.025 | 2.11E-03 | 8.49E-04 | 8.42E-04 | 5.16E-04 |
| 'Ground_Vehicles' | -0.025 | 0.042 | 0.052 | 0.051 | 0.036 | 0.006 | 0.026 | 0.010 | 0.022 | 2.59E-05 | 4.82E-04 | 7.10E-05 | 3.25E-04 |
| 'Hand' | -0.015 | 0.002 | -0.019 | 0.001 | -0.003 | 0.036 | 0.004 | 0.030 | 0.016 | 1.00E-03 | 1.12E-05 | 6.60E-04 | 1.74E-04 |
| 'Helicopter_Hovering' | 0.003 | 0.000 | 0.006 | 0.006 | 0.007 | 0.005 | 0.007 | 0.005 | 0.010 | 2.75E-05 | 5.21E-05 | 3.22E-05 | 7.62E-05 |
| 'Indoor' | 0.076 | -0.149 | -0.119 | -0.129 | -0.094 | -0.041 | -0.048 | -0.035 | -0.052 | 1.22E-03 | 1.64E-03 | 8.62E-04 | 2.00E-03 |
| 'Indoor_Sports_Venue' | 0.019 | -0.009 | -0.024 | -0.022 | -0.017 | 0.018 | 0.011 | 0.017 | 0.010 | 2.38E-04 | 8.74E-05 | 2.11E-04 | 6.40E-05 |
| 'Infants' | -0.001 | -0.015 | -0.011 | 0.000 | -0.008 | -0.010 | -0.005 | -0.001 | -0.010 | 7.36E-05 | 1.94E-05 | 7.52E-07 | 8.83E-05 |
| 'Instrumental_Musician' | -0.012 | -0.032 | -0.007 | 0.010 | -0.011 | 0.006 | -0.001 | 0.010 | 0.003 | 2.53E-05 | 8.04E-07 | 6.56E-05 | 7.23E-06 |
| 'Landscape' | 0.019 | -0.033 | 0.009 | -0.014 | -0.013 | 0.007 | 0.033 | 0.034 | 0.057 | 4.08E-05 | 7.82E-04 | 8.77E-04 | 2.28E-03 |
| 'Male_Person' | 0.062 | -0.070 | -0.035 | -0.059 | -0.053 | 0.004 | -0.013 | -0.012 | -0.018 | 1.35E-05 | 1.14E-04 | 9.82E-05 | 2.50E-04 |
| 'Military_Base' | -0.006 | -0.005 | -0.007 | -0.011 | -0.005 | 0.015 | 0.018 | 0.009 | 0.008 | 1.85E-04 | 2.40E-04 | 5.93E-05 | 4.12E-05 |
| 'Mountain' | -0.007 | 0.009 | 0.015 | 0.011 | 0.017 | 0.029 | 0.039 | 0.042 | 0.058 | 6.63E-04 | 1.12E-03 | 1.35E-03 | 2.30E-03 |
| 'News_Studio' | 0.063 | -0.088 | -0.065 | -0.087 | -0.044 | -0.055 | -0.035 | -0.041 | -0.037 | 2.05E-03 | 9.18E-04 | 1.18E-03 | 1.07E-03 |
| 'Nighttime' | 0.001 | 0.065 | 0.076 | 0.079 | 0.105 | 0.017 | 0.040 | 0.032 | 0.052 | 2.32E-04 | 1.17E-03 | 8.26E-04 | 1.82E-03 |
| 'Old_People' | 0.026 | -0.024 | -0.020 | -0.018 | -0.013 | 0.002 | 0.008 | 0.001 | 0.014 | 4.10E-06 | 4.37E-05 | 3.84E-07 | 1.42E-04 |
| 'Plant' | -0.002 | 0.003 | -0.005 | -0.010 | -0.014 | -0.012 | -0.016 | -0.018 | -0.019 | 1.02E-04 | 1.77E-04 | 2.42E-04 | 2.51E-04 |
| 'Road' | -0.060 | 0.165 | 0.118 | 0.120 | 0.116 | 0.016 | 0.031 | 0.010 | 0.031 | 1.83E-04 | 7.09E-04 | 7.44E-05 | 6.82E-04 |
| 'Running' | -0.003 | 0.038 | 0.021 | 0.020 | 0.017 | 0.018 | 0.017 | 0.023 | 0.015 | 2.75E-04 | 2.07E-04 | 4.38E-04 | 1.42E-04 |
| 'Scene_Text' | -0.035 | -0.003 | -0.037 | -0.001 | -0.021 | 0.000 | -0.054 | -0.030 | -0.056 | 1.19E-08 | 2.21E-03 | 6.42E-04 | 2.40E-03 |
| 'Singing' | -0.019 | -0.028 | -0.013 | 0.019 | -0.018 | -0.002 | 0.012 | 0.035 | 0.006 | 4.24E-06 | 1.03E-04 | 9.12E-04 | 2.69E-05 |
| 'Sitting_Down' | 0.056 | -0.062 | -0.047 | -0.060 | -0.031 | -0.031 | -0.045 | -0.041 | -0.029 | 6.63E-04 | 1.52E-03 | 1.19E-03 | 6.50E-04 |
| 'Stadium' | 0.002 | 0.008 | -0.010 | -0.002 | -0.005 | 0.034 | 0.026 | 0.035 | 0.021 | 9.23E-04 | 4.70E-04 | 9.10E-04 | 3.05E-04 |
| 'Swimming' | -0.003 | -0.011 | -0.003 | 0.000 | -0.007 | -0.003 | 0.003 | 0.003 | -0.011 | 5.46E-06 | 7.28E-06 | 6.95E-06 | 1.04E-04 |
| 'Telephones' | -0.013 | -0.023 | -0.024 | -0.029 | -0.023 | -0.021 | -0.036 | -0.030 | -0.038 | 3.02E-04 | 9.96E-04 | 6.39E-04 | 1.24E-03 |
| 'Throwing' | 0.006 | 0.010 | -0.003 | -0.002 | 0.001 | 0.020 | 0.015 | 0.016 | 0.013 | 3.73E-04 | 1.61E-04 | 2.08E-04 | 1.09E-04 |
| 'Vehicle' | -0.029 | 0.040 | 0.067 | 0.060 | 0.024 | 0.021 | 0.045 | 0.022 | 0.035 | 3.09E-04 | 1.47E-03 | 3.43E-04 | 8.37E-04 |
| 'Walking' | -0.004 | 0.022 | 0.012 | 0.003 | 0.009 | 0.014 | 0.004 | 0.001 | -0.001 | 1.41E-04 | 9.52E-06 | 1.52E-06 | 9.57E-07 |
| 'Walking_Running' | -0.015 | 0.072 | 0.051 | 0.038 | 0.042 | 0.028 | 0.026 | 0.018 | 0.017 | 5.87E-04 | 4.89E-04 | 2.36E-04 | 2.06E-04 |
| 'Waterscape_Waterfront' | 0.006 | -0.047 | 0.028 | -0.014 | -0.017 | 0.007 | 0.036 | 0.042 | 0.044 | 3.25E-05 | 9.10E-04 | 1.33E-03 | 1.36E-03 |

Table 4.6: Label-concept correlation, Binarized label-concept correlation and mutual information between label and concept for 4 camera motion types

ST features in concept detection. While motion itself obviously contributes to the understanding of a scene for humans, perhaps the current quantity and quality of TRECVID videos, or the way classifier performance is evaluated are not suitable enough for ST descriptors. In our experiments no true spatio-temporal baseline has been found, and subsequent experiments have used the SIFT-BOW as baseline.

For all its simplicity, ST-MP7EH has surprisingly proved to fulfill all the requirements of a TRECVID feature: fast, lightweight, easy to classify and with good fusion improvement.

The camera motion experiment shows that at very large scale data becomes so diverse that simple discriminative approaches such as the correlation between concept and camera motion become unusable.

# Chapter 5

# BOW Methods inspired by action recognition

Upon studying the literature it becomes evident that our research goal of detecting concepts in video using dynamic features has many common points with the action recognition field. In both areas one extracts motion information and identifies highly-variable movement patterns in un- or weakly-constrained video. Particularly [98] seems to report very similar challenges: "individual variations of people in expression, posture, motion and clothing; perspective effects and camera motions; illumination variations; occlusions and disocclusions; the distracting effect of the surroundings". Also [104] defines a dataset of videos "in the wild" obtained from Youtube, hence exhibiting the same characteristics as TRECVID videos: poor quality/resolution, camera motion, overlaid text, etc.

Extensive review papers such as [2] show that comparatively much more research is carried out on action recognition than on general concept retrieval. As a consequence a more varied selection of video datasets are available for action recognition: KTH, Weizmann, Youtube, Hollywood2. These datasets vary in size, quality and complexity but they are all much smaller than TRECVID. Unfortunately many of the state-of-the-art action recognition methods are simply too computationally heavy to apply on TRECVID, for reasons of high memory and CPU consumption, higher needed storage and lower degree of generality of action detection compared to concept detec-

tion. In this chapter, we develop a content description method applicable on action recognition datasets *and* on the TRECVID collection at the same time. Additionally, we compare this multi-dataset performance with other descriptors found in literature, wherever available.

The motivation for the research in this chapter comes from the fact that recent results show improvement in categorizing human actions both on specialized action datasets (KTH, Weizmann) and on more realistic web videos (Youtube, Hollywood2) [20, 46, 48, 104, 116, 117] yet such methods are rarely used when detecting similar concepts on larger sets of unconstrained videos such as TRECVID [3, 4].

The contributions presented in this part can be summarized as following: in this chapter we provide a method for high level motion characterization in unconstrained video tailored for use as feature in TRECVID Semantic Indexing classifiers, named BOMT (Bags of Motion Templates). On chapter 5.3 we experiment on 5 video datasets that range from very small and constrained to real life Internet video "in the wild", as well as varying heavily in size and quality, and show how recognition performance degrades with respect to the dataset chosen. In chapter 5.5 we apply BOMT on the TRECVID dataset, calculate the mean average precision gained by the inclusion of BOMT in a linear fusion system and we illustrate the performance vs. speed balance of BOMT by comparing it in terms of accuracy and speed with several other state of the art action descriptors over the KTH dataset.

## 5.1 Related Work

Dynamic features have been studied mainly for recognizing human activity. [35, 36, 56, 64, 68, 69, 105, 106, 114, 115]. For a complete review of these methods, see [2]. A large subset of these methods deal with sparse local spatio-temporal features: Laptev's Spatio-Temporal Interest Points [42] detect corners in video volume and describe the local cube with histograms of gradient and optic flow, Dollar's cuboids [43] detect maximal response zones of temporal Gabor filters, Scovanner [45] and Kläser [44] extend to spatio-temporal level existing image features SIFT [17] and HOG [23]. More recently, Chen [46] enhances the SIFT descriptor by adding a matching histogram representation of the optic flow in the patch, and Wang [48] updates the HOGHOF feature by adding motion boundary features computed along densely sampled trajectories.

The focus of our research is however on video classification in large video collections. To this end, the TRECVID [4] evaluation campaign offers a very

good insight to concept-based video retrieval. In recent years a dominating strategy of TRECVID systems has been observed: local visual features, of which SIFT [17] is the most used, are extracted sparsely or densely from a video keyframe; the keyframe is then represented as a sparse vector of occurrence counts of local image features. This is known as the Bag of Words model. The resulting histograms are then used as inputs to a Support Vector Machine, that will learn to separate between positive and negative examples. Since such video classifiers represent a video shot by only a keyframe, they are effectively performing CBIR. [47]

The analogy between the CBIR features and their 3D counterparts mentioned earlier makes logical the adoption in video content analysis of the BOW + SVM model. Indeed, heavily cited work [20,48,68,69] shows promising results by using bags of visual words from local spatio-temporal features. However, they are applied experimentally on human actions, which can be intuitively considered as a specific set of semantic concepts. The scalability of the approaches is still in question, since the datasets are considerably smaller both in number of classes and number of sequences, while showing better quality than TRECVID videos (see section 3.2.3 for further details).

Some earlier research on action recognition [118] also based on the MHI and a two-view SVM_2K report average accuracy of 65% on KTH. Some more recent results on Youtube, KTH and UCF50 are presented in [119].

## 5.2 Bags of Motion Templates method

Our method consists of a feature extraction stage, an intermediary representation stage and a classification stage, as seen in figure 5.1.

During the first stage, Motion History Image is extracted (b) for all the frames of the sequence (a), and Motion Templates are segmented (c). These Motion Templates are then represented by a HOG-like feature. The resulting HOG vectors make the input of the Bag of Words model, which in turn provides a video-level representation of the motion (d). This Bag of Words histogram is used in training and testing one Support Vector Machine for every concept/action that we want to detect. The following chapter describes all these steps in detail.

### 5.2.1 Motion History Image

Each of the studied datasets contains videos with some degree of camera motion, therefore a stabilization method should be used. We use the camera stabilization function described in A. This method does dominant motion

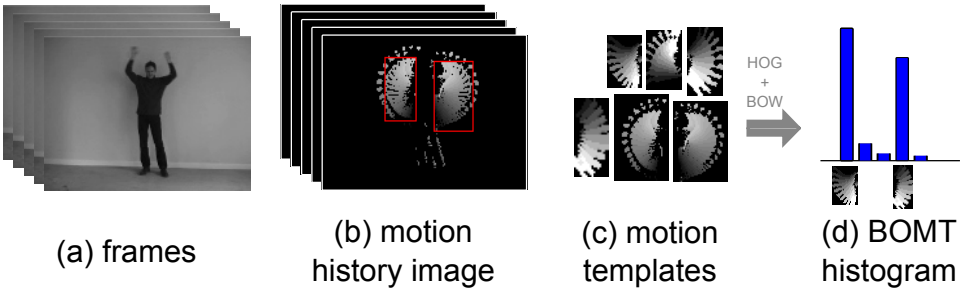


Figure 5.1: Overview of BOMT. From the frame sequence (a) Motion History Image (b) is extracted at every frame. The motion in each MHI is segmented (c), resulting in Motion Templates that represent parts of actions. A motion template is described by a HOG-like feature, which is then quantized into a Bag of Words histogram (d)

compensation by estimating a homography with RANSAC over detected feature correspondences. Using the resulting homography we make a back-projection (warping) of the previous frame and, while carefully keeping track of the pixels that fall outside the projection, we compute the difference between the warped previous frame and the current frame. Thresholding this frame differencing image gives the motion mask $\Psi(x, y, t)$ for the current frame $t$. Ideally the motion mask is a binary image of the moving parts of an object of interest.

