Supervised Statistical Representations for Human Action Recognition in Video

Muhammad Muneeb Ullah

N° d'ordre : ANNÉE 2012

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : (Informatique)

Ecole doctorale (Matisse)

présentée par

Muhammad Muneeb Ullah

préparée à l'unité de recherche (INRIA)
(Institut National de Recherche en Informatique et en Automatique)

Représentations Statistiques Supervisées pour la Reconnaissance d'Actions Humaines dans les Vidéos

. . .

Supervised Statistical Representations for Human Action Recognition in Video

Thèse soutenue à (INRIA, Paris)
le (23-10-2012)

devant le jury composé de :

Patrick BOUTHEMY
DR, INRIA, France (Président)

Matthieu CORD
Prof., UPMC-Sorbonne, France (Rapporteur)

Tinne TUYTELAARS
Prof., Univ. Leuven, Belgiam (Rapporteur)

Frederic PRECIOSO
Prof., Univ. Sophia-Antipolis, France (Examinateur)

Ewa KIJAK
A. Prof., Univ. Rennes 1, France (Examinateur)

Patrick PÉREZ
Scientist, Technicolor, France (Directeur de thèse)

Ivan LAPTEV
CR, INRIA, France (Co-directeur de thèse)

Abstract

This thesis addresses the problem of human action recognition in realistic video data, such as movies and online videos. Automatic and accurate recognition of human actions in video is a fascinating capability. The potential applications range from surveillance and robotics to medical diagnosis, content-based video retrieval, and intelligent human-computer interfaces. The task is highly challenging due to the large variations in person appearances, dynamic backgrounds, view-point changes, lighting conditions, action styles and other factors.

Statistical video representations based on local space-time features have been recently shown successful for action recognition in realistic scenarios. Their success can be attributed to the mild assumptions about the data and robustness to several variations in the video. Such representations, however, often encode videos by disordered collection of low-level primitives. This thesis extends current methods by developing more discriminative features and integrating additional supervision into Bag-of-Features based video representations, aiming to improve action recognition in unconstrained and challenging video data. We start by evaluating a range of available local space-time feature detectors and descriptors under the standard Bag-of-Features framework. We then propose to improve the basic Bag-of-Features model by integrating additional supervision in the form of non-local region-level information. We further investigate an attribute-based representation, wherein the attributes range from objects (e.g., car, chair, table, etc.) to human poses and actions. We demonstrate that such representation captures high-level information in video, and provides complementary information to the low-level features. We finally propose a novel local representation for human action recognition in video, denoted as *Actlets*. Actlets are body part detectors undergoing characteristic motion patterns. We train Actlets using a large synthetic video dataset of rendered avatars and demonstrate the advantages of Actlets for action recognition in realistic data. All methods proposed and developed in this thesis represent alternative ways of constructing supervised video representations and demonstrate improvements of human action recognition in realistic settings.

Résumé en Français

Dans cette thèse, nous nous occupons du problème de la reconnaissance d'actions humaines dans des données vidéo réalistes, telles que des films et des vidéos en ligne. La reconnaissance automatique et exacte d'actions humaines dans les vidéos est une capacité fascinante. Les applications potentielles vont de la surveillance et de la robotique au diagnostic médical, à la recherche d'images par leur contenu et aux interfaces homme-machine intelligentes. Cette tâche représente un grand défi en raison des variations importantes dans les apparences des personnes, les arrière plans dynamiques, les changements d'angle de prise de vue, les conditions de luminosité, les styles d'actions et bien d'autres facteurs encore.

Les représentations statistiques de vidéos basées sur les caractéristiques spatio-temporelles locales se sont dernièrement montrées très efficaces pour la reconnaissance dans des scénarios réalistes. Leur succès peut être attribué aux hypothèses favorables sur la nature des données et à la robustesse vis à vis de plusieurs types de variations dans les vidéos. De telles représentations encodent néanmoins souvent les vidéos par un ensemble désordonné de primitives de bas niveau. Cette thèse élargit les méthodes actuelles en développant des caractéristiques ("features") plus distinctives et en intégrant une supervision additionnelle sur les représentations de vidéos basées sur les sacs de caractéristiques ("bags-of-features"), afin d'améliorer la reconnaissance d'actions dans des données vidéos aux caractéristiques non contraintes et particulièrement difficiles.

Dans la présente thèse, nous évaluons tout d'abord un éventail de détecteurs et de descripteurs de caractéristiques spatio-temporelles dans le cadre du modèle standard des sacs de caractéristiques. Nous proposons ensuite d'améliorer le modèle de base des sacs de caractéristiques en intégrant une supervision additionnelle sous la forme d'informations non locales au niveau des régions. Nous examinons ensuite une représentation basée sur attributs, où les attributs sont par exemple des objets (par exemple, voiture, chaise,

table, etc.), des postures ou encore des actions humaines. Nous montrons que de telles représentations capturent des informations de haut niveau sur les séquences vidéos et fournissent des informations complémentaires aux caractéristiques de bas niveau. Enfin, nous proposons une nouvelle représentation locale pour la reconnaissance d'actions humaines en vidéo, dénotée Actlets. Les Actlets sont des détecteurs de parties du corps soumis à des modèles de mouvement caractéristiques. Pour entraîner les Actlets, nous créons un jeu de données sur les mouvements humains relativement important en exploitant des vidéos générées automatiquement en animant des personnages synthétiques à l'aide de données de capture de mouvement. L'évaluation empirique démontre l'efficacité de la représentation basée sur les Actlets dans des ensembles de données vidéo très difficiles.

Cette thèse démontre que la supervision aide à apprendre efficacement quelles sont les caractéristiques distinctives, ce qui améliore les résultats sur les données vidéos réalistes des techniques de reconnaissance basées sur le modèle des sacs de caractéristiques.