Our motion feature is based on the Motion History Image [35], which is a representation of recent movement (see figure 5.1) that is a 2D representation of the spatio-temporal movement in the sequence. [61] The previously computed motion mask $\Psi(x, y, t)$ serves as an update function for the MHI $H(x, y, t)$. The MHI is defined [61] as:

$$H(x,y,t) = \begin{cases} 1 & \text{if } \Psi(x,y,t) = 1 \\ max(0, H(x,y,t-1) - \delta) & \text{otherwise} \end{cases} \tag{5.1}$$

In essence, the MHI is created by lowering the intensity of the previous MHI and updating it with the new motion mask. The $\delta$ parameter controls the decay of the MHI; the higher the value, the quicker old motion will fade away.

### 5.2.2 Motion Templates

We use the technique described in [6] and implemented in OpenCV [62] to segment regions of the MHI that contain parts of motion. This method computes the 2D gradient of the MHI and checks locally for the uniformity of this gradient map. This process removes most of the noise that comes from the motion mask. Remaining MHI zones have uniform gradient which can be further interpreted as an indication of motion direction and speed. The orientation of the 2D gradient vector represents local motion direction, and its magnitude is inversely proportional with motion velocity. The borders of these regions can be easily segmented by detecting connected components with a special connectivity condition: two adjacent MHI pixels belong to the same region if their MHI values differ by less than a priorly chosen threshold value $\delta_{max}$. Thus, by filtering out zones with noisy gradient and by segmenting parts of the MHI we can obtain a set of regions. The resulting regions are named motion templates (MT). There is no restriction on the number of MTs that can be found in one MHI.

A common conception in the action recognition community says that action classifiers should be invariant to the horizontal direction of the action (i.e. a classifier should detect equally *running left* and *running right*). A simple way of enforcing this constraint and to double the number of training samples is to train with both the original videos and with vertically flipped videos. We achieve the same effect by duplicating and flipping the motion template after extraction.

Figure 5.2 provides an example of motion template extraction: the person on the left is walking towards the right, a fact shown by the MHI intensity increasing from left to right. The basketball displays a trace characteristic of a rebound movement. A car is moving in the background and is extracted too. Because there is little movement in the shoulder and neck area, the MHI for person to the right is oversegmented into 2 motion templates. Thus there is a total of 5 motion templates extracted from this frame.

MTs have the property of being characteristic to the action performed [6]. Since they do not incorporate any information on the shape, color, or texture of the subject performing the action, they are invariant to these factors. Instead, they can be seen as *atoms of actions*, generally each following the movements of a single body part. The classic example from figure 5.1 in the KTH dataset shows an MHI with two MTs highlighted, each describing the movement of one arm. It is easy to see that the arms have just been raised, since the more recent motion (higher intensity) is located above the head while the older is below (lower intensity).

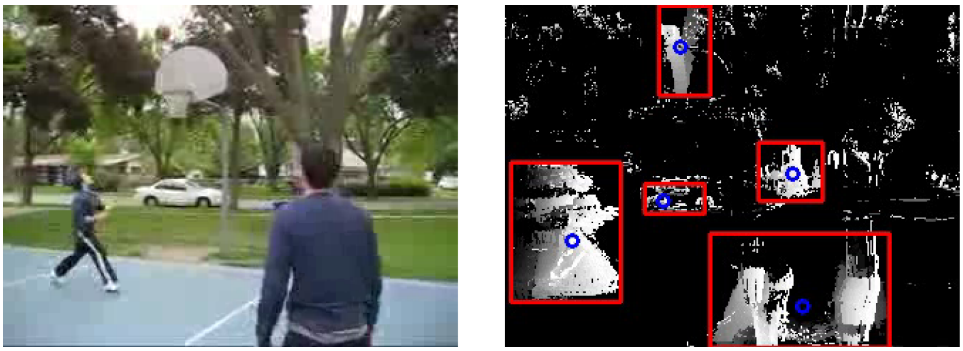

Figure 5.2: Motion Templates segmented in a sequence from the Youtube action dataset.

When studying the extracted MTs for each frame in the sequence, one can make two important observations: the first is that MTs relevant to the action tend to repeat themselves in time. That is to say, very similar motion templates are likely to be found at different time instances for the same action. This is intuitively true for repetitive actions such as walking, but in practice it holds true for basically any action in short time spans. The second observation is that many "parasite" MTs may appear in a sequence, because of either dynamic backgrounds (foliage, clouds, water), artifacts caused by imprecise camera motion compensation, movement of other objects that are irrelevant to the action, superimposed graphics such as logos etc. The quantity of these parasite MTs is highly dependent on the previous steps, and camera stabilization or motion mask extraction methods are not the purpose of this work. A simple way to reduce noisy MTs is to impose a lower size threshold, thus supressing very small MTs.

### 5.2.3 HOG feature

In order to have a vectorial representation of the MTs, we use a descriptor similar to HOG. [23] We first rescale the MT to a constant square size of $256 \times 256$ pixels in order to obtain scale and aspect invariance of the MT. We divide the MT in 16 regions of a $4 \times 4$ grid (see figure 5.3), and in each grid we quantize orientations of the image gradient in 8 bins plus one extra bin for zero gradient. Our feature has dimensionality 144.

There is considerable variability in the way people execute actions naturally. Walking pace is known to vary greatly across individuals. Invariability

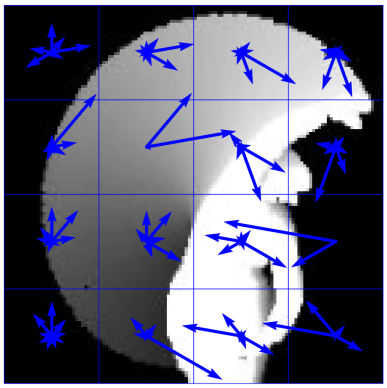

Figure 5.3: Example of a Motion Template with the corresponding HOG bins

to the speed of the action is desired. We accomplish it through two mechanisms:

- *intra-motion template velocity invariance* is achieved by the HOG representation: when speeding up an action the MHI gradient does not change orientation but merely increases in magnitude. Since the HOG-like representation uses only gradient orientations and not magnitudes, this makes it theoretically invariant to speed.

- *inter-motion template velocity invariance* is a consequence of the Bag of Words model: all features from one shot are "bagged" together, regardless of their time coordinate. In effect, speeding up an action merely scales up the $t$ coordinates of the MTs which are simply ignored by the BOW model.

### 5.2.4 Bag of Words

The process is repeated for every frame in the video and all feature vectors are accumulated. We then use the well known Bag of Words model for characterizing the video, hence the name Bag of Motion Templates (BOMT). We first construct a codebook by clustering all features into $k$ clusters using k-means. We then vector quantize every feature vector of one video into one of the $k$ bins, namely the one corresponding to the nearest codeword. The resulting histogram is then L1 normalized forming the final feature vector.

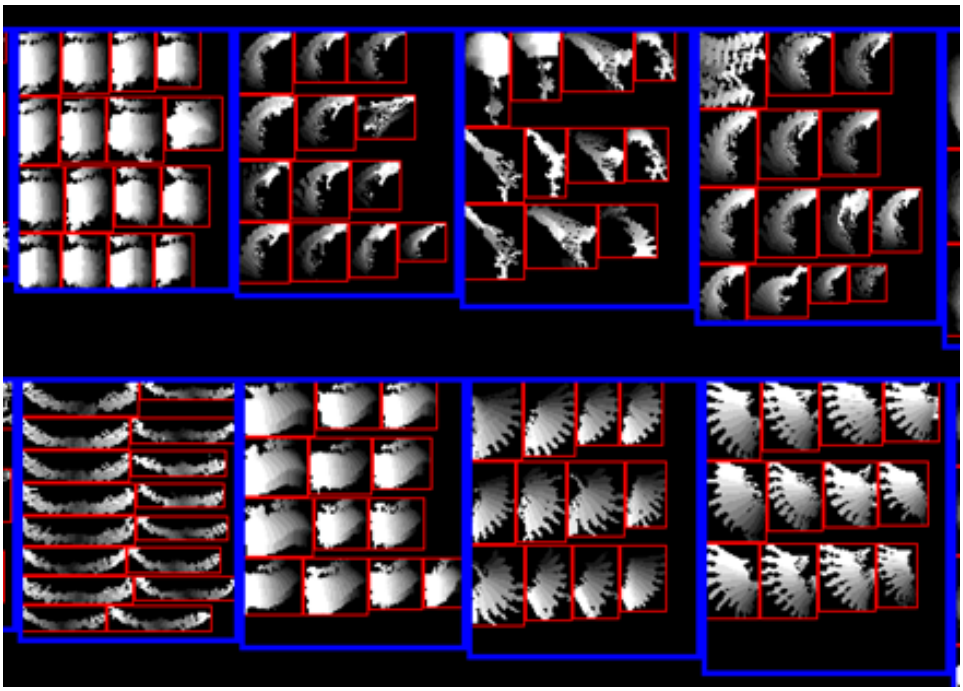

Figure 5.4: Detail of a mosaic showing Motion Templates (red border) grouped into codewords (blue border) extracted from KTH

Figure 5.4 shows examples of the resulting clusters. It can be noted that most of the MTs can be visually interpreted, which as a consequence allows one to mine how a complex action can be decomposed into atomic elements. In figure 5.4 we can clearly see the discriminative action parts that BOMT is able to extract, such as the torso of a person moving left (top-left), hands spreading during clapping (bottom-left), arms lowering and raising during hand waving (3rd and 4th, bottom row)

As a result of feature accumulation, objects that appear throughout the whole video will produce similar MTs that will weigh much more than noisy detections that appear only once or twice in the video. The flip side of this technique is that in short sequences the noise factor becomes proportionally important.

Figure 5.5 shows an overview of the vectorial representation of KTH videos. Each row represents the feature vector for one video. Each column represents a codeword, and pixel intensity represents the weight. The columns have been horizontally re-arranged so that codewords discriminative

for concept 'Handwaving' are the leftmost. It is evident that the first few codewords appear almost exclusively in Handwaving videos. Also, the confusion between 'Running', 'Jogging' and 'Walking' is easily explained since most of their characteristic codewords (which in this image are to the right) appear almost in videos of all 3 actions almost equally.

### 5.2.5 SVM learning

We train our classifiers using non linear Support Vector Machines. For each action we train a one-against-all binary SVM classifier using the exponential $\chi^2$ kernel. SVM learning parameters are optimized using cross-validation. Using Platt's rule implemented in LibSVM [63], we can obtain the confidence values of each classifier as concept probabilities. In the case of multiclass learning, the highest classifier score decides the detected action.

## 5.3 Experimental Setup

We test BOMT on the 4 main action recognition datasets KTH, Youtube, UCF Actions, Hollywood2 and on TRECVID 2010. The characteristics of these datasets and their evaluation metrics are presented in detail in section 3.2.3. The BOMT method requires the tuning of several parameters. In the MHI extraction stage, the lifespan parameter of the MHI $\delta$ should ideally match the frequency of the executed action. In our experiments, we have found the influence of this parameter to be negligible. We set it empirically to $\delta = 1/30$, which means that an updated pixel on the MHI will completely fade in 30 frames. The MT segmentation parameter $\delta_{max}$ described in section 5.2.2 is also set empirically to 30% of the maximum gradient intensity. In the Bag of Words model, the size of the dictionary (codebook) was set to $k = 500$ on the KTH dataset and $k = 5000$ for TRECVID. In the learning stage, for every concept classifier the $C$ and $\gamma$ parameters are optimized by cross-validation by training over half of the training data and testing on the other half. The best performing parameters are then used to train classifier models over the entire training dataset.

## 5.4 Results

In this section we present experimental results obtained according to each dataset's definition in section 3.2.3. We also add the performance of a random classifier for comparison. Additionally, some known motion features are


Figure 5.5: Visual representation of the BOMT feature vectors for KTH videos

| | boxing | handclapping | handwaving | jogging | running | walking |
|---|---|---|---|---|---|---|
| boxing | 97 | 1 | 0 | 1 | 0 | 1 |
| handclapping | 1 | 97 | 1 | 0 | 0 | 0 |
| handwaving | 0 | 5 | 94 | 0 | 1 | 0 |
| jogging | 1 | 0 | 0 | 79 | 13 | 7 |
| running | 3 | 0 | 0 | 23 | 74 | 0 |
| walking | 1 | 0 | 0 | 6 | 1 | 92 |

Figure 5.6: Confusion matrix of BOMT for KTH

tested for reference, where available. The confusion matrix for BOMT on the KTH dataset is shown in figure 5.6. The average accuracy of this classifier is 88.97%.

| dataset<br>classes<br>metric | KTH<br>5 actions<br>avg. acc. | Youtube<br>11 actions<br>avg. acc. | UCF<br>10 actions<br>avg. acc. | Hollywood2<br>12 actions<br>mAP | TV2010<br>50 concepts<br>mAP |
|---|---|---|---|---|---|
| BOMT | 88.97% | 64.36% | 56.92% | 19.96% | 1.51% |
| STIP [20] | 92.1% | 59.1% [64] | 82.6% | 47.4% | 0.99%[1] |
| DT [48] | 94.2% | 84.2% | 88.2% | 58.3% | |
| MoSIFT [46] | 95.8% | 63.1% | | | 2.27% |
| chance | 20.0% | 9.09% | 10.0% | 9.63% | 1.03% |

Table 5.1: Performance comparison between BOMT and state of the art across 5 datasets.

[1]This result is unofficial and uncertain. It comes from an internal IRIM [12] document marking the MAP of all features. The line featuring STIP is noted "bug on test"

The results in table 5.1 show that the descriptor does not outperform the state of the art, but that for the datasets containing web videos (Youtube and TRECVID) performance is comparable to STIP performance. This is a significant result because BOMT is a super-lightweight method that makes minimal use of optic flow (just for image stabilization) and runs several times faster than the other descriptors (see Chapter 5.5 for a quantitative

discussion). Performance is low in UCF and Hollywood2 probably because the video resolution is much higher, which makes for large variability in the size of the motion templates.

## 5.5 Feature fusion in TRECVID and performance considerations

In this section we analyze BOMT performance is using two criteria that are more realistic and relevant to the retrieval problem than the metrics used in the previous section. The first part consists of a performance versus computation comparison to other descriptors: specifically for BOMT and several state-of-the-art descriptors we show the time to extract features from all the KTH sequences and the accuracy of the corresponding classifier. The second criteria is the improvement in MAP over TRECVID obtained using linear fusion. The results show some surprising qualities of BOMT: for slightly lower than baseline accuracy we have a 4× extraction speed and the gain in average precision due to linear fusion is around 5.5%.

### 5.5.1 Speed vs. accuracy tradeoff

In order to illustrate the extraction speed of our descriptor, we compare the CPU time required to extract features from all the KTH dataset (500 sequences, total playback time ∼50 minutes at 25 fps) for our BOMT descriptor with STIP [42], Dense Trajectories [48] and MoSIFT [46], with the corresponding accuracy reported on the KTH dataset.

| descriptor | time | fps | KTH acc. |
|---|---|---|---|
| BOMT | $1h08m55s$ | 18.12 | 88.97% |
| STIP | $5h11m45s$ | 4.01 | 92.10% |
| Dense Trajectories | $5h54m20s$ | 3.52 | 94.20% |
| MoSIFT | $6h45m50s$ | 3.08 | 95.00% |
| playback | $49m57s$ | 25.00 | |

Table 5.2: Comparison on descriptor extraction speed

We can observe from table 5.2 that BOMT is at about 10% less in precision, yet it is computed over 4.5 times faster. In retrieval applications, where the volume of data is 3 orders of magnitude larger than the size of KTH, such a tradeoff may become advantageous. The dimensionality of the

BOMT feature (144) is the lowest of all studied descriptors, so is the density of detections per video. These characteristics translate to small storage and memory consumption, which is also very important in mining and retrieval applications.

### 5.5.2 Improving the MAP via linear fusion

An effective video semantic concept detection system works by fusing different types of features. In this experiment, we perform late linear fusion of BOMT with DSIFT and MoSIFT [46]. Because of the unavailability of validation data, fusion in this experiment is of the optimist type (see annex B for details). The 2010 edition of TRECVID Semantic Indexing was used for this experiment.

One significant problem of BOMT is that many shots are either static or contain too little motion, which translates into null feature vectors. Since these vectors carry effectively no information, they are best removed from both training and testing data. However the linear fusion process requires a valid score for every shot. This forces us to assign a default score to the missing shots in the testing set. Upon experimenting with different strategies, the highest performance both for the individual classifier and in fusion was with a constant default value of $-1$ (or the minimal value in the unnormalized case). The result of the linear fusion experiment on TRECVID data is displayed in the table below.

| DSIFT | BOMT | MoSIFT | fusion | gain |
|---|---|---|---|---|
| | 0.0151 | 0.0227 | **0.0234** | 3.029% |
| 0.0862 | | 0.0227 | **0.0897** | 3.974% |
| 0.0862 | 0.0151 | | **0.0848** | 1.911% |
| 0.0862 | 0.0151 | 0.0227 | **0.0910** | 5.545% |

Table 5.3: Linear fusion of DSIFT, BOMT and MoSIFT on TRECVID 2010 test data

Results confirm that improvement is obtained when fusing features with complementary information. If we consider DSIFT a baseline (as it is common), BOMT improves on it by about 2%, MoSIFT by 4% and both by 5.5%. This result confirms that indeed there is some redundancy between BOMT and MoSIFT as expected.

## 5.6 Other considerations on BOMT

Aside from the main approach described in earlier sections, many different minor and major variations on the BOMT method have been tested. Throughout these experiments non-optimized classifiers are used, causing baseline accuracies to be lower than the final scores for KTH and Youtube shown earlier. The accuracy values should only be considered relatively.

- Before the HOG-like descriptor was used, a first approach involved Hu moments as descriptors. The 7 Hu moments are computed based on spatial moments of a 2D binary image and are discriminative on shapes. However, given the large intra-class variability and the very small dimensionality of the feature, Hu moments performed poorly with a MAP of only 0.0016 and were thereafter replaced with HOG.

- An intermediate feature that grouped MTs together into MT-tracks has been developed. Based on the overlap between MTs in frame $t$ and $t+1$, a simple agglomerative algorithm groups together MTs into tracks. Instead of bagging HOG features of all MTs extracted in the shot, this approach averaged HOG features from all MTs in one track and passed the average HOG values as input for the BOW model, creating essentially a "bag of tracks of motion templates". By eliminating tracks that are too short some of the parasite MTs can be eliminated. The only parameter of the method was the maximum allowed length of a track before splitting. However, at infinity the performance is 0.34344, and as the parameter value reduces, performance increases. Maximal value of 0.4960 is for length 1, which corresponds to the normal BOMT.

- The influence on training and testing with flipped videos increases the performance from 0.57 to 0.6 in Youtube but doubles training time. Flipping is performed after the MTs are extracted: HOG features for the original MT and a flipped version of the MT are computed and pooled together. We have not experimented with flipping a training video and duplicating the label because of the expected computational cost of doubling training samples. Testing is done by max-pooling the scores for the initial and flipped test video. Since this experiment the technique has become part of standard BOMT.

| mean accuracy | train normal | train normal+flipped |
|---|---|---|
| test normal | 0.5711 | 0.5564 |
| test normal+flipped | 0.5631 | 0.6028 |

- Concatenation (a.k.a. feature level fusion) of each HOG feature extracted from the MT with a HOG feature of the grayscale image from the same ROI as the MT yields a precision of 0.6145 on Youtube but fails to improve on any other dataset. Classifying MT-HOG and image-HOG separately and adding the scores gives very good results on Youtube (0.6436) but lowers performance on the other datasets as well.

- In an attempt to isolate the source of error in feature extraction, a subset of the Youtube dataset containing only videos without camera motion has been created. This new dataset had 449 out of the initial 1600 videos, which was insufficient for classification on 3 actions: *biking*, *horse riding* and *golf swing*. The classification could only be performed on the remaining 8 actions. An improvement of 7% was detected as a result of removal of camera movement.

| mean accuracy | 8 actions | 11 actions |
|---|---|---|
| static videos | 0.5843 | not enough training data |
| all videos | 0.5110 | 0.5043 |

- There are bounding box annotations for the actions in the Youtube dataset. By extracting MTs from the MHI at the corresponding bounding boxes, we avoid the imprecise MT segmentation step. The mean accuracy increases this way from 0.5073 to 0.6976 (or 0.6362 if bounding boxes are used only in testing), thus proving that the MT segmentation step is one of the sources of error in the classifiers. Unfortunately bounding box data is only available for the Youtube dataset.

Also, a number of interesting statistics on KTH and Youtube datasets have been made.

- The following figure shows a 2D embedding of the KTH feature vectors obtained using LDA. From this representation it is easy to observe the confusion between *walking, jogging* and *running* samples, also visible in figure 5.5 (note how the last 3 actions have very similar patterns).

- In order to explain the performance, the *mean intra-cluster variance* has been computed on KTH and Youtube datasets. Every HOG feature gets assigned to one codeword based on the nearest-distance to its center. By accumulating all HOG features for one codeword, we can compute the variance of this data, which tells us how varied are the

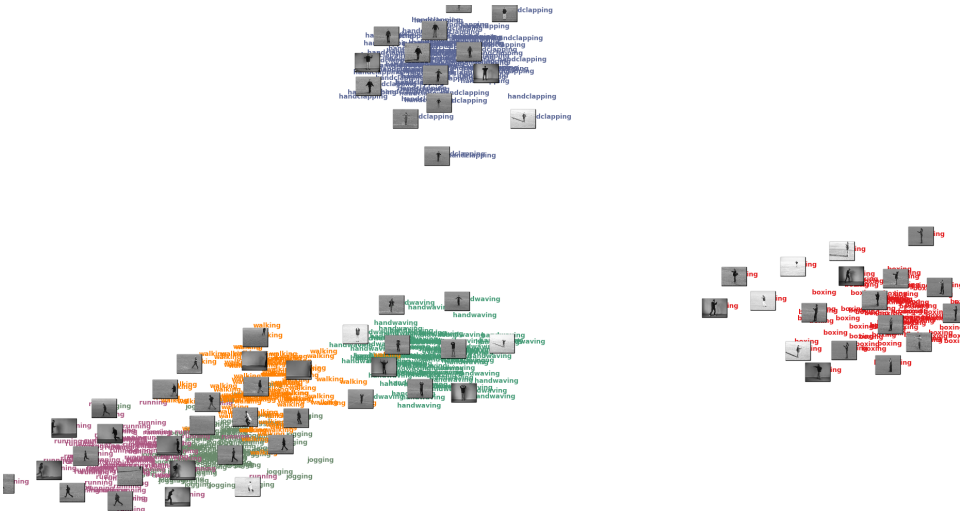

Figure 5.7: 2D projection of KTH vectors using LDA.

HOG features therein. The mean of these variances is the mean intra-cluster variance. The surprising result is that even if the Youtube dataset has more concepts, contains more objects, and has significantly more noise, there is less intra-cluster variance than in KTH, for the same number of clusters.

| k | KTH | Youtube |
|---|---|---|
| 500 | 0.0141 | 0.0128 |
| 4000 | 0.0103 | 0.0098 |

- Taking the cluster analysis further, cluster-specific action entropy was computed on the 500 visual words. This is a measure on information about action labels based on the weight of one cluster/codeword in all the videos. In the following chart, entropy values have been sorted. Note that many of the most informative KTH clusters have zero entropy: this means we can perfectly determine the action from the value of the corresponding codeword. However, even for Youtube the lowest uncertainty cluster has 1.5 bits entropy (1.5 bits of information loss out of $log_2(11) = 3.45$ bits of theoretical self-information). This can be an indication that the clustering process was not as successful on Youtube as on KTH, thus justifying the big difference in accuracy.

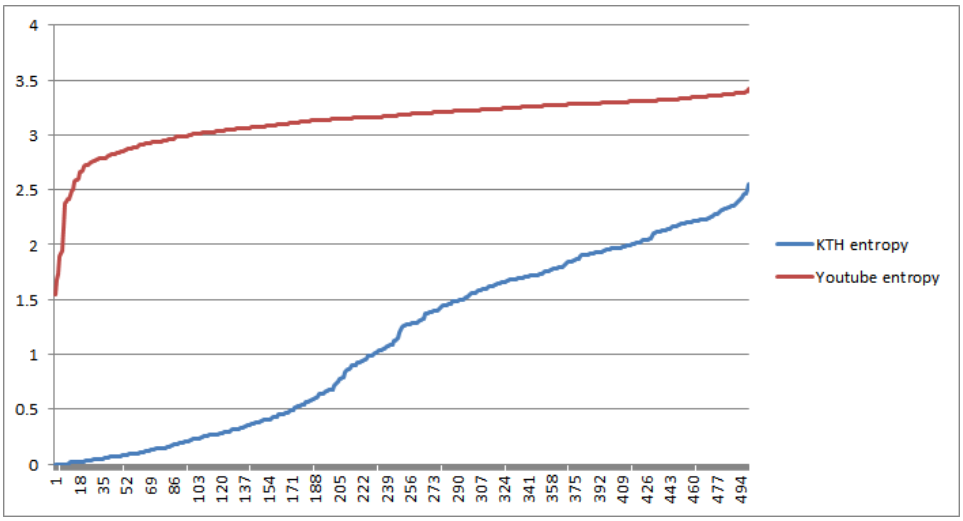


Figure 5.8: Concept entropy for each codeword in KTH and Youtube clusterings.

## 5.7 Conclusions

We have proposed a video action description method based on the Motion History Image that is computationally efficient in both speed and memory, thus making the inclusion of motion descriptors on the TRECVID dataset more approachable. In comparison with state-of-the-art descriptors, BOMT performs within more than 96% of STIP accuracy on KTH, with precision similar to STIP on Youtube and TRECVID, while taking only 22% of the time needed to compute STIP. While STIP is thus more accurate, our descriptor is more suitable for TRECVID since it bears almost the same results in a much shorter computation time. Results also indicate that BOMT can significantly increase the MAP of a TRECVID system by using linear fusion. BOMT presents a Motion History Image visual representation of the action that is easy to understand and interpret by a human, thus allowing the visual codebook to be visually inspected. Results confirm that indeed a method for action categorization can be used to successfully detect semantic concepts in a large collection like TRECVID. We believe this finding justifies the use of motion features in concept detection. In the perspective of a large-scale video retrieval framework, where the combined precision of many features, computation time and memory consumption are more important than individual feature performance, BOMT may represent a practical solution. It

is our opinion that BOMT's fast computation and low memory and storage far outweigh the relatively lower performance in action categorization and concept detection.

# Chapter 6

# Adding motion to DSIFT-BOW

We have previously seen that semantic concept detection in large scale video collections is mostly achieved through a static analysis of selected keyframes. A popular choice for representing the visual content of a keyframe image is based on the pooling via bag of words of local descriptors such as Dense SIFT [17]. However, simple motion features such as optic flow can be easily extracted from keypoint positions in the keyframe. In the following chapters we propose efficient enhancements to the DSIFT-BOW approach by exploiting information derived from optic flow. In chapter 6.1 we use the dense optic flow values to categorize DSIFT keypoints as static or dynamic. In chapter 6.2 we take the idea further and develop the *Z* features*, which quantize the motion of the DSIFT keypoints based on direction and orientation. Different learning strategies for these new features are experimentally evaluated, and their relevance in concept retrieval is tested by combining them with the established TRECVID descriptors, resulting in significant performance improvement.

## 6.1 Separating BOW histograms into static-dynamic zones

The traditional Bag-of-Words approach involves extracting local visual features such as SIFT from a video keyframe. More recent research suggests that dense extraction of visual features ultimately gives better performance [20] than interest points. The extracted features are clustered using K-means, with a large enough value of $k$ (typically hundreds or thousands) and the resulting cluster centroids form a codebook. Ultimately, videos are characterized by quantizing the extracted features into a histogram of codewords by assigning each feature to its nearest codeword. If the chosen descriptor is SIFT, this method is informally known as Dense SIFT or DSIFT. The overlapping zones around keypoints are referred to as *patches*. In this chapter, we employ simple motion estimation techniques to compute a motion mask at keyframe level. This motion mask allows us to separate static SIFT keypoints from SIFT keypoints in motion and deal which each of these classes separately. We propose a method based on the construction of two separate Bag-of-Words histograms for the static and dynamic classes with concatenation of the resulting feature vectors. The contributions presented in this chapter are the following: firstly, we are proposing an enrichment of the DSIFT descriptor by adding motion information. We show how to extract, quantize, normalize and classify the features in order to obtain relevant concept classifiers. In the process, 3 static-dynamic separation methods and 3 normalization techniques are tested. Secondly, we show that linear fusion of several flavors of the newly obtained descriptor can outperform individual descriptor performance and thus help increase the performance of the retrieval system. Work in this chapter has been published in [120].

### 6.1.1 Related work

Traditionally concept detection in TRECVID is done as a fusion of numerous classifiers, mostly visual, but also audio and metadata-derived [4]. One visual descriptor used in the overwhelming majority of TRECVID submissions is David Lowe's SIFT [17]. In its original iteration SIFT was both an interest point detector and descriptor. The extrema values of a difference of gaussians at different scales was used to detect keypoints. However, more recent results [20, 121] in semantic indexing show that dense extraction may give better results than keypoints, especially in a large data context such as TRECVID. The extracted features are then processed according to the Bag of Words paradigm.

Our work bears some resemblance to SIFT Flow [122], in that displacements of SIFT patches are extracted, but while their method estimates the apparent motion between images representing different scenes, we use "real" motion information from video for describing the image content. Chen's MoSIFT [46] is an extension of SIFT that adds in a similar feature where the gradient is replaced by optic flow.