1. Évaluation des caractéristiques spatio-temporelles

On trouve dans la littérature différentes méthodes de détection et de description, et des résultats de reconnaissance prometteurs sont présentés pour différents ensembles de données d'actions. Néanmoins, la comparaison de ces méthodes est limitée, en raison des différences entre les environnements expérimentaux et les méthodes de reconnaissance utilisées. Cette partie de la thèse vise à définir en premier lieu un contexte d'évaluation commun afin de comparer les détecteurs et les descripteurs spatio-temporels locaux. Toutes les expériences sont effectuées dans le cadre du même modèle de reconnaissance basé sur les sacs de caractéristiques. Dans un second temps, nous effectuons une évaluation systématique des différentes caractéristiques spatio-temporelles. Nous évaluons l'efficacité de plusieurs détecteurs et descripteurs des points spatio-temporels intéressants en même temps que leurs combinaisons sur des jeu de données dont le degré de difficulté varie. Nous introduisons et évaluons également les caractéristiques denses, obtenues par un échantillonnage régulier des patches spatio-temporels locaux.

Détecteurs de caractéristiques. Dans notre évaluation expérimentale, nous considérons les détecteurs de caractéristiques suivants.

(i) Le détecteur *Harris3D* [88] qui étend aux séquences d'images le détecteur de Harris [62] pour les images. En chaque point vidéo, la matrice spatio-temporelle du moment d'ordre 2 μ est calculée en utilisant une fonction gaussienne lissante séparable et les gradients spatio-temporels. Les points d'intérêt spatio-temporels sont localisés aux maxima locaux de $H = \det(\mu) - k\,\mathrm{trace}^3(\mu)$.

(ii) Le détecteur *Cuboid* [31] sur les filtres temporels de Gabor. La fonction de réponse a la forme : $R = (I * g * h_{ev})^2 + (I * g * h_{od})^2$, où $g(x, y; \sigma)$ et le noyau gaussien lissant 2D et h_{ev} et h_{od} sont des filtres de Gabor 1D. Les points d'intérêt spatio-temporels sont détectés aux maxima locaux de R.

(iii) Le détecteur *Hessian3D* [184] est une extension spatio-temporelle de la mesure hessienne de saillance [9, 100]. Le déterminant de la matrice hessienne 3D est utilisé pour mesurer la saillance. Le déterminant est calculé à plusieurs échelles spatiales et temporelles. Un algorithme de suppression non-maximale sélectionne les extrêma comme points d'intérêt.

(iv) *Dense sampling* extrait des blocs vidéo multi-échelles régulièrement échantillonés dans l'espace et le temps pour des échelles variables. Dans nos expériences, nous échantillons des cuboïdes qui se chevauchent spatialement et temporellement à 50%.

Descripteurs de caractéristiques. Nous examinons les descripteurs de caractéristiques suivants.

(i) Pour le descripteur *Cuboid* [31], les gradients calculés pour chaque pixel dans une région cuboïde sont concaténés en un seul vecteur. On utilise ensuite l'analyse en composante principale (ACP) pour projetter les vecteurs sur un espace de dimension plus faible.

(ii) Les descripteurs *HOG/HOF* [91] divisent une région cuboïde en une grille de cellules. Pour chaque cellule, on calcule des histogrammes à 4 classes des orientations du gradient (*HOG*) et des histogrammes à 5 classes sur le flot optique (*HOF*). Les histogrammes normalisés sont concaténés pour former les descripteurs HOG, HOF et HOGHOF.

(iii) Le descripteur *HOG3D* [77] est basé sur les histogrammes des orientations du gradient 3D. Les gradients sont calculés via une représentation vidéo intégrale. Des polyèdres réguliers sont utilisés pour quantifier de façon uniforme l'orientation des gradients spatio-temporels. Un volume 3D donné est divisé en une grille de cellules. Le descripteur correspondant concatène les histogrammes de toutes les cellules.

(iv) Le descripteur *extended SURF (ESURF)* [184] étend le descripteur d'image SURF [8] aux vidéos. À nouveau, les cuboïdes 3D sont divisées en une grille de cellules. Chaque cellule est représentée par une somme pondérée de réponses d'ondelettes de Haar, uniformément échantillonnées, alignées sur les trois axes.

Contexte d'évaluation. Nous représentons les séquences vidéo par des sacs de caractéristiques spatio-temporelles locales [157]. Les caractéristiques spatio-temporelles sont tout d'abord quantifiées en des mots visuels, et les vidéos sont représentées en

[%]	HOG3D	HOGHOF	HOG	HOF	Cuboid	ESURF
Harris3D	89.0	91.8	80.9	**92.1**	-	-
Cuboid	90.0	88.7	82.3	88.2	89.1	-
Hessian3D	84.6	88.7	77.7	88.6	-	81.4
Dense	85.3	86.1	79.0	88.0	-	-

Table 1: Précision moyenne sur le jeu de données KTH-Actions.

[%]	HOG3D	HOGHOF	HOG	HOF	Cuboid	ESURF
Harris3D	79.7	78.1	71.4	75.4	-	-
Cuboid	82.9	77.7	72.7	76.7	76.6	-
Hessian3D	79.0	79.3	66.0	75.3	-	77.3
Dense	**85.6**	81.6	77.4	82.6	-	-

Table 2: Précision moyenne sur le jeu de données UCF-Sports.

conséquence comme les histogrammes normalisés L1 sur les mots visuels. Une machine à vecteurs de support (SVM) non-linéaire [23] avec le noyau χ^2 [91] est employée pour classer les échantillons vidéo.

Les jeux de données utilisés dans cette évaluation sont KTH-Actions [157], UCF-Sports [148] et Hollywood-2 [111]. Les résultats de la classification pour ces ensembles de données et différentes combinaisons de détecteurs et descripteurs sont présentés dans les Tableaux 1-3. Les trois meilleures combinaisons de détecteur et de descripteur de caractéristiques sont soulignées.