Since motion segmentation is an integral part of the system, we present here some of the known tehniques: basic methods for obtaining a foreground mask are frame differencing, optic flow thresholding and background modeling [123]. Frame differencing is fast but shows very little robustness to uniform objects, textureless zones and illumination changes. Background model methods are far more reliable but demand a stationary camera and require a longer sequence. Optic flow is a good compromise in speed and performance because it requires only a pair of frames instead of a full sequence and gives relatively accurate estimates on the displacement.

### 6.1.2 Obtaining the Motion mask

One way to simulate a stationary camera at frame level is to compensate for camera motion between frames. Again we use the stabilization method described in annex A. This method does dominant motion compensation by estimating a homography with RANSAC over detected feature correspondences. This homography is then used to produce a synthetic motion vector field that models the camera movement which is used as an initial estimate for the full-frame optic flow using Farneback's method [124]. The displacement between the synthetic background motion field and the actual motion field can then be used as an estimation of foreground objects. One simple way of doing this is by thresholding the flow magnitude in each pixel. If the flow magnitude is higher than a predetermined threshold, the pixel is in motion and is considered foreground. In figure 6.1 green dots are placed at the position of the SIFT keypoint. The dots color intensity is proportional to the amount of motion.

### 6.1.3 Codebook construction

The standard approach when classifying using bag-of-features involves the construction of a codebook. This is usually achieved by clustering the features using the K-means algorithm using euclidean distance. Each resulting cluster centroid is considered a codeword. Here we follow the standard approach in computing the codebook. We empirically choose a codebook size of

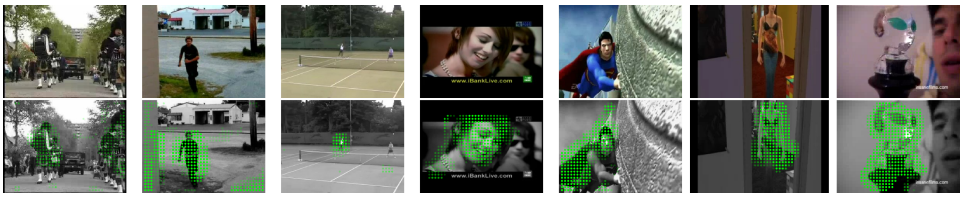

Figure 6.1: Examples of motion masks extracted using optic flow.

$k = 500$, which we found as a reasonable compromise between performance and speed.

### 6.1.4 Static-dynamic separation

The core of our approach is in the separation between static and dynamic patches. We do this separation based on the corresponding value of compensated optic flow, which we obtain earlier for the motion mask. Figure 6.2 shows how this separation influences the final feature. We experimented with different variants of separation.

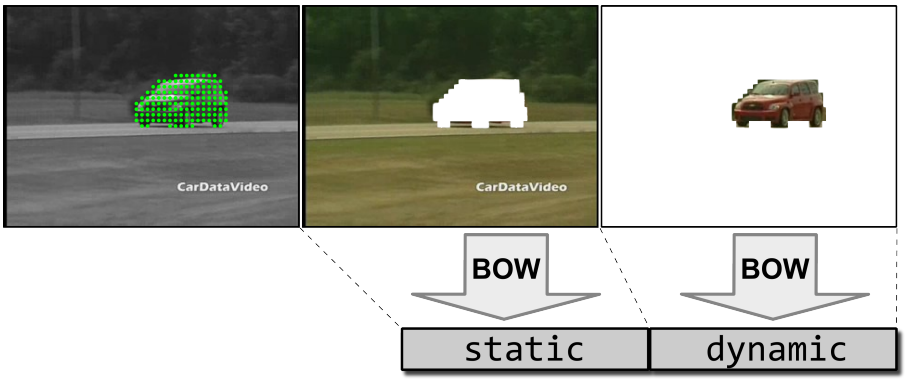


Figure 6.2: Separation between static and dynamic patches followed by the Bag of Words model applied separately on each of the 2 resulting sets of patches. The resulting vectors are concatenated into the final feature.

A collection-wide statistic on flow magnitude in all keypoint positions gives us a distribution of flow velocities. We exclude from this statistic points with zero velocity flow since these points dominate the distribution without providing any information, and because failed optic flow estimation

results in zero flow as well. The $\theta_1$ median of this distribution would be the value that divides keypoints with slower motion and keypoints with faster motion into two equal sets.

The decomposition of the initial DSIFT-BOW feature $DSIFT_i$, with $i = 1..k$ and $p$ describing a keypoint detection is described by equation 6.1. The $\delta$ signifies the Kronecker delta. It is easy to verify that $static_i + dynamic_i = DSIFT_i$. The focus of the research in this chapter is in the choice of the static and dynamic weights $w_s, w_d$ for a keypoint $p$. As shown below, the weights are functions of the keypoint's optic flow magnitude $v_p$. The resulting feature is the concatenation:
$[static_1, \cdots, static_k, dynamic_1, \cdots, dynamic_k]$

$$\begin{aligned} DSIFT_i &= \sum_p \delta_{i,code(p)} \\ static_i &= \sum_p w_s(p) \cdot \delta_{i,code(p)} \\ dynamic_i &= \sum_p w_d(p) \cdot \delta_{i,code(p)} \\ &\text{with } w_s(p) + w_d(p) = 1 \end{aligned} \tag{6.1}$$

The three separation strategies, as seen in figure 6.3 are:

- When constructing our feature vector, instead of quantizing all DSIFT features into one occurrence histogram, we separate the features based on threshold $\theta_1$ and construct two histograms separately: one for the "slow" flow, which we call static, and one for the "fast" flow which we call dynamic. This is the first separation strategy.

- A second separation strategy uses the same rule, but with the threshold at a minimal level $\theta_2$. This causes more patches to be considered dynamic, and only the patches that are certain to be stationary as static. This choice was motivated by the fact that too few dynamic patches were selected

- The third strategy involves soft assignment. Using a fixed threshold value means that patches with a velocity close to threshold level may fall either on the static or dynamic part, which creates noise. This can easily be avoided by making a soft assignment instead of a binary static/dynamic choice. We achieve this by assigning each patch a *static weight* $w_s$ and a *dynamic weight* $w_d$ with $w_s + w_d = 1$. The flow

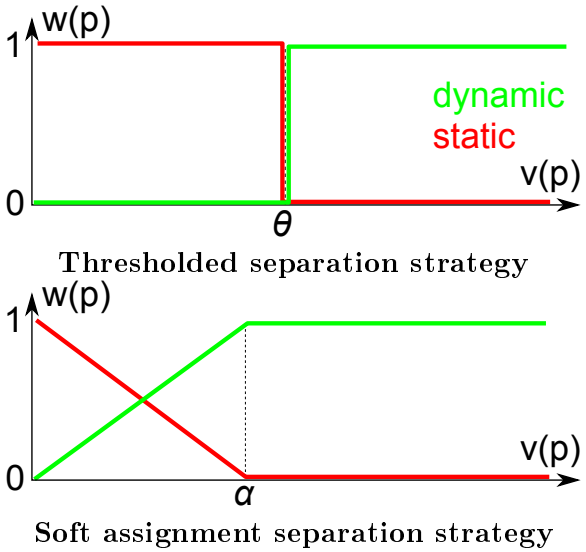


Figure 6.3: The functions defining the static and dynamic weight with respect to flow magnitude

velocity $v$ is used to directly determine the dynamic weight by using a clipped ramp function (see figure 6.3). The value of the $\alpha$ parameter is empirically fixed.

### 6.1.5 Normalization

The usage of dense feature sampling causes the number of features sampled in each keyframe to be constant (assuming all videos have the same resolution). This makes for an implicit L1 normalization of the resulting histograms: the sum of the histogram will be equal to the total number of features in the shot, which will be constant for all shots. However, since the principle of our approach is to separate between static and dynamic zones in the image, the total number of features in each zone will vary, thus a special normalization technique is required. We compare 3 normalization strategies, as seen in figure 6.4

- The first and most simple is the L1 normalization of the static-dynamic concatenated vector. Each element of the feature vector is divided by the total sum of the vector.
- The second normalization method normalizes the static and dynamic

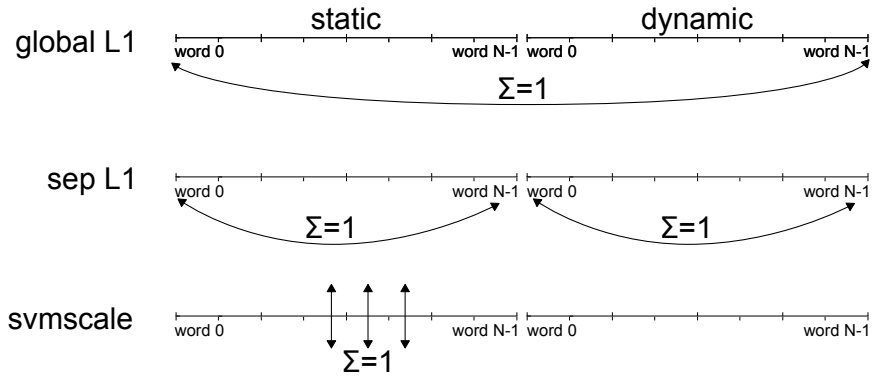


Figure 6.4: Normalization strategies for the static-dynamic separation

histograms separately. Each histogram is separately normalized using L1 norm and the resulting histograms are then concatenated.

- The third method is component-based normalization, which we call "svmscale" according to LibSVM [63]. This method works by scaling min-max and then normalizing components instead of features. Each component of the feature vector is divided by the total sum of that component over all vectors.

The separation and normalization described are reasonably fast, taking in average 1.48 seconds CPU time per shot.

### 6.1.6 Classification and fusion

We use LibSVM to train concept classifiers. We use exponential $\chi 2$ kernels, that have proven in practice to give state of the art results in visual feature based classification. Specifically, we use a modification of the binary C-SVM implementation in LibSVM. We optimize the learning parameters $C$, $\gamma$ ($\chi 2$ kernel width) and weight of the positive class through brute force search. For comparison we add the "default" DSIFT approach, which simply bypasses the separation stage.
We also experiment with the simple linear fusion technique described in annex B. This is done by combining the score of each classifier using a weighted sum, with weights summing to one. Since the computation of weighted sums and of the mean average precision are almost instantaneous, a grid search on the weight values is possible. We experiment with the fusion of 4 best performing variants, as described in the next section. Each weight is tested in 0.1 increments.

| *Mountain* | default | trhesh=$\theta_1$ | thresh=$\theta_2$ | soft |
|---|---|---|---|---|
| globalL1 | 0.13164 | 0.0297 | 0.0581 | 0.0316 |
| sepL1 | N/A | 0.0045 | 0.0173 | 0.0003 |
| svmscale | 0.2699 | 0.2866 | 0.1808 | 0.2605 |
| *Running* | default | trhesh=$\theta_1$ | thresh=$\theta_2$ | soft |
| globalL1 | 0.02351 | 0.001 | 0.0235 | 0.0134 |
| sepL1 | N/A | 0.001 | 0.0235 | 0.0009 |
| svmscale | 0.01811 | 0.03843 | 0.0198 | 0.0474 |

Table 6.1: Average precision of the different separation and normalization techniques for two concepts

### 6.1.7 Experimental Results

In the TRECVID 2010 training dataset, the threshold values for flow are $\theta_1 = 14$, $\theta_2 = 1$, $\alpha = 5$. As described in section 6.1.4, $\theta_1$ is a corpus-specific median value that splits the set of displacements into equal static and dynamic sets. Lowering the $\alpha$ parameter would essentially lower performance by approaching the $threshold = \theta_2$ situation, whereas excessively increasing it would lead to unbalancing between the static and dynamic classes. $\theta_2$ should function as a noise threshold, and is chosen based on the precision of optic flow detection.

If we count the 'default' run, there are 11 valid combinations between separation 4 strategies: default (no separation), medium threshold, low threshold, and soft assignment and 3 normalization methods: global L1, separate L1 and svmscale. We exemplify the resulting combinations on 2 concepts: predominantly static concept "Mountain" and a predominantly dynamic "Running". Table 6.1 shows these results.

In a first experiment the general impact on performance is measured by averaging precision over 50 concepts of the TRECVID 2010 test data. Since thresholding methods and the sepL1 consistently gave lower performance (see table 6.1) we have done this only with two of the above 4 separation strategies: 'default' and 'soft' and two out of 3 normalization methods: 'globalL1' and 'svmscale'. The final run is the best performing linear fusion of the 4. The resulting average precisions are summarized in the following table:

Note that "default globalL1" actually means the standard DSIFT. It is clear to see that both soft assignment and svmscale normalization improve on the initial approach. The late fusion experiment improves performance

| run | default-globalL1 | soft-globalL1 | default-svmscale | soft-svmscale | fusion |
|---|---|---|---|---|---|
| MAP | 0.0615 | 0.0637 | 0.0651 | 0.0701 | 0.1067 |

Table 6.2: Mean average precision on 50 concepts

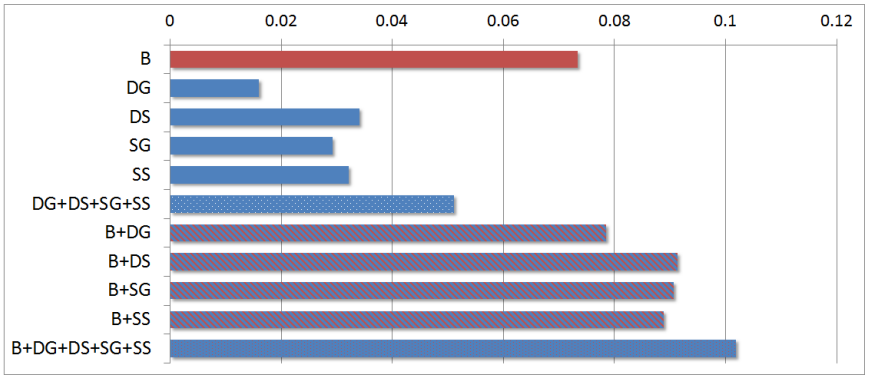


Figure 6.5: Baseline fusion for TRECVID 2010 Lite - 10 concepts

significantly: in average each concept is improved by 10.05% over the best single classifier out of the 4 variants. These improved classifiers give a final mean average precision of 0.106 which is an improvement of 68.33% over DSIFT.

In the second experiment, fusion performance is measured against a EURECOM submission to the Lite SIN task of TRECVID 2010. This submission has a MAP of 0.0733. The Lite task contains 10 concepts instead of 50, therefore results from this experiment and the previous are incomparable. Figure 6.5 shows the result of several combinations of the baseline with the 4 features. The meanings of the notations are: $B = baseline$, $DG = default - globalL1$, $DS = default - svmscale$, $SG = soft - globalL1$ and $SS = soft - svmscale$. While the DSIFT feature ('default-globalL1') has a MAP of 0.0159, the highest MAP is for the fusion of the baseline with all 4 features for a MAP of 0.1018.