Parmi les principales conclusions, nous remarquons que l'échantillonnage dense introduit surpasse systématiquement tous les autres détecteurs de points d'intérêt lors des tests sur des vidéos réalistes, c'est à dire sur les ensembles de données UCF-Sports et Hollywood-2. Les résultats relativement mauvais des caractéristiques denses sur les ensembles de données non réalistes KTH-Actions peuvent être expliqués par de larges portions d'arrière plan homogène dans ces ensembles de données. Ces résultats soulignent à la fois (i) l'importance d'utiliser des données vidéos expérimentales réalistes ainsi que (ii) les limites des détecteurs de points d'intérêt actuels. D'un autre côté, un échantillonnage dense produit également un grand nombre de caractéristiques, typiquement 15 à 20 fois plus que les détecteurs de caractéristiques. Cela peut avoir des implications pratiques puisqu'il est plus difficile de manier une grande quantité de caractéristiques denses qu'un nombre relativement réduit de points d'intérêt. De plus, nous remarquons une performance des détecteurs de points d'intérêt plutôt similaire sur chaque ensemble de données. En comparant les ensembles de données, Harris3D obtient des meilleurs

[mAP]	HOG3D	HOGHOF	HOG	HOF	Cuboid	ESURF
Harris3D	43.7	45.2	32.8	43.3	-	-
Cuboid	45.7	46.2	39.4	42.9	45.0	-
Hessian3D	41.3	46.0	36.2	43.0	-	38.2
Dense	45.3	**47.4**	39.4	45.5	-	-

Table 3: Moyenne des précisions moyenne (mAP) sur le jeu de données Hollywood-2.

résultats sur l'ensemble de données KTH-Actions, tandis que le détecteur cuboïde obtient de meilleurs résultats sur les ensembles de données UCF-Sports et Hollywood-2. Parmi les descripteurs testés, la combinaison de descripteurs basés sur le gradient et sur le flot optique apparaît comme le meilleur choix. La combinaison de l'échantillonnage dense avec les descripteurs HOGHOF fonctionne le mieux sur le plus difficile des ensembles de données, le Hollywood-2. Sur l'ensemble UCF-Sports, c'est le descripteur HOG3D qui donne les meilleurs résultats en combinaison avec l'échantillonnage dense.

2. Bag-of-Features avec éléments non locaux

Les caractéristiques locales et les descripteurs ne peuvent fournir qu'un pouvoir discriminatoire limité, ce qui conduit à une ambiguïté entre les caractéristiques et des résultats de reconnaissance sous-optimaux. Dans cette partie de la thèse, nous proposons de désambiguïser les caractéristiques spatio-temporelles locales et d'améliorer la reconnaissance d'actions, en intégrant des éléments non-locaux additionnels à la représentation des sacs de caractéristiques (BoF). À cette fin, nous décomposons la vidéo en classes régionales et augmentons les caractéristiques locales avec les labels de classes régionales correspondants. Par exemple, les régions d'un *parking lot* et *side walks* sur la Figure 1 vont être probablement corrélées à des actions spécifiques, telles que *opening a trunk* et *running*. La propagation des labels régionaux au niveau des caractéristiques locales dans cet exemple est alors censée améliorer le pouvoir distinctif des caractéristiques locales par rapport aux actions particulières.

Nous utilisons ici la méthode des sacs de caractéristiques et représentons les vidéos avec les descripteurs Harris3D [88] et HOGHOF. Les descripteurs de caractéristiques sont quantifiés vectoriellement en utilisant soit le dictionnaire visuel entraîné avec l'algorithme k-means, soit une méthode de quantification supervisée basée sur les ERC-Forests [119]. Notre représentation vidéo basée sur les BoF correspond aux histogrammes normalisés l_1 des mots visuels. Pour enrichir la représentation BoF, nous proposons de décomposer la vidéo en un ensemble de régions r associées aux labels l, $l \in \left\{L^1, \ldots, L^M\right\}$ telles que les régions associées aux mêmes labels partagent des propriétés communes. Nous accumulons ensuite un histogramme BoF séparé h^i à partir de toutes les caractéristiques au sein des

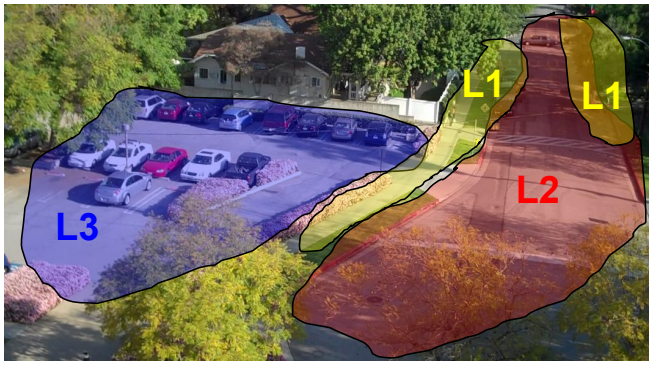


Figure 1: Les différentes régions dans la vidéo telles que routes, trottoirs et parkings sont très souvent accompagnées d'actions spécifiques (par ex : conduite, course, ouverture du coffre) et peuvent fournir des informations prioritaires pour la reconnaissance d'actions.

régions labélisées L^i. Un descripteur vidéo (appellé *canal*) est construit en concaténant les histogrammes BoF pour tous les labels de la région, c'est à dire, $x = \left[h^1, \ldots, h^M\right]$ comme illustré à la Figure 2. Pour classer les actions, nous utilisons une SVM avec le noyau χ^2 et le noyau multicanal [193] (c'est-à-dire le produit des noyaux) pour unir de multiple canaux.

Nous testons notre approche en utilisant des méthodes de segmentation préexistantes et facilement utilisables et nous explorons des stratégies de segmentation alternatives pour (a) pour améliorer la discrimination des différentes classes d'actions et (b) pour réduire les effets des erreurs de chaque approche de segmentation. Ensuite, nous résumons rapidement les cinq types de segmentation vidéo utilisées, illustrés à la Figure 3.