### 6.1.8 Conclusions

It is easy to observe from table 6.1 that separate L1 normalization is not beneficial. The explanation could be that keyframes containing very little motion have few dynamic patches. Separate L1 normalization artificially

boosts the weight of these dynamic patches so that overall, static and dynamic end up having the same weight.

Results confirm that the best combination is soft assignment separation with svmscale normalization. We conclude that this is due to the fact that component based normalization deals with static and dynamic features indiscriminately. Also, svmscale mitigates the problem of having very little motion information.

The differences in MAP between the two experiments on 10 and 50 concepts are notable: by increasing the number of concepts both the overall classifier MAPs increase for all features (in average by 2.53× ), and the fusion to DSIFT ratio increases from 1.73 to 3.2. This is an indication of the generality of the approach and shows that our method is suitable for large scale concept detection.

We have shown in this chapter that straightforward motion analysis methods can significantly improve the performances of established visual descriptors. We have proposed the usage of 3 static-dynamic feature separation strategies, as well as 3 normalization methods for the resulting features. Thus, using simple and fast motion estimation methods and with the help of efficient linear classifier fusion we increase the MAP of DSIFT from 0.061 to 0.106. Additionally, according to the fusion with the baseline we have shown that when combining a real life baseline system with one of our 4 features we can increase the MAP of the system by around 20% and when combining all 4 features together with the baseline we obtain an improvement of 38%.

## 6.2 Z* features

In the previous chapter we showed that by separating static and dynamic DSIFT patches based on the intensity of their optic flow we added significant new information to the classifier. It is however intuitive that direction of movement - more than mere presence - conceals even more information. In this chapter we try to exploit directional information by developing relevant new descriptors which we call *Z* features*. Using a simple binning technique, we create 3 features named ZN, ZHV and ZLRUD that capture not only the codeword information stored in DSIFT-BOW histograms but also the quantity and direction of movement of the SIFT patch.

The addition of our features to an existing concept detection system based on DSIFT features comes with the relatively low cost of extracting sparse optic flow from one keyframe per video shot. By using the well-known homogeneous kernel map [65] method we can efficiently approximate the non-linear $\chi 2$ kernel and train linear SVMs in a fraction of the non-linear SVM computation time: classification takes in average around one second for a feature matrix of 119,685 shots $\times$ 2500 components. By using a linear score combination, we combine TRECVID submission run and with our Z* features and obtain an increase in Mean Average Precision (MAP) of about 5%. By applying an official TRECVID tool that compares submission runs based on randomization testing, we are able to confirm that the aforementioned improvement is statistically significant. A paper on Z* features has been published [125].

### 6.2.1 Related Work

The traditional concept detection in video, as shown by works published in TRECVID workshops [4], uses keyframe techniques. One keyframe is selected from the shot in question and all subsequent processing deals with the keyframe as sole representative of the shot, much like a CBIR system. Compared to the few existing spatio-temporal content descriptors [42, 46], this approach is hugely more efficient in time and memory. The disadvantage is that obviously all the motion and sequence information is lost.

Of these keyframe methods, the Bag of Words (BOW) technique has been prevailing in TRECVID for many years. Although newer methods like Fisher vectors [126] and super-vectors [29] supersede BOW, it remains widely used in the community. In the vast majority of situations, this technique uses SIFT [17] as the visual feature, whether using the original interest point detection or by dense sampling [127]. Recent work seems to suggest that in

this context the dense extraction seems to slightly outperform interest point methods [20, 121, 128].

There are several local features called spatio-temporal descriptors that describe image sequences, most notably used in the action recognition community. One notable example is Laptev's Spatio-Temporal Interest Point descriptor [42], which detects 3D interest points using a 3D extension of the Harris operator and describes them using histograms of oriented gradients and histograms of flow (HOGHOF). Wang's dense trajectories descriptor [48] extracts and tracks features throughout the entire video volume. Chen's MoSIFT [46] is an extension of SIFT that adds in a similar feature where the gradient is replaced by optic flow. Since all these methods analyze a 3D volume instead of a 2D image, the computational complexity is much higher than the standard keyframe approaches, to the point that implementing a spatio-temporal technique on TRECVID might prove too computationally heavy for some systems. By comparison, our feature computes the optic flow on a single keyframe in the video. The extraction of DSIFT, codebook assignment and motion feature construction take in average 8.27 seconds for any TRECVID video, while any of the 3 descriptors mentioned earlier take in average well beyond 30 seconds to compute, depending on video length.

Our approach shares some similarity with part of the work of Wang et. al. [129] in that codeword motion is considered. Two important aspects differ. Firstly, the input features come from dense sampling in our work and from keypoint detections in [129]. Features computed at keypoints will extract information from salient zones in the image and will ignore uniform or weakly-textured zones. They concentrate on details so that they are better suitable at object recognition rather than detection. Dense sampling ensures that every pixel in an image is covered by at least one patch, which makes it less likely to miss an object. This leads to the background forming an important integrating element of the model. Also, in dense sampling more features are selected and variability is higher. In short, keypoints are better for precision, dense is better for recall. The second difference is in the way motionless patches are processed. In [129] an orientation histogram is built using projections that cumulate in each bin. Patches with small amounts of motion contribute little to the resulting orientation histogram. Our features contain a Z (zero) bin, which stores the number of patches with little or no motion, which means that the information on static codewords is not lost.

It should be noted that after the publication of our papers [120, 125] that describe the activity in this chapter, Liang et.al. [130] proposed a velocity pyramid that essentially extracts features similar to the Z* features at different pyramid levels. Combining their velocity pyramid with the classical

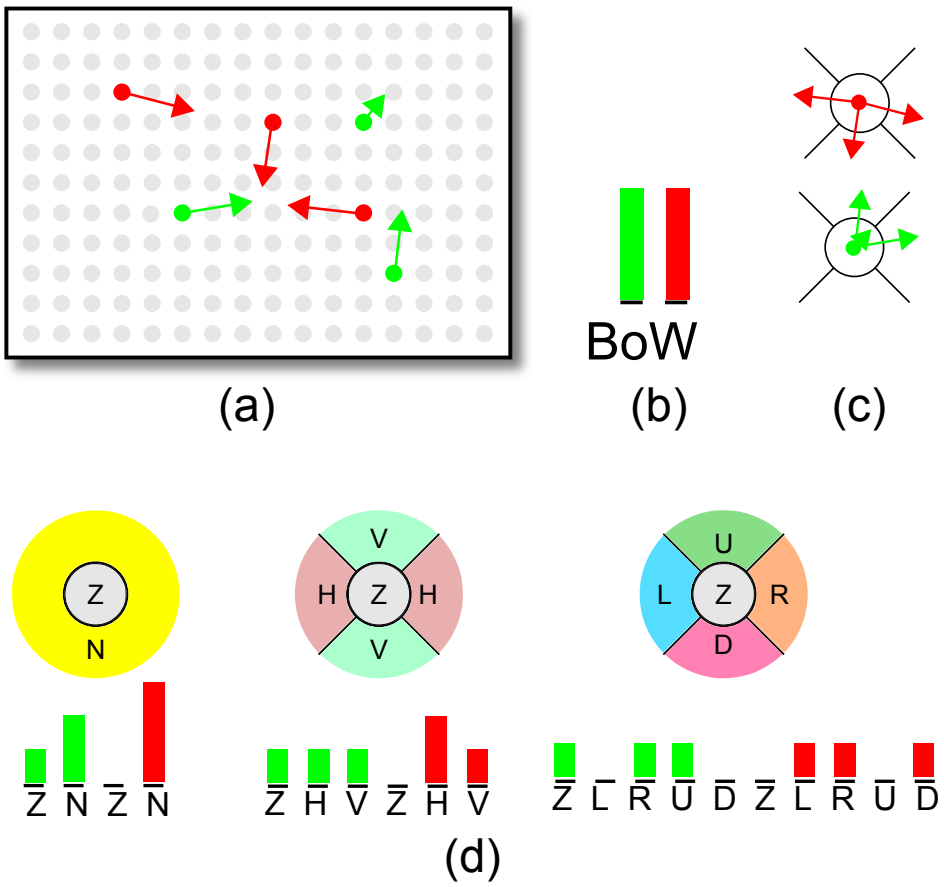


Figure 6.6: Example of Z* feature construction

spatial pyramid [27] they report improvement of 13% in the MED task of TRECVID 2013[1].

### 6.2.2 Extracting codeword motion

Dense SIFT works by extracting features from evenly spaced keypoints. In our version we use a grid of size 8 pixels. The densely extracted SIFT features are quantized and assigned to one of the $k = 500$ codewords. The

[1] http://www-nlpir.nist.gov/projects/tvpubs/tv13.papers/tokyotechcanon.pdf

value of $k$ has been empirically chosen as a good compromise between performance and computation speed. In the normal DSIFT, keypoint locations are ignored and only the total count of codeword occurrences are taken into consideration. However we keep for each keypoint position $x, y$ the index of the codeword $c$.

In order to extract motion, we first access the keyframe in the video file. We then advance by a small time interval and extract a second frame corresponding to slightly later time in order to have sufficient difference. The motion between these frames is subjected to a mostly uniform background camera movement that can be compensated for. We use the method in described in annex A.

This homography is then used to produce a synthetic motion vector field modeling the camera movement which is used as an initial estimate for the full-frame optic flow using Farneback's method [124]. The displacement between the synthetic background motion field and the actual motion field can then be used as an estimation of foreground objects, since motion in background areas is compensated for. Having computed the motion compensated flow, we can now sample it at the keypoint positions $x, y$. Thus, for every keypoint $i$ we now have its coordinates $x_i, y_i$, a codeword value $c_i$ and the optic flow $f_i^x, f_i^y$.

An overview of our method is presented in figure 6.6: first DSIFT patches assigned to codewords (green and red) are extracted (a) along with their optic flow. The classical (b) DSIFT-BOW histogram is built by counting the occurrences of each codeword. On our approach (c) motion vectors are grouped by codeword and (d) quantized by creating histograms for every bin and every codeword by counting the number of flow vectors in each bin.

### 6.2.3 Spatial Codeword Motion Histograms

We now group our features by codeword. For each codeword $c$, we quantize the flow coordinates $f_i^x, f_i^y$ according to the spatial histograms in figure 6.6. The bin corresponding to the region where the $(f_i^x, f_i^y)$ point falls is incremented. The zero (Z) bin will capture features with zero or small motion. The value of the Z bin radius $\theta$ has been chosen as the median of all the optic flow velocities in the collection in order to ensure the balance between the number of features falling inside and outside of Z (which is the non-zero bin N). Since there is an intuitive conceptual distinction between horizontal and vertical movement, we separate our space in 2 corresponding bins. Horizontal and vertical bins H and V take advantage of origin symmetry, are spatially discontinuous and quantize feature orientation. Left, right, up

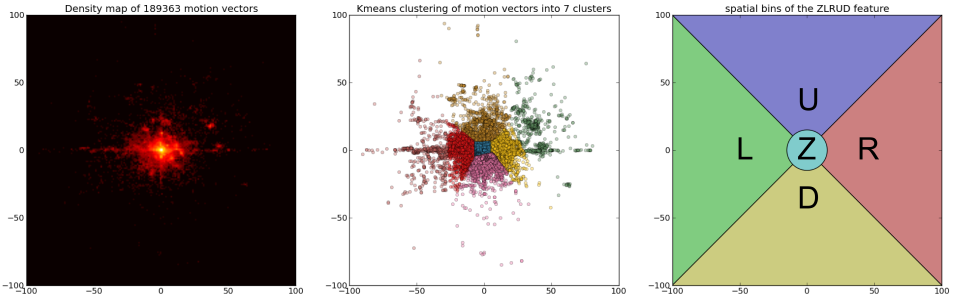


Figure 6.7: Result of K-means clustering of motion vectors compared to ZLRUD spatial arrangement.

and down bins L, R, U and D separate motion direction (bearing). The 3 features we are proposing are:

1. ZN which describes *whether the codeword moves or not*
2. ZHV which discriminates between codewords moving horizontally and vertically, thus contains information on *orientation*, and
3. ZLRUD which discriminates codewords moving left, right, up and down, therefore encodes the *direction*

Since there are several spatial bins for each codeword, the final feature size will have size $2 \times K$ for the ZN variant, $3 \times K$ for ZHV and $5 \times K$ for ZLRUD. Since Z features are decompositions of the DSIFT features, the relation between these bins is given by the following formula:

$$DSIFT = Z + N = Z + H + V = Z + L + R + U + D \tag{6.2}$$

The baseline DSIFT BOW approach works by directly counting codeword occurrences, which makes it equivalent to a single bin covering all the space. All resulting histograms are normalized using the L1 norm.

In figure 6.7 a density plot of all the extracted motion vectors shows the very high concentration around the origin, corresponding to slow movement or static patches. By clustering the $f_i^x, f_i^y$ coordinates of the motion vectors using K-means, we see that they roughly organize in a pattern similar to the ZLRUD spatial histogram.

### 6.2.4 Classification and Fusion

The experimental setup closely follows the TRECVID 2012 Semantic Indexing evaluation. We are evaluating 50 concepts, with sparse training anno-

tations available on a development set containing 400,289 sequences, and applied on a test set of 145,634 sequences.

In order to benefit from the superior classification power of non-linear kernels, we employ kernel approximation techniques described in [65]. In practice, the $\chi 2$ kernel seems to perform very well when using Bag of Words features. We map our features using the homogeneous kernel map of order $N = 3$, implemented in the Scikit-learn library [66] which yields a new feature dimensionality of $7\times$ the original one. The new feature vectors can be used to train a linear SVM, which will approximate the non-linear $\chi 2$ SVM classifier. For that we use the Liblinear [67] implementation found in Scikit-learn by training on half of the development set (197,185 sequences) and cross-validating on the other half in order to optimize the $C$ parameter of the SVM. After finding the optimal value of $C$, we train another linear SVM with this value of C on the entire development set and test on the testing set. The classifier confidence values found in the validation set are kept for later use in the linear fusion. We apply this procedure for DSIFT, ZN, ZHV and ZLRUD features. As it is routinely done in TRECVID, this process is done using training annotations from one of the 50 concepts at a time. Non-annotated shots are not part of the training set, so the number of training samples varies between concepts. Using the SVM confidence values and the ground truth on the testing set we compute the Average Precision for each concept. The run is finally evaluated by averaging these average precisions (MAP).

Classification scores from the different features are then combined using late fusion, as described in annex B. This is done by finding the linear combination of score weights that maximize the MAP on the validation set and reapply these weights on the testing set confidence values. Since the computation of weighted sums and of the MAP are almost instantaneous, a grid search on the weight values is possible. We experiment with the fusion of the baseline, DSIFT, ZN, ZHV and ZLRUD. Each weight is tested in 0.1 increments.

We use the TRECVID randomization testing [4] to estimate the statistical significance of the increase in MAP. Each test implements a partial randomization test of the hypothesis that two search runs, whose effectiveness is measured by MAP, are significantly different - against the null hypothesis that the differences are due to chance. We use this approach to pairwise compare DSIFT, ZN, ZHV, ZLRUD and the fusion result.