(i) *Spatio-temporel grilles (STGrid-24)* : Nous divisons chaque vidéo en un ensemble de 24 grilles spatio-temporelles prédéfinies [91] qui résultent en 24 canaux.

(ii) *Segmentation de mouvement avant-plan/arrière-plan (Motion-8)* : Nous segmentons une vidéo en régions premier-plan et arrière-plan, à l'aide d'une segmentation du mouvement. Les histogrammes de caractéristiques pour les 2 types de régions et 4 valeurs de seuil de segmentation génèrent ainsi 8 canaux.

(iii) *Détection d'action. (Action-12)* : Nous entraînons le détecteur d'action de Felzenszwalb [40] sur des images d'actions recueillies sur le Web et nous segmentons la vidéo en régions action et non-action en fonction des détections et de leurs boîtes englobantes associées et selon six valeurs de seuil de détection. Nous générons ainsi 12 canaux par action correspondant à 6 valeurs seuil et aux 2 types de régions.

(iv) *Détection personne (Person-12)* : Nous utilisons le détecteur de personne Calvin [1] et

[1]Disponible sur: `http://www.vision.ee.ethz.ch/calvin/calvin_upperbody_detector`

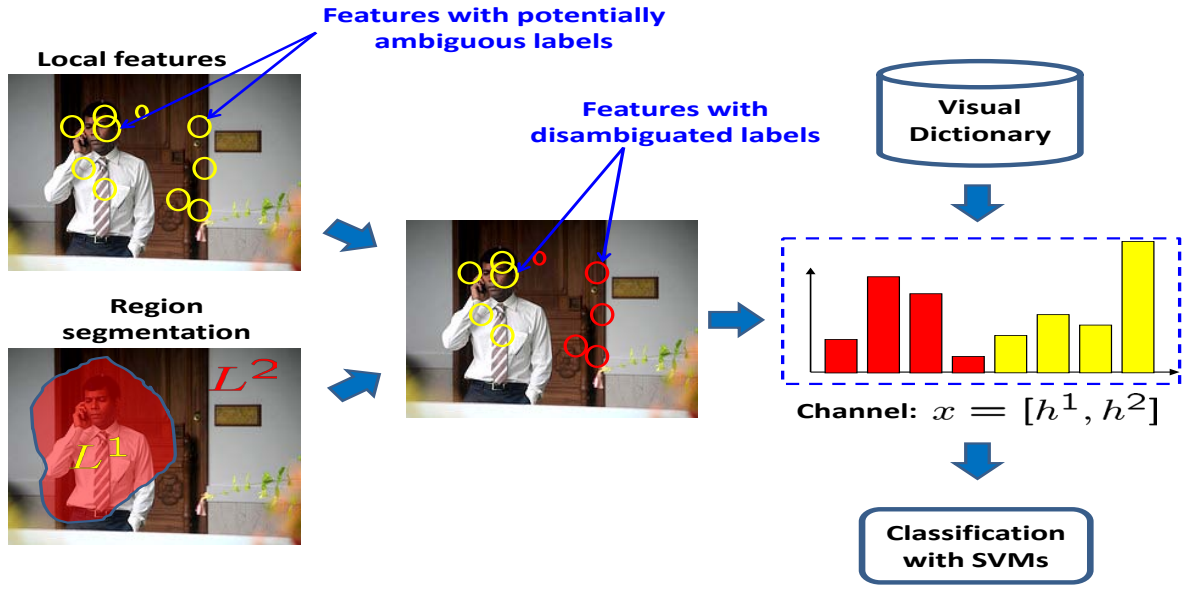


Figure 2: Une illustration de notre approche pour désambiguïser les descripteurs locaux avec l'assistance de la segmentation vidéo sémantique.

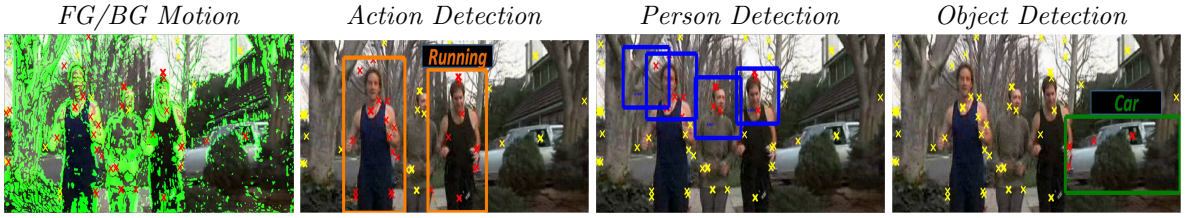


Figure 3: Illustration de l'extraction de zones sémantiques et de la séparation de caractéristiques dans les vidéos.

segmentons la vidéo en régions personne et non-personne. Comme pour Action12, nous obtenons 12 canaux pour les 6 valeurs de seuil et les 2 types de régions.

(v) *Détection d'objets (Objects-12)* : Nous utilisons les détecteurs d'objets de Felzenszwalb pré-entraînés sur Pascal VOC 2008 [40] et nous segmentons la vidéo entre les régions objet et non-objet pour quatre classes d'objets : voiture, chaise, table, et sofa. Nous générons 12 canaux par classe d'objets pour 6 valeurs de seuil et les 2 types de régions, comme pour les canaux Action12 et Person12 ci-dessus.

Nous rapportons les résultats de classification d'actions sur l'ensemble Hollywood-2 [111] en utilisant la moyenne des précisions moyennes (mAP). Le Tableau 4 compare les résultats de base pour les deux méthodes de quantification alternatives. Il s'avère que la quantification supervisée ERC-Forest apporte de meilleurs résultats que la quantification non-supervisée k-means. De plus, le Tableau 5 présente les résultats pour les canaux

Channels	Performance (mean AP)
BoF with k-means	0.481
BoF with ERC-Forest	**0.482**
STGrid-24 with k-means	0.509
STGrid-24 with ERC-Forest	**0.525**

Table 4: Performance de la classification sur le canal de référence sur l'ensemble des données Hollywood-2 [111].

Video channels	Performance (mean AP)
Motion-8	0.503
Person-12	0.496
Objects-12	0.490
Action-12	**0.526**
STGrid-24 + Motion-8	0.533
STGrid-24 + Person-12	0.535
STGrid-24 + Objects-12	0.530
STGrid-24 + Action-12	**0.560**
STGrid-24 + Motion-8 + Action-12 + Person-12 + Objects-12	**0.553**

Table 5: Performance sur chacun des canaux et leurs différentes combinaisons.

individuels de même que pour leurs combinaisons en utilisant la quantification ERC-Forest. Nous voyons maintenant que tous les nouveaux canaux améliorent les performances de base lorsqu'ils sont combinés avec les canaux STGrid24. Plus encore, la combinaison de tous les canaux améliore encore plus significativement les performances de base jusqu'à mAP 0.553. En conclusion, la méthode proposée améliore la classification d'actions de façon significative et possède un réel potentiel pour pouvoir profiter ultérieurement de stratégies de segmentation additionnelles.

3. Attribute Bank pour une reconnaissance d'actions

Inspiré par les récentes avancées dans la reconnaissance d'objets et de scènes basée sur attributs (par exemple, [43, 83, 85, 39, 165]), le présent travail vise à représenter les vidéos en se basant sur des attributs visuels de haut niveau dotés de sens. À cette fin, nous considérons un éventail varié d'attributs incluant de simples objets (comme voiture, chaise, table, etc.), des actions statiques, des personnes de même que des poses distinctives. Notre cadre se sert d'un classificateur pré-entraîné pour chaque attribut, entraîné sur un grand nombre d'images statiques. Suivant l'approche de Object Bank [98, 99] ("banque d'objets"), nous appliquons tous les classificateurs sur chaque trame à des échelles multiples. Pour chaque attribut, nous calculons la valeur maximale de

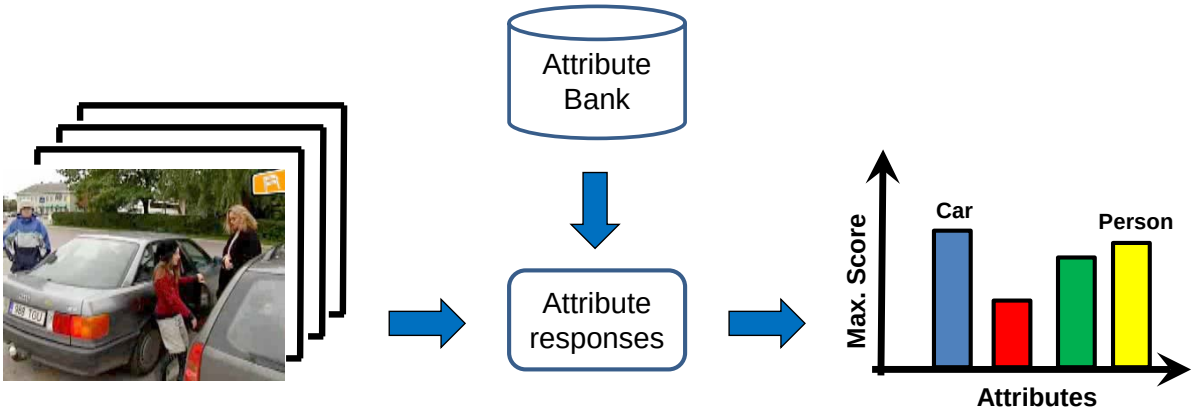


Figure 4: **Illustration d'Attribute Bank.** Un éventail de classificateurs d'attribut est appliqué sur une séquence vidéo, et la valeur de réponse maximale correspondant à chaque classificateur d'attribut est ensuite concaténée en une représentation vectorielle (se référer au texte pour plus d'information).

réponse spatio-temporelle de ce filtre. La représentation vidéo finale est la concaténation des valeurs de réponse maximale pour chaque classificateur d'attribut. Nous appellons cette représentation *Attribute Bank*. De plus, la représentation de la banque d'attributs ne possède pas de vocabulaire et peut donc être directement calculée.

La représentation Attribute Bank. Soit une séquence vidéo v, un volume de réponse de filtre d'attributs Ω_{a_k} est obtenu en estimant la probabilité d'occurrence $p(a_k|v)$ pour le classificateur d'attributs a_k à des échelles multiples. Soit n le nombre total de classificateurs d'attribut. Nous utilisons la technique appelée max-pooling sur les volumes de réponses obtenus n et concaténons le résultat maximal de chaque classificateur d'attribut a_i en une représentation vectorielle:

$$\left[\max_{(x,y,t)} \Omega_{a_1}, \ldots, \max_{(x,y,t)} \Omega_{a_n}\right], \tag{1}$$

où (x, y, t) dénote le volume spatio-temporel pour le pooling maximal lequel dans ce cas est la vidéo entière, comme illustré à la Figure 4. De plus, nous utilisons des grilles spatio-temporelles de 24 niveaux [91] et divisons chaque volume de réponse Ω_{a_i} en 24 types de grilles différentes. Chaque grille divise un volume de réponse en un ensemble de cellules prédéfinies. Pour chaque grille avec m cellules, la représentation vidéo correspondante est la concaténation de caractéristiques d'attribut dans chaque cellule c de la grille:

$$\left[\max_{(x,y,t)_c} \Omega_{a_1}, \ldots, \max_{(x,y,t)_c} \Omega_{a_n}\right]_{c=1}^{m}. \tag{2}$$

En conséquence, une séquence vidéo est encodée en 24 différents canaux grilles auxquelles on se réfère comme la représentation de la banque d'attributs.