### 6.2.5 Experimental Results

Table 6.3 shows the fusion MAP for the classifier DSIFT, ZN, ZHV and ZLRUD and a TRECVID 2012 submission of MAP 0.1798. Although more information is contained within the Z* features than in DSIFT, their overall MAP is lower. However the MAP comparisons are not conceptually sound since the features have different dimensionalities ($DSIFT = 500, ZN = 1000, ZHV = 1500, ZLRUD = 2500$). A more robust comparison would have been for instance DSIFT with a codebook of size $k = 2500$ to ZLRUD, but such a comparison would require calculating DSIFT features and Bag of Words for both $k = 500$ and $k = 2500$, which would defeat the purpose of this work. Note how the fusion between just the Z* features does not show improvement, probably because there is no complementarity between the Z features. Fusion with a lower MAP than it's constituent features is possible when the optimal weight vector on the validation set is very different from the actual optimal weight vector on the test set.

| baseline | DSIFT 500 | ZN 1000 | ZHV 1500 | ZLRUD 2500 | fusion | gain |
|---|---|---|---|---|---|---|
| 0.1798 | 0.0985 | 0.0926 | 0.0844 | 0.0779 | 0.1898 | 5.55% |
| 0.1798 | | 0.0926 | 0.0844 | 0.0779 | 0.1882 | 4.63% |
| | 0.0985 | 0.0926 | 0.0844 | 0.0779 | 0.1008 | 2.37% |
| | | 0.0926 | 0.0844 | 0.0779 | 0.0917 | -0.95% |

Table 6.3: Mean average precision of the 4 features and fusion with a strong baseline in different combinations on TRECVID 2012

Figure 6.8 shows the weight of each feature in fusion and can be interpreted as an indication of what type of movement is the most informative for classifying the concept, e.g. if the DSIFT component has a high weight, then the concept is more easily classified based only on static visual information. High ZN weight means that the presence or absence of movement is a good cue for detecting the concept. ZHV has high weight if the direction of movement is important and ZLRUD is high when both direction and orientation of movement are relevant.

The TRECVID randomized test for statistical significance has confirmed that for a significance level of 0.05 the fusion run statistically outperforms all of the features. The conclusion of said test is that the improvement of the MAP from 0.1798 to 0.1904 is in fact a statistically significant improvement and not due to chance.

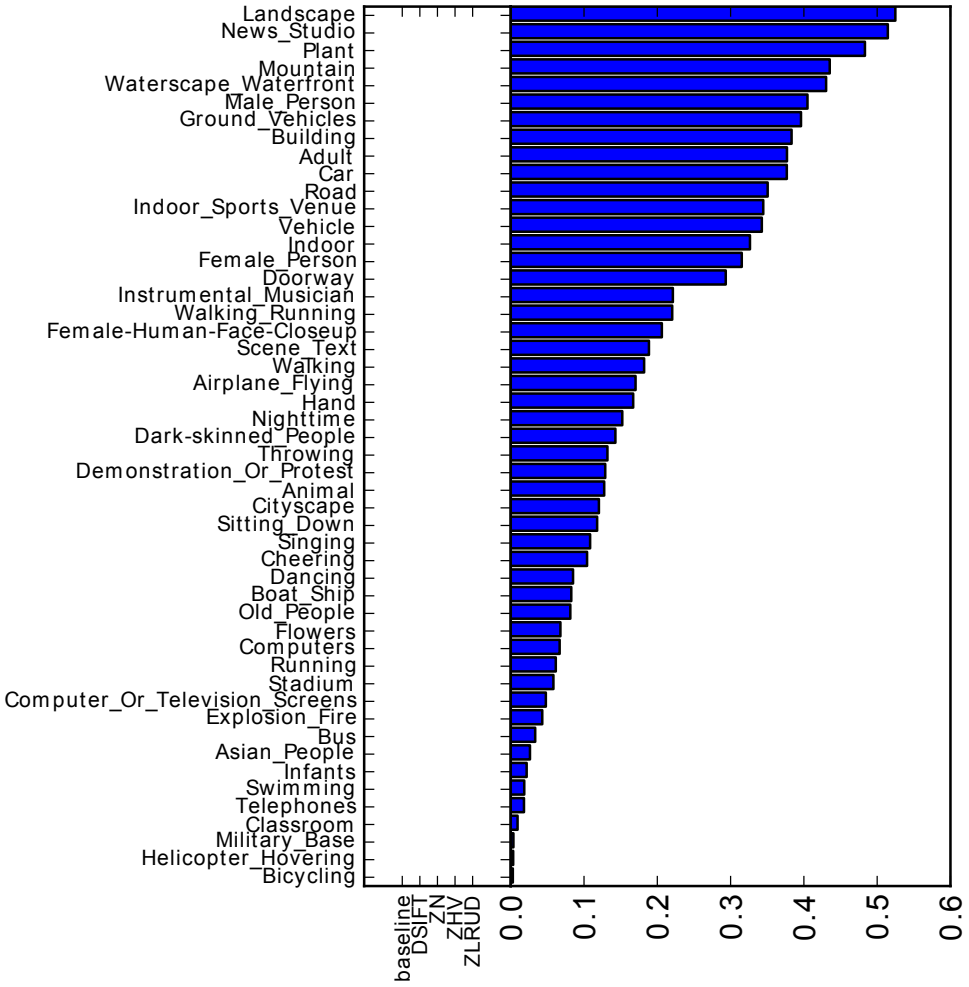


Figure 6.8: Weight contributions of each feature in the optimal best performing concept classifier and the corresponding AP

### 6.2.6 Conclusions

In this chapter we have presented a new feature that builds on the DSIFT Bag of Words classifier by incorporating local motion information. The proposed features are constructed using optic flow information at the DSIFT patch positions and the corresponding codewords. Using minimal computation, 3 spatial histogram features ZN, ZHV and ZLRUD are constructed from existing DSIFT feature codeword data and optic flow information and are mapped using the homogeneous kernel maps and classified using linear SVM. The strength of the method is that it relies on the widely available DSIFT bag of words feature data, and requires minimal feature extraction, namely optic flow at a keyframe level, and that linear classification is extremely fast, thanks to the homogeneous kernel map. Linear descriptor fusion show that the new features can improve the performance of the retrieval system by a statistically significant 5% without requiring any of the more computationally complex spatio-temporal techniques.

## 6.3 Summary on computation time

In this section we will present a timeline in figure 6.9 that shows the average time needed for one shot to be completely extracted, coded and classified by a system. In the comparison we have added the descriptors proposed in this thesis and what we consider the most suitable candidates: STIP, Dense Trajectories and MoSIFT. The numbers are based on average computation times for one shot on TRECVID 2010 (except TRECVID 2012 for Z* features). All the computation in this timeline is done on the EURECOM cluster. Classification for DT, MoSIFT and STIP has not been performed (because of time limitations) so the values are not counted in the final figures. As stated in the figure Z* features can save time by reusing DSIFT BOW data, which in a real world system is widely available. Note that the scale differs between color-coded categories. It is obvious that the most costly operation across all methods is the feature extraction, which dominates by 2 degrees of magnitude the average time cost. As a general observation all the proposed features are faster than any of the prior local feature methods presented.

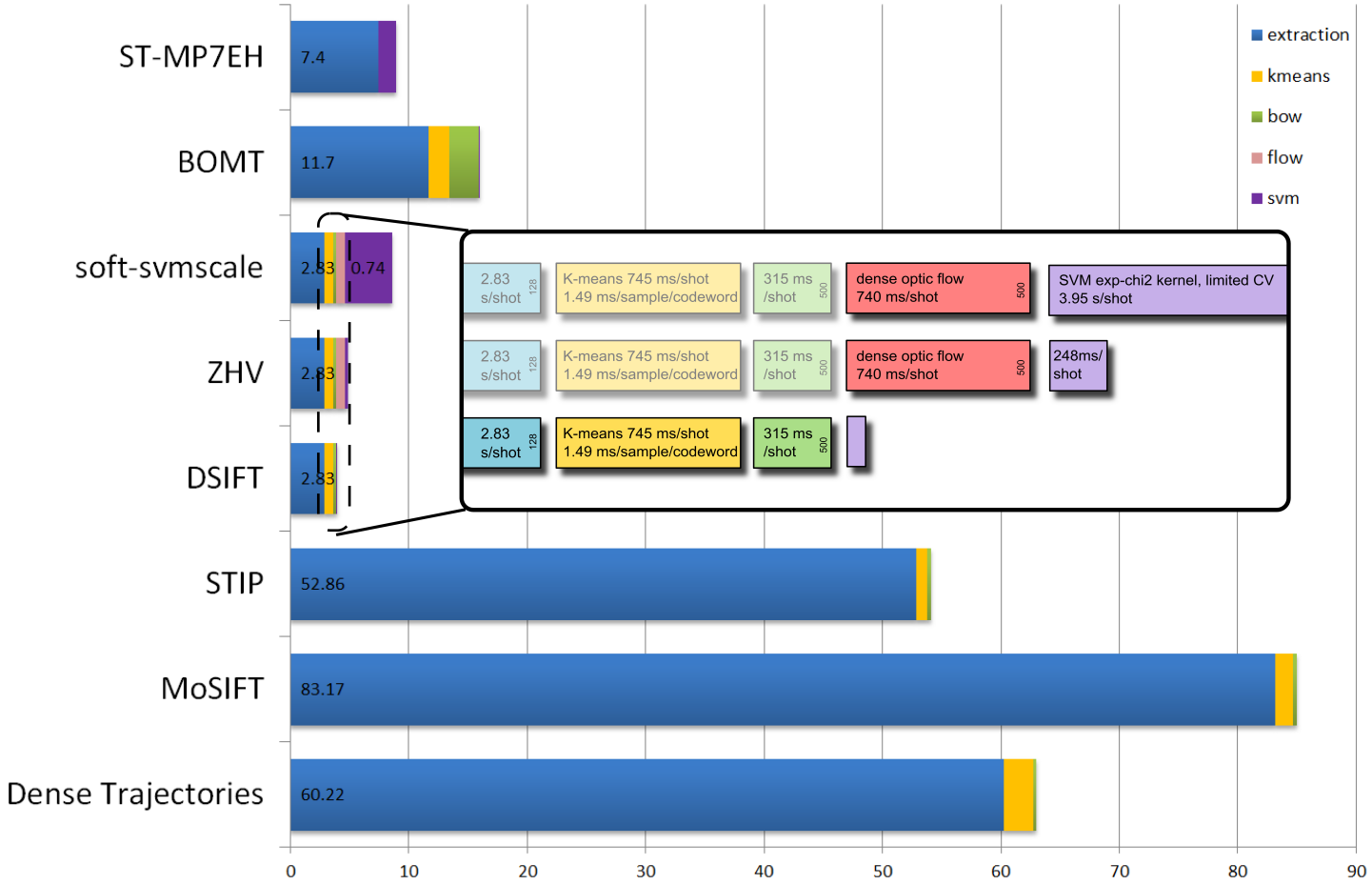


Figure 6.9: Timeline of average computation time expressed in seconds per shot for proposed descriptors, DSIFT and State-of-the-art ST features. For every feature the SVM classification time is estimated for a number of 50 concepts. Feature extraction is in blue, codebook construction in yellow, Bag of Words assignment in green, classification in purple and optic flow extraction in red. The first 3 phases of DSIFT can be reused by ZHV and soft-svmscale so that the new features can be processed in under 1 second per shot. Classification times for the last 3 features are unknown therefore not displayed.

# Chapter 7

# Conclusions

In this chapter we present general conclusions on the thesis. With respect to the points of the problem statement in section 2.3 we conclude the following:

1. Regarding the development of spatio-temporal features specifically for concept detection, we have proposed **3 main new types of spatio-temporal features** and experimented with them in a concept detection context. The direct borrowing of action recognition techniques, extensions of keyframe SIFT and simplistic correlation methods have been dismissed in chapter 4. The new features described in chapters 4.3, 5 and 6 show improving results in TRECVID via fusion and give interesting results on other datasets. Also, they share little with commonly used techniques in video retrieval.

   In reference to figure 3.2 in the State of the Art chapter, the proposed methods contain elements that cover all the levels of the Semantic Indexing pipeline. ST-MP7EH is a global ST method based on edge orientations relevant to the dynamic feature extraction, classification and fusion stages. Z* features study the ways to separate and fuse DSIFT data based on motion at Bag of Words level and employ particular learning techniques in the classification stage. The study on BOMT spans all the stages, but mostly it deals with feature extraction and comprehensive performance comparison over many datasets on the final testing step.

2. Regarding computational efficiency and optimization. Figure 6.9 illustrates in a timeline the average time taken to compute each processing stage for one shot, from extraction to classification. It can be seen that the proposed methods are at least four times **faster than state of the art**. A more subtle question is: *can accuracy be traded for extraction speed?* There are two possible views on this question. The first interpretation is: can the parameters of a dynamic feature be tuned to sacrifice some performance to gain speed?

   Empirically the temporal sampling parameter (by default 1/5 frames) in ST-MP7EH can be safely changed between 1/1 to 1/30 to influence computation speed, without significant apparent impact to accuracy. In BOMT, altering the lifetime parameter $\delta$ of the Motion History Image influences the number of detections: long lifetimes mean more detailed representations of longer actions (more history), but also more motion templates which slow down the extraction. Short lifetimes reduce the number of motion templates but decrease accuracy by failing to model long motions. In Z* features, the quality of optic flow indirectly influences the quality of the features. The speed vs accuracy issue is thus a question of optic flow extraction parameters: number of scale pyramids used, size of the search window, maximum number of iterations or other stop criteria etc. The impact of the quality of optic flow is very difficult to estimate, and highly impractical to test because the entire process must be redone from the extraction step.

   The second interpretation is: can we develop features that run faster than the state of the art with an acceptable loss in accuracy? Answering this question is not straightforward for two reasons: comparisons with the state of the art are troublesome on TRECVID (see section 7.1) and there is no consensus in what 'acceptable' accuracy loss would be in the context.

   However, we have shown that on KTH BOMT performs at 88.97% compared to STIP 92.10% but is extracted at 4.5 times the speed of STIP. The fact that a very simplistic and fast method can perform better than what has been the state-of-the-art in action recognition until 2008 [38,43,68,69] is remarkable. On TRECVID we lack the figures for experimental comparison of STIP or Dense Tracks classification accuracy, but it is reasonable to believe that BOMT slightly underperforms STIP.