Classificateurs d'attribut pour le Attribute Bank. Nous utilisons des classificateurs SVM latents [40] (décrits dans la Section 2) entraînés pour les quatre classes d'objets (*car*, *chair*, *table*, et *sofa*), et huit classes d'actions (*answering phone*, *hugging*, *hand shaking*, *kissing*, *running*, *eating*, *driving*, et *sitting*). De plus, nous utilisons le détecteur Calvin pour la partie haute du corps[2] pour détecter l'attribut *person* dans les vidéos. On se réfère à la représentation de la banque d'attributs basée sur les attributs d'objets, d'actions et de personnes mentionnés plus haut comme les canaux *OAP-Bank*. De plus, nous utilisons comme attributs 150 types différents de *poselet* [17]. Les poselets sont des détecteurs basés sur des parties et opèrent sur de nouvelles parties du corps. Ces détecteurs spécialisés ont été entraînés sur une base de données images relativement importante de parties de corps annotées manuellement et insensible aux variations dans l'apparence visuelle des images. Nous proposons ici de calculer la représentation de la banque d'attributs avec 150 différents types de poselets comme attributs. Nous nous référons à ces canaux vidéo comme *Poselet-Bank*.

Pour la classification d'actions à l'aide de la banque et des canaux Poselet-Bank, nous utilisions une SVM non-linéaire avec un noyau RBF. Comme référence de comparaison, nous utilisons STGrid-24 canaux (introduites dans la Section 2) et nous employons une SVM non-linéaire avec un noyau χ^2 pour la classification. De plus, nous combinons les différents canaux vidéo en utilisant un noyau multicanal [193] et nous utilisons une approche *one-against-rest* pour la classification.

Nous évaluons la performance de la représentation de notre banque d'attributs sur l'ensemble de données Hollywood-2. Le Tableau 6 présente les résultats pour les canaux de référence STGrid-24 de même que pour les canaux basés sur la banque d'attributs que nous proposons et sur leurs combinaisons. Nous observons que les performances individuelles des canaux de la banque OAP (c'est à dire, 0,413 mAP) et de la banque de Poselet (c'est à dire, 0,344 mAP) sont inférieures à celles des canaux de base STGrid-24 (c'est à dire, 0,525 mAP). Néanmoins, lorsque les canaux des banques OAP et Poselet-Bank sont combinés avec les canaux de base STGrid-24, leurs performances s'améliorent respectivement d'environ 3% et 2% sur la base. Les performances supérieures des canaux de la banque OAP comparées à ceux de la banque Poselet sont probablement dues au fait que le premier encode la présence/absence des actions spécifiques (*answering phone*,

[2]Disponible sur: `http://www.vision.ee.ethz.ch/calvin/calvin_upperbody_detector`

Channels	STGrid-24 (Baseline)	OAP-Bank	OAP-Bank + STGrid-24	Poselet-Bank	Poselet-Bank + STGrid-24	OAP-Bank + Poselet-Bank + STGrid-24
mean AP	0.525	0.413	0.558	0.344	0.541	**0.571**
AnswerPhone	0.259	0.347	0.360	0.230	0.292	**0.366**
DriveCar	0.859	0.694	0.880	0.571	0.876	**0.881**
Eat	**0.607**	0.248	0.580	0.243	0.533	0.564
FightPerson	**0.749**	0.482	0.733	0.282	0.695	0.705
GetOutCar	0.447	0.307	0.426	0.303	0.438	**0.457**
HandShake	0.285	0.471	0.512	0.392	0.433	**0.523**
HugPerson	**0.461**	0.283	0.420	0.136	0.406	0.407
Kiss	0.569	0.521	**0.668**	0.398	0.600	0.665
Run	0.698	0.577	0.700	0.649	**0.767**	0.762
SitDown	**0.589**	0.366	0.556	0.381	0.573	0.566
SitUp	0.202	0.193	0.244	0.138	0.288	**0.334**
StandUp	0.574	0.473	**0.617**	0.404	0.596	0.616

Table 6: Performance en terme de précision moyenne par classe (AP) des différents canaux/combinaisons de canaux sur l'ensemble des données d'Hollywood-2.

hugging, *hand shaking*, *kissing*, *running*, *eating*, *driving*, et *sitting*), qui sont directement liés aux classes d'action dans l'ensemble de données Hollywood-2. De plus, les canaux de la banque OAP capturent l'information sur différents objets (*car*, *chair*, *table*, et *sofa*), ce qui permet également de distinguer entre les classes d'action.

Plus encore, lorsque les canaux des banques OAP et Poselet sont tous les deux combinés aux canaux de base STGrid-24, nous obtenons une amélioration de 4,6% sur la base. Nous pouvions voir que nos canaux basés sur la banque d'attributs aident à améliorer huit de nos douze classes d'actions (la précision moyenne est notée en gras). Cela démontre que la représentation à l'aide la banque d'attributs proposée, capturant des informations de haut niveau sur les vidéos, est réellement très distinctive. De plus, cela montre que les caractéristiques de la banque d'attributs enrichissent les caractéristiques de bas niveau en les combinant à des informations de haut niveau sur les vidéos.

4. Descripteurs locaux de mouvement caractéristiques d'actions

Des changements significatifs d'angle de prise de vue et d'apparence des objets d'une scène modifient profondément les descripteurs locaux classiques et affectent en conséquence les approches basées sur de telles représentations locales. Pour répondre à ce problème, nous proposons dans cette partie de la thèse une approche supervisée pour apprendre des descripteurs dynamiques locaux à partir d'un large ensemble de données vidéo annotées. L'idée principale de cette méthode est de construire des représentations articulaires exploitant les dynamiques propres à certaines actions tout en incorporant par apprentissage l'invariance aux variations de point de vue, d'illumination, d'habillement, entre autres facteurs. Nous proposons une approche supervisée d'apprentissage d'"*Actlets*": il

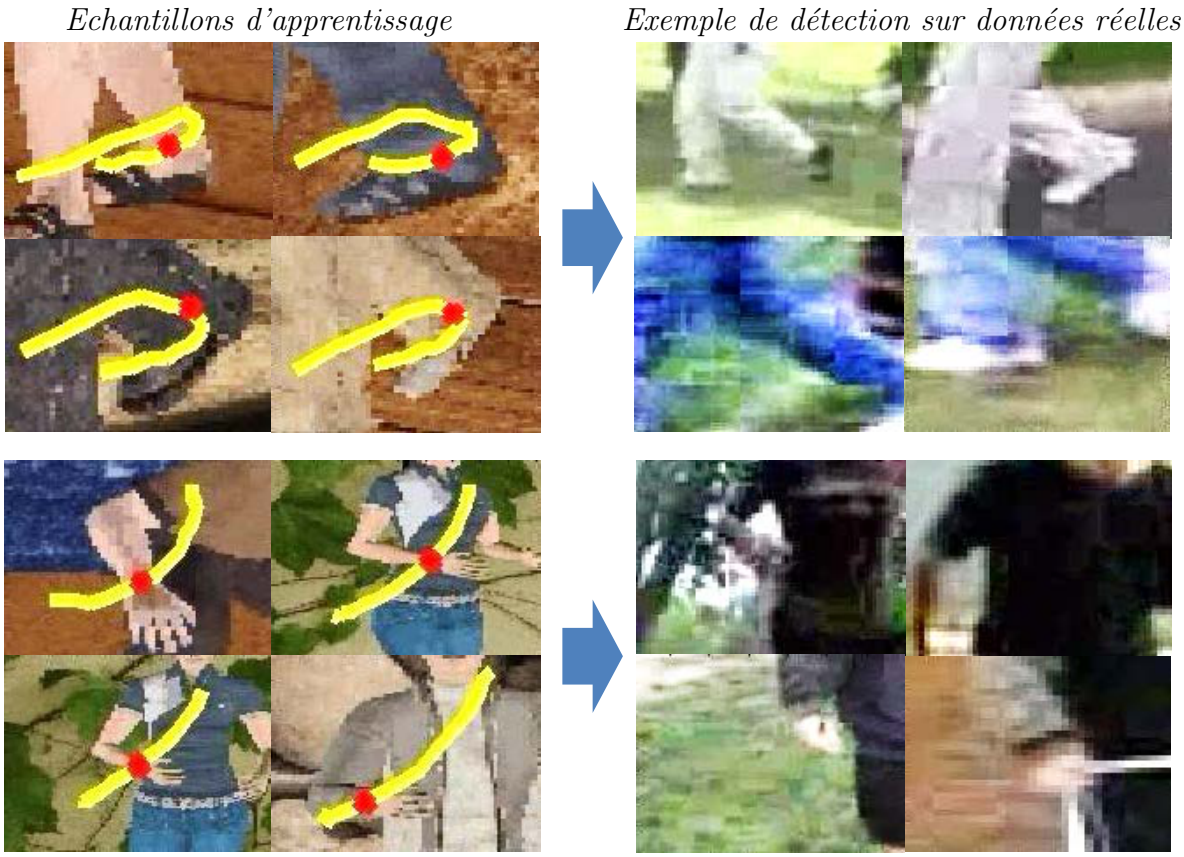


Figure 5: **Illustration des *Actlets***. Les *Actlets* sont des détecteurs spécialisés qui sont appris sur données synthétiques (à gauche) et appliqués sur des vidéos réelles (à droite). Les trajectoires des articulations automatiquement annotées sont affichées sur la gauche.

s'agit de détecteurs de parties spécifiques du corps animées d'un mouvement spécifique. L'apprentissage des *Actlets* exige une quantité importante de données annotées d'entrainement. Pour recueillir de telles données, nous proposons d'éviter la tâche difficile d'annotation manuelle de vidéo en lui substituant une génération automatique de telles données sur la base de vidéos synthètiques issues de l'animation d'avatars par capture de mouvement (voir Figure 5). Nous utilisons ensuite les *A*ctlets pour la reconnaissance d'actions humaines dans des données vidéo réelles.

Base de données synthétiques de mouvements humains. Pour entraîner un ensemble représentatif d'*Actlets*, nous avons besoin d'une quantité relativement importante de données d'apprentissage. Ces données d'entraînement doivent couvrir un large éventail de mouvements humains et devraient contenir le positionnement des articulations du corps au fil du le temps. De plus, un vaste panel de variations en termes d'apparence (par exemple, les vêtements et le fond), de point de vue, d'illumination, de mouvement