3. Regarding complementarity of features and improvement by feature fu-

sion. All proposed features **improve in feature fusion on TRECVID**: for 10 concepts in TRECVID 2011, ST-MP7EH and SIFT produce a 22% improvement, BOMT improves by 5% when combined with DSIFT and MoSIFT on TRECVID 2010, Z* features improve together a TRECVID 2012 baseline system containing many features of 0.179 MAP to 0.189. The improvements are verified by the TRECVID randomization test to be statistically significant (not due to chance) for a confidence level of 0.05. In some cases e.g. BOMT, the improvement by fusion is surprising because of the low MAP of the descriptor alone: BOMT has a MAP of 0.015 which is only about 0.005 higher than the random classifier. It is perhaps worth noting that the mechanisms of the three main proposed features are very different, thus supporting the idea of complementarity.

## 7.1 Considerations on TRECVID

A few observations can be made regarding the use of spatio-temporal features in video retrieval, in particular TRECVID.

At the time of writing publicly available information on the exact contribution of STIP or MoSIFT to TRECVID system performance is insufficient. This makes impossible a comparison of our features with a hypothetical "state of the art in spatio-temporal video retrieval". The probable cause of this missing data is the computation time. Indeed, at the time of writing one of our own experiments extracting Dense Trajectories features from TRECVID 2010 videos, parallelized over 9 processing nodes after about 14 weeks of continuous computation has yet to finish. Using the values in the timeline 6.9, total extraction time of STIP for the 266,473 shots of TRECVID 2010 would take an estimated 185 days of CPU time, not counting any BOW or SVM stage following extraction.

The mosaic 7.1 shows examples of shots ranked much higher by Z* features than by DSIFT. For each concept a positive shot having high rank in one of the Z* features and low rank in DSIFT is presented. Extracted optic flow is shown with color-coded motion arrows. The colors correspond to the nearest codeword the underlying DSIFT patch is assigned to. We can see that although in some cases the optic flow detection or camera stabilization is inaccurate in most cases the object presents some characteristic movement: motion of the face in person concepts, uniform motion in scenery concepts, etc. In general, when motion is correctly detected spatio-temporal features do work better for concepts containing significant amounts of motion. One


Figure 7.1: Examples of shots retrieved by spatio-temporal Z* features from the TRECVID 2010 test on 50 concepts.

interesting example in this image is the concept 'Infants' where the classifier correctly learns that most training examples are not actual video but repeating frames made from recording a slideshow. Another interesting effect that is not unique to spatio-temporal retrieval is learning of the context instead of the object: in concept 'Boat_Ship', the classifier basically learns motion patterns of water which are incidentally present in shots containing boats.

## 7.2 Future directions in the field

All the contributions presented in this work depend in their initial phase on **computer vision methods**. ST-MP7EH uses a Sobel detector to find edges and quantize their orientations. BOMT extracts foreground motion by frame differencing and segmenting motion into motion templates. Z* features divide DSIFT visual words into motion categories based on Lukas-Kanade optic flow. Camera motion stabilization that uses optic flow and RANSAC is performed as a preprocessing step for all descriptors. Any noise coming from the low-level visual methods affects the entire chain of processing and lowers the performance of the system.

We remind that in the case of BOMT alone (see section 5.6) the use of the provided object bounding boxes on Youtube (instead of the noisy motion template extraction) increased accuracy by 37%. Better **background/foreground separation** would greatly benefit BOMT. One possible candidate is Brox' motion segmentation [70], but for the moment the cost of a few seconds per pair of frames makes the approach prohibitively slow for TRECVID. More advanced methods such as Liu's MRF method [71] could be used to segment the foreground but again at very high computational cost. Also, eliminating videos with camera shake increases accuracy of BOMT by 15%. This is a sign that our motion compensation approach may be insufficiently robust. Multiple homography approaches such as [72] have been tried but are still not scalable.

Z* features determine the movement of each DSIFT patch by using Lukas Kanade optic flow. There are too many freely available implementations of sparse **optic flow** to count (we use OpenCV [62] in our experiments), but the choice of parameters that could work consistently well on all TRECVID videos is unclear. There is considerable noise encountered in samples of Z* extraction that mutually affects the camera stabilization method. Empirical measurements are not easily obtainable since changing configuration on the extraction step requires the entire experiment to be run several times. Perhaps a more robust, unified way to extract motion from video, or a way

to automatically estimate flow parameters from the data itself could reduce such effects.

Since the focus of this work was on the features and not on the classification methods, we haven't dealt much with more recent developments in learning models. While it is true that many researchers still use SVMs with RBF kernels to quasi state-of-the-art results, a number of modeling techniques that can be used completely independently of the feature extraction have developed. On the pooling stage, **Fisher vectors** [28] combine discriminative and generative approaches by modeling a GMM based on feature distribution information. Good results on TRECVID SIN and PASCAL VOC place this method as a likely candidate for the replacement of Bag of Words, in spite of higher computation costs. Another approach uses **supervectors** [29], a similarly large dense representation that uses distances to every cluster center has also given good results on TRECVID. The state of the art in classification for retrieval is in a process of shifting from SVMs to **deep belief networks**. The work by Krizhevsky et. al. on ImageNet [73] has been only recently replicated on keyframe-based video retrieval by the University of Amsterdam in the 2013 SIN edition, giving top performance. It is interesting to note that in deep learning the design of the low-level features seems not to matter anymore (e.g. [74] uses a simple flattened $15 \times 15 \times 7$ volume patch in temporal gradient). Rather the features are self-learned in the form of convolutional filters.

In the introduction of this thesis, exponential increase of video content has been discussed. This is the combined result of three factors: growing availability of video recording devices, exponential increase with time of image sensor resolutions and available storage capacity. If we consider that these three variables conform to Moore's law $2^t$ we can approximate the growth of a web video storage collection as a product of three exponentials (roughly $2^t \cdot 2^t \cdot 2^t = 8^t$). The growth of CPU power is only $2^t$, which means that if the trends are held, **computing power will become insufficient for video analysis**. Spatio-temporal descriptors may pass from "slow and impractical" to "impossible to test". Practical alleviation solutions exist: subsampling large video collections is in the practice of TRECVID, which at the end causes unreliability in classification, annotation and evaluation. [79] Although parallel and distributed computing are developing in full force, immediate improvements in the video content analysis practice are not easily foreseeable. GPU/CUDA extraction solutions for exist mostly for particular computer vision problems such as optic flow and image segmentation, but researchers only test them on small data. Cluster computing is the only practical solution for many researchers working on video retrieval,

but long computation times and the occasional need for highly technical expertise makes this approach challenging in a research environment. If large scale retrieval experiments are to keep up with the video data, future spatio-temporal features should take computational constraints more seriously into consideration. Currently some researchers still prefer to ignore CPU time in their experiments and instead test their video descriptors on small sets such as the venerable KTH. The descriptors proposed in this work are fast enough to even allow TRECVID experiments to be run on a modern desktop computer.

# Appendices

# Appendix A

# Camera motion compensation

Analyzing video content requires some special preprocessing methods. Here is a brief description on how camera motion is compensated prior to spatio-temporal analysis.

Camera motion stabilization is used to reduce the noize induced by ego-movement of the camera and to possibly facilitate subsequent operations such as background extraction. For all the experiments in this thesis, a motion stabilization method based on optic flow correspondences and RANSAC is used, somewhat similarly to the stabilization in [117].

Motion stabilization works by aligning two consecutive frames. Using code from the OpenCV library [62], at $8 \times 8$ equally spaced points $a_i$ in the first image grid-based Lukas-Kanade optical flow $a_i \rightarrow b_i$ is extracted. Extraction for some of the 64 points can potentially fail, so a double-check filtering criteria is used: the flow destination $b_i$ (i.e. the estimated coordinate of the displaced point on the second image) serves as location for a reversed optic flow estimation $b_i \rightarrow a'_i$. If the distance between the origin and the reverse flow destination is greater than a threshold value $\|a_i - a'_i\| > \epsilon$, the point is discarded. $\epsilon$ is empirically chosen at 1 pixel. The remaining $(a_i, b_i)$ pairs are considered *correspondences* beetween the two images and are used in the next step.

Using the correspondences, a RANSAC [131] function is used to estimate the parameters of a perspective transform that best fits the transformation from the $a_i$ points to the $b_i$ points. RANSAC randomly initializes a consen-

sus set from the data and iteratively reestimates the model based on how many points appear as model inliers. On convergence, RANSAC most likely produces the optimal model (i.e. homography) that fits the real data inliers (i.e. the point correspondences that match static points in the scene) and ignores outliers (point correspondences caused by moving objects).

Using the resulting transformation, we can easily apply the geometric transformation to the pixels of the first image and make pixelwise comparisons with the second image. The two images will be aligned such that in most situations no camera motion is perceived. Examples of stabilization can be seen in figure A.1. Note that for demonstration purposes, the number of correspondences is much higher than the 64 used in practice.

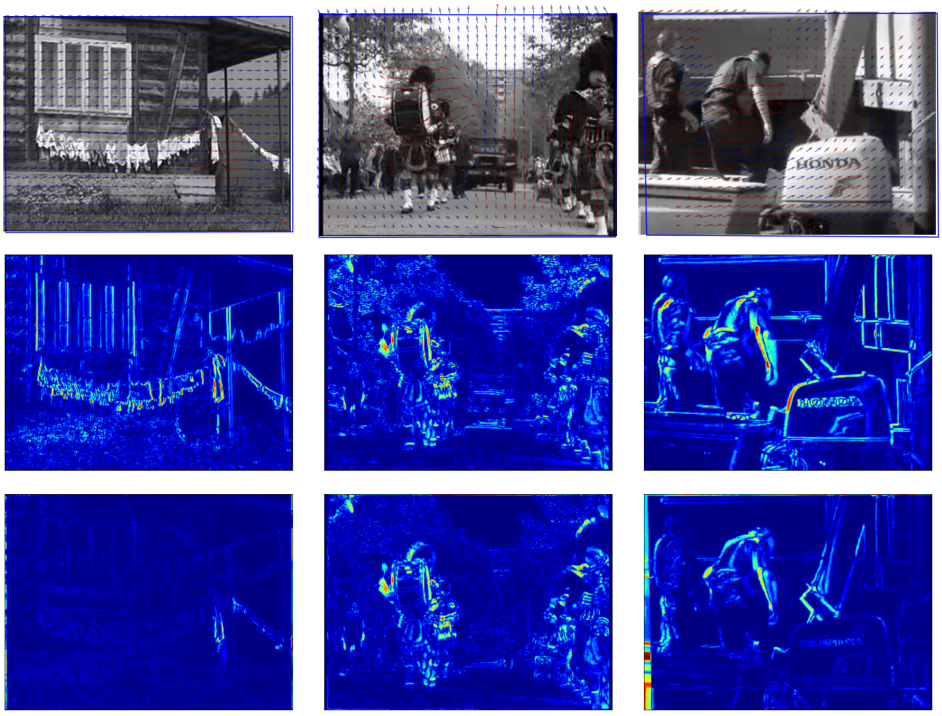


Figure A.1: Camera motion stabilization by homography estimation. First row: dense optic flow is used to create correspondences. RANSAC models the homography by rejecting the red outliers and including the blue ones. The blue outline signifies the shape of the transformed frame according to the estimated homogrpahy. Using the transformation, change detection can be estimated by differencing the transformed current frame with the unchanged next frame. Middle row: change detection without stabilization by homography transformation. Bottom row: change detection with stabilization. Note how the outlier correspondences match the foreground movement determined by differencing

# Appendix B

# Late weighted linear fusion

The most used type of feature fusion in information retrieval is the weighted linear fusion. In our experiments weighted linear fusion is performed in the following way: first $i$ individual binary classifiers are constructed for each feature. The classifiers are then applied on the test data, thus obtaining score/confidence values $score_{classifier_i}$ for every sample of the test data in every feature. A new set of scores is created by applying a predetermined linear combination of the feature scores, using constant parameters $\alpha_i$:

$$score_{fusion}(sample) = \sum_{i \in classifiers} \alpha_i * score_{classifier_i}(sample) \quad \text{with} \sum_i \alpha_i = 1 \tag{B.1}$$

In practice $\alpha_i$ parameters are searched for using brute force: first the interval $[0, 1]$ is discretized into a small number $n$ of steps. Then all the combinations $(\alpha_1, \alpha_2, \ldots, \alpha_i)$ of size $i$ of these steps are generated and the ones that do not sum up to one are discarded.[1] The number of valid combinations is thus much smaller than $n^i$.

There are two ways of performing this parameter search, which separate the two kinds of late fusion: *Optimistic fusion* evaluates all the combina-

[1] Another possibility is to generate combinations of $i-1$ and to calculate the last one as $\alpha_i = 1 - \sum_{j=1}^{i-1} \alpha_j$, and then filtering out values that do not fit in $[0, 1]$

tions of parameters directly on the test set and retrieve the best performing combination as the fusion result. It does however have the disadvantage of being biased because of using test data in training. Resulting classifier is overfit on the test data. The performance of the optimist classifier represents an upper bound on the improvement possible through linear fusion. It can be used in practice when there is to little or no validation data available, for instance when only confidence values on the test set are provided for an external classifier.

*Pessimist fusion* addresses this issue by taking a cross-validation approach: the training data is subdivided into two parts, classifiers are trained on one half, and the parameter search is done on the second half like in optimist fusion. The difference is that the weight parameters are transfered to the actual classifier (built on the full training data) and test dataset directly. Since the parameter search uses training data only, this method has no training bias. The MAP of this classifier gives the actual improvement due to fusion, as opposed to optimist fusion that gives an upper bound.

The standard way to perform linear fusion is the pessimist type. Throughout this thesis, if the type of fusion is unspecified, it should be considered pessimist. In table B.1 it can be noted that the performance ratio between pessimist and optimist fusion is around 90% if we exclude the outlier third column.

| | Run2+ DSIFT+ ZN+ ZHV+ ZLRUD | DSIFT+ ZN+ ZHV+ ZLRUD | Z+L+ R+U+D | Z+H+V | Z+N | image_HOG+ mt_HOG |
|---|---|---|---|---|---|---|
| dataset | TRECVID 2012 | TRECVID 2010 | TRECVID 2010 | TRECVID 2010 | TRECVID 2010 | Youtube |
| metric | MAP | MAP | MAP | MAP | MAP | avg. acc. |
| mean of components | 0.1084 | 0.0829 | | | | 0.51 |
| optimist fusion | 0.1966 | 0.1000 | 0.0860 | 0.0809 | 0.0899 | |
| pessimist fusion | 0.1898 | 0.0901 | 0.0498 | 0.0786 | 0.0663 | 0.62 |
| early fusion | | 0.0424 | 0.0756 | 0.0809 | 0.0853 | 0.61 |

Table B.1: Comparison of different fusion strategies on TRECVID 2010, 2012 and Youtube.

# Bibliography


[1] G. E. Moore *et al.*, "Cramming more components onto integrated circuits," 1965.

[2] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Comput. Vis. Image Underst.*, vol. 115, pp. 224–241, February 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314210002171

[3] P. Over, G. Awad, J. Fiscus, B. Antonishek, and G. Quenot, "TRECVID 2010 - an overview of the goals, tasks, data, evaluation mechanisms and metrics," pp. 1–34, 2010.

[4] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A. Smeaton, W. Kraaij, G. Quénot *et al.*, "An overview of the goals, tasks, data, evaluation mechanisms and metrics," in *TRECVID 2011-TREC Video Retrieval Evaluation Online*, 2011.

[5] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. Smeaton, and G. Quéenot, "Trecvid 2012–an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID*, 2012.

[6] G. Bradski and J. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, no. 3, pp. 174–184, 2002.

[7] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on.* IEEE, 1997, pp. 762–768.

[8] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological cybernetics*, vol. 61, no. 2, pp. 103–113, 1989.

[9] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1. IEEE, 1994, pp. 582–585.

[10] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic *et al.*, "Query by image and video content: The qbic system," *Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[11] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern recognition and artificial intelligence*, vol. 116, pp. 374–388, 1976.

[12] N. Ballas, B. Labbé, A. Shabou, H. Le Borgne, P. Gosselin, M. Redi, B. Merialdo, H. Jégou, J. Delhumeau, R. Vieux *et al.*, "Irim at trecvid 2012: Semantic indexing and instance search," in *Proceedings of the workshop on TREC Video Retrieval Evaluation (TRECVID)*, 2012.

[13] U. Niaz, M. Redi, C. Tanase, and B. Merialdo, "Eurecom at trecvid 2013: The semantic indexing task," 2013.

[14] C. Zhu, C.-E. Bichot, and L. Chen, "Color orthogonal local binary patterns combination for image region description," Technical Report, LIRIS UMR5205 CNRS, Ecole Centrale de Lyon, Tech. Rep., 2011.

[15] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[16] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.

[17] D. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[18] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[19] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 524–531.

[20] H. Wang, M. Ullah, A. Klaser, I. Laptev, C. Schmid *et al.*, "Evaluation of local spatio-temporal features for action recognition," in *BMVC 2009-British Machine Vision Conference*, 2009.

[21] J. Van De Weijer and C. Schmid, "Coloring local feature extraction," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 334–348.

[22] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[24] B. S. Manjunath, J. rainer Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 703–715, 1998.

[25] D. K. Park, Y. S. Jeon, and C. S. Won, "Efficient use of local edge histogram descriptor," in *Proceedings of the 2000 ACM workshops on Multimedia*, ser. MULTIMEDIA '00. New York, NY, USA: ACM, 2000, pp. 51–54.

[26] N. Inoue, Y. Kamishima, T. Wada, K. Shinoda, and S. Sato, "Tokyotech+ canon at trecvid 2011."

[27] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[28] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[29] X. Zhou, K. Yu, T. Zhang, and T. Huang, "Image classification using super-vector coding of local image descriptors," *Computer Vision–ECCV 2010*, pp. 141–154, 2010.

[30] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *British Machine Vision Conference*, 2011.

[31] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[32] Y. Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis-using both audio and visual clues," *Signal Processing Magazine, IEEE*, vol. 17, no. 6, pp. 12–36, 2000.

[33] C. G. Snoek, M. Worring, and A. W. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 2005, pp. 399–402.

[34] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976 – 990, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885609002704

[35] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 3, pp. 257–267, 2001.

[36] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, oct. 2003, pp. 726 –733 vol.2.

[37] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1395–1402.

[38] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 166–173.

[39] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 984–989.

[40] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 405–412.

[41] A. Ogale, A. Karapurkar, G. Guerra-Filho, and Y. Aloimonos, "View-invariant identification of pose sequences for action recognition," in *Video Analysis and Content Extraction Workshop (VACE)*. Citeseer, 2004.

[42] I. Laptev and T. Lindeberg, "Space-time interest points," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, oct. 2003, pp. 432 –439 vol.1.

[43] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*. IEEE, 2005, pp. 65–72.

[44] A. Kläser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *In BMVC 08*.

[45] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*, ser. MULTIMEDIA '07. New York, NY, USA: ACM, 2007, pp. 357–360.

[46] M. Chen and A. Hauptmann, "Mosift: Recognizing human actions in surveillance videos," Carnegie Mellon University, Tech. Rep. CMU-CS-09-161, 2009.

[47] W. Ren, S. Singh, M. Singh, and Y. Zhu, "State-of-the-art on spatio-temporal information-based video retrieval," *Pattern Recognition*, vol. 42, no. 2, pp. 267 – 282, 2009.

[48] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176.

[49] A. Noguchi and K. Yanai, "A surf-based spatio-temporal feature for feature-fusion-based action recognition," in *Trends and Topics in Computer Vision*. Springer, 2012, pp. 153–167.

[50] Q. Zhu, D. Liu, T. Meng, C. Chen, M. Shyu, Y. Yang, H. Ha, F. Fleites, and S.-C. Chen, "Florida international university and university of miami trecvid 2012," in *TRECVID 2012 Workshop, Gaithersburg, MD, USA*, 2012.

[51] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1932–1939.

[52] T. Alonso, M. Barrena, P. G. Rodríguez, A. Polo, M. Salas, A. Caro, M. Durán, F. Rodríguez, L. Arévalo, A. Corral *et al.*, "Gim at trecvid 2012 the light semantic indexing task," 2012.

[53] L. Cao, S.-F. Chang, N. Codella, C. Cotton, D. Ellis, L. Gong, M. Hill, G. Hua, J. Kender, M. Merler *et al.*, "Ibm research and columbia university trecvid-2012 multimedia event detection (med), multimedia event recounting (mer), and semantic indexing (sin) systems," in *Proc. TRECVID 2012 workshop. Gaithersburg, MD, USA*, 2012.

[54] M. Rautiainen, M. Varanka, I. Hanski, M. Hosio, A. Pramila, J. Liu, and T. Ojala, "TRECVID 2005 Experiments at MediaTeam Oulu," 2005.

[55] F. Mahmoudi, J. Shanbehzadeh, A.-M. Eftekhari-Moghadam, and H. Soltanian-Zadeh, "Image retrieval based on shape similarity by edge orientation autocorrelogram," *Pattern Recognition*, vol. 36, no. 8, pp. 1725 – 1736, 2003.

[56] M. Yang, S. Ji, W. Xu, J. Wang, F. Lv, K. Yu, Y. Gong, M. Dikmen, D. J. Lin, and T. S. Huang, "Detecting Human Actions in Surveillance Videos."

[57] W. Zhao and C. Ngo, "Lip-vireo: Local interest point extraction toolkit," *Software available at http://www. cs. cityu. edu. hk/ wzhao2/lipvireo. htm*, 2008.

[58] C. G. M. Snoek and K. E. A. van de Sande, "The MediaMill TRECVID 2010 semantic video search engine," in *Proceedings of the TRECVID Workshop*, 2010.

[59] C. Tănase and B. Merialdo, "Efficient spatio-temporal edge descriptor," in *Advances in Multimedia Modeling*. Springer, 2012, pp. 210–221.

[60] P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, no. 7, pp. 1030–1044, 1999.

[61] M. Ahad, J. Tan, H. Kim, and S. Ishikawa, "Motion history image: its variants and applications," *Machine Vision and Applications*, pp. 1–27, 2012.

[62] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[63] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[64] Q. Hu, L. Qin, Q. Huang, S. Jiang, and Q. Tian, "Action recognition using spatial-temporal context," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, aug. 2010, pp. 1521 –1524.

[65] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. ACM, 2010, pp. 3539–3546.

[66] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[67] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[68] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3. IEEE, 2004, pp. 32–36.

[69] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[70] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," *Computer Vision–ECCV 2010*, pp. 282–295, 2010.

[71] F. Liu and M. Gleicher, "Learning color and locality cues for moving object detection and segmentation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 320–327.

[72] M. Kirchhof, "Linear constraints in two-view multiple homography estimation of uncalibrated scenes," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. ISPRS*, vol. 2008, pp. 13–20, 2008.

[73] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1106–1114.

[74] A. Castrodad and G. Sapiro, "Sparse modeling of human actions from motion imagery," *International journal of computer vision*, pp. 1–15, 2012.

[75] S. Ayache and G. QuÃ©not, "Video Corpus Annotation using Active Learning," in *European Conference on Information Retrieval (ECIR)*, Glasgow, Scotland, mar 2008, pp. 187–198.

[76] E. Yilmaz, E. Kanoulas, and J. 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.* ACM, 2008, pp. 603–610.

[77] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[78] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high-level features using large scale unsupervised learning," *arXiv preprint arXiv:1112.6209*, 2011.

[79] J. Yang and A. G. Hauptmann, "(Un)Reliability of video concept detection," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*, ser. CIVR '08. New York, NY, USA: ACM, 2008, pp. 85–94.

[80] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 6, pp. 797–819, 2011.

[81] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 248–255.

[82] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[83] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[84] R. Aly, K. McGuinness, S. Chen, N. O'Conner, K. Chatfield, O. Parkhi, R. Arandjelovic, A. Zisserman, B. Fernando, T. Tuytelaars *et al.*, "Axes at trecvid 2012," in *Proceedings of the TRECVid Workshop*, 2012.

[85] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[86] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[87] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. d. Bona, A. Binder, C. Gehl, and V. Franc, "The shogun machine learning toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 1799–1802, 2010.

[88] V. Franc and S. Sonnenburg, "Optimized cutting plane algorithm for support vector machines," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 320–327.

[89] W. Adams, G. Iyengar, C.-Y. Lin, M. R. Naphade, C. Neti, H. J. Nock, and J. R. Smith, "Semantic indexing of multimedia content using visual, audio, and text cues," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 2, pp. 170–185, 2003.

[90] S. T. Strat, A. Benoit, H. Bredin, G. Quénot, and P. Lambert, "Hierarchical late fusion for concept detection in videos," in *Computer Vision–ECCV 2012. Workshops and Demonstrations*. Springer, 2012, pp. 335–344.

[91] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 6.

[92] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[93] N. Shuicheng YA, G. Zhongyang HUAN, N. Qiang CHE, G. Zheng SON, U. Si LI, N. Xiangyu CHE, N. Xiaotong YUA, A. Tat-Seng CHU, A. Yang HU, and N. Shengmei SHE, "Boosting classification with exclusive context," 2010. [Online]. Available: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/workshop/nuspsl.pdf

[94] S. T. Strat, A. Benoit, P. Lambert, and A. Caplier, "Retina enhanced surf descriptors for spatio-temporal concept detection," *Multimedia Tools and Applications*, pp. 1–27.

[95] R. Negrel, D. Picard, and P.-H. Gosselin, "Compact tensor based image representation for similarity search," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 2425–2428.

[96] D. Weinland and E. Boyer, "Action recognition using exemplar-based embedding," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–7.

[97] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion context: A new representation for human action recognition," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 817–829.

[98] I. Laptev and P. Pérez, "Retrieving actions in movies," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.* IEEE, 2007, pp. 1–8.

[99] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[100] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Computer Vision–ECCV 2008.* Springer, 2008, pp. 650–663.

[101] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," INRIA, Research Report RR-8050, Aug. 2012. [Online]. Available: http://hal.inria.fr/hal-00725627

[102] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Computer Vision–ECCV 2006.* Springer, 2006, pp. 428–441.

[103] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 915–928, 2007.

[104] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 1996–2003.

[105] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, june 2008, pp. 1 –8.

[106] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, june 2009, pp. 2929 –2936.

[107] K. Shirahama and K. Uehara, "Kobe university and muroran institute of technology at trecvid 2012 semantic indexing task," in *Proceedings of TREC Video Retrieval Evaluation Workshop (TRECVID 2012)*, 2012.

[108] M. Al Ghamdi, L. Zhang, and Y. Gotoh, "Spatio-temporal sift and its application to human action classification," in *Computer Vision–ECCV 2012. Workshops and Demonstrations.* Springer, 2012, pp. 301–310.

[109] A. Moumtzidou, A. Dimou, N. Gkalelis, S. Vrochidis, V. Mezaris, and I. Kompatsiaris, "ITI-CERTH participation to TRECVID 2010," 2010.

[110] Y. Shimoda, A. Noguchi, and K. Yanai, "UEC at TRECVID 2010 semantic indexing task."

[111] M. Naito, K. Hoashi, K. Matsumoto, M. Shishibori, K. Kita, A. Kutics, A. Nakagawa, F. Sugaya, and Y. Nakajima, "High-level feature extraction experiments for TRECVID 2007," in *TRECVID'07*, 2007.

[112] S. Tang, Y. dong Zhang, J. tao Li, X. feng Pan, T. Xia, M. Li, A. Liu, L. Bao, S. chang Liu, Q. feng Yan, and L. Tan, "Rushes Exploitation 2006 By CAS MCG."

[113] R. Polana and R. Nelson, "Recognition of motion from temporal texture," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, jun 1992, pp. 129 –134.

[114] T. Darrell and A. Pentland, "Space-time gestures," in *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, jun 1993, pp. 335 –340.

[115] F. Liu and R. Picard, "Finding periodicity in space and time," in *Computer Vision, 1998. Sixth International Conference on*, jan 1998, pp. 376 –383.

[116] S. Sadanand and J. Corso, "Action bank: A high-level representation of activity in video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 1234–1241.

[117] N. Ikizler-Cinbis and S. Sclaroff, "Object, scene and actions: Combining multiple features for human action recognition," *Computer Vision–ECCV 2010*, pp. 494–507, 2010.

[118] H. Meng, N. Pears, and C. Bailey, "Human action classification using svm_2k classifier on motion features," in *Multimedia Content Representation, Classification and Security.* Springer, 2006, pp. 458–465.

[119] S. Wang, Z. Ma, Y. Yang, X. Li, C. Pang, and A. Hauptmann, "Semi-supervised multiple feature analysis for action recognition," 2013.

[120] C. Tanase and B. Mérialdo, "Introducing motion information in dense feature classifiers," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2013 14th International Workshop on*. IEEE, 2013, pp. 1–4.

[121] D. Gorisse and F. Precioso, "IRIM at TRECVID 2010: Semantic Indexing and Instance Search," in *TREC online proceedings*, Gaithersburg, United States, Nov. 2010, pp. –, gDR ISIS.

[122] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 978–994, 2011.

[123] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3099–3104.

[124] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," *Image Analysis*, pp. 363–370, 2003.

[125] C. Tănase and B. Mérialdo, "Semantic concept detection using dense codeword motion," in *Advanced Concepts for Intelligent Vision Systems*. Springer, 2013, pp. 705–713.

[126] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. [Online]. Available: http://hal.inria.fr/inria-00633013

[127] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

[128] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 604–610.

[129] F. Wang, Y.-G. Jiang, and C.-W. Ngo, "Video event detection using motion relativity and visual relatedness," in *Proceedings of the 16th ACM international conference on Multimedia*. ACM, 2008, pp. 239–248.

[130] Z. Liang, N. Inoue, and K. Shinoda, "Event detection by velocity pyramid," in *MultiMedia Modeling*. Springer, 2014, pp. 353–364.

[131] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.