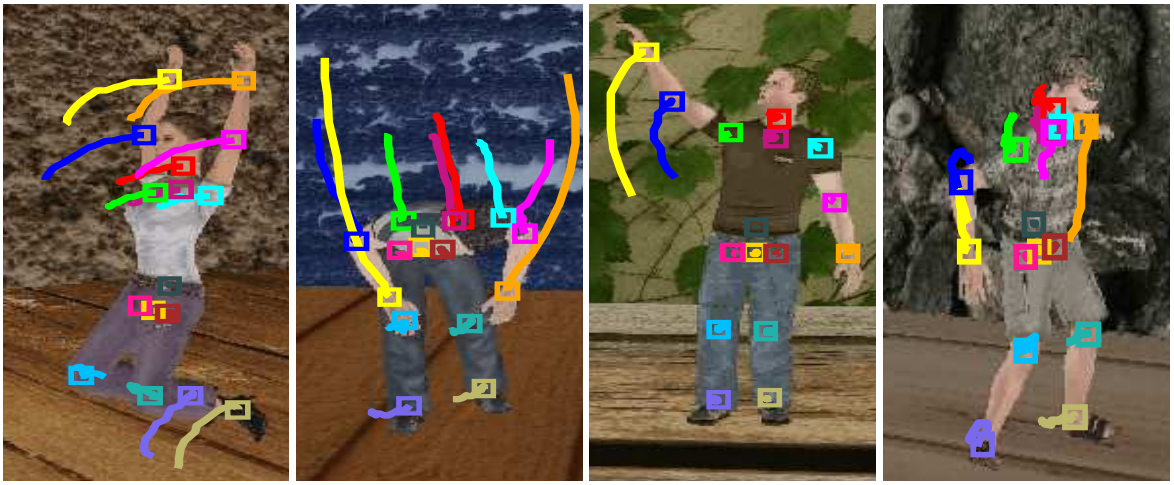


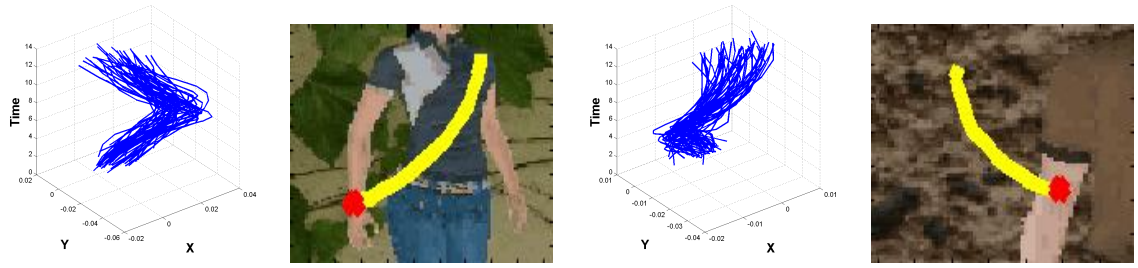
Figure 6: **Illustration de la base de données synthétiques.** Exemples issus de notre base de données synthétiques illustrant la variabilité des vidéos générées en termes de point de vue, d'arrière-plan, de caractéristiques physiques, d'habillement ou de mouvement. Les courbes de couleur montrent les trajectoires des articulations automatiquement annotées par projection des données MoCap.

de caméra et de style d'action est nécessaire pour couvrir la variabilité attendue dans les vidéos à traiter. L'annotation manuelle des articulations du corps et de leurs mouvements dans des vidéos étant très chronophage et donc peu pratique, nous proposons de faire appel aux techniques d'animation à base de capture de mouvement pour construire un ensemble synthétique de données. Le principal avantage de cette approche est l'accès direct à l'information sur les positions des articulations du corps dans chaque vidéo synthétisée via la projection 2D des positions 3D de ces articulations fournies par MoCap. Nous effectuons un *retargeting* des séquences MoCap de la base CMU[3] sur des humanoïdes 3D avec l'aide d'Autodesk MotionBuilder 2011 et nous réalisons le rendu des vidéos pour un ensemble donné de points de vue. Nous utilisons dix personnages 3D, hommes et femmes aux proportions et tenues variées. Nous calculons les vidéos pour trois points de vue différents (avant, droite et gauche relativement au personnage) tout en utilisant cinq fonds statiques différent. De plus, nous simulons un panoramique de caméra qui suit les mouvements du personnage dans chaque vidéo. Nous calculons une vidéo pour chaque séquence MoCap de la base de données CMU, tout en choisissant au hasard un personnage, un fond et un angle de prise de vue. Comme résultat, nous obtenons 2549 séquences vidéos synthétiques (voir Figure 6).

Apprentissage des *Actlets*. Nous considérons le mouvement de 9 articulations

[3]Disponible à `http://mocap.cs.cmu.edu`

(a) clusters pour 1 articulation



(b) clusters pour paires d'articulations

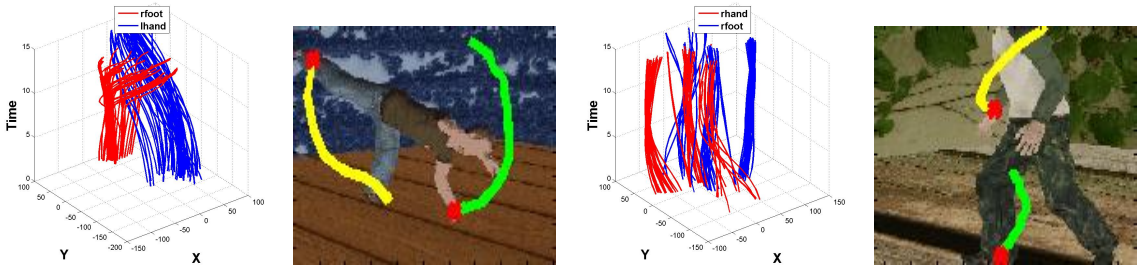


Figure 7: **Illustration des groupes de trajectoires d'articulation.** Il existe deux types de groupes : (a) basés sur les mouvements d'une seule articulation, et (b) basés sur les mouvements conjoints de deux articulations. Toutes les trajectoires d'un même groupe sont tracées dans un même graphe à l'aide de courbes bleues et rouges. Une image typique est également affichée pour chaque groupe.

(tête, épaules gauche/droite, poignets gauche/droit, genoux gauche/droit et chevilles gauche/droite) permettant d'accéder à une description riche des d'actions. Ces 9 articulations du corps sont traitées de deux façons : (a) regroupement de mouvements similaires associés à chaque articulation séparément, et (b) regroupement de mouvements similaires associés à deux articulations. Pour chacune des 9 articulations et pour chaque vidéo synthétique, la trajectoire 2D associée est subdivisée en sous-trajectoires qui se chevauchent, chacune d'une longueur de $L = 15$ instants. La forme d'une sous-trajectoire encode localement le mouvement de l'articulation concernée. Suivant [130], nous représentons la forme d'une sous-trajectoire à l'aide de vitesses. Pour regrouper les mouvements similaires associés à chaque articulation (ou à une paire d'articulations), nous effectuons un *clustering* par k-moyennes (nous fixons $k = 75$) de toutes les sous-trajectoires associées à chacune des articulations (ou paires d'articulations) dans l'ensemble des 2549 vidéos synthétiques. Nous réalisons un *clustering* par prise de vue et un autre indépendemment de la prise de vue, les trajectoires issues des trois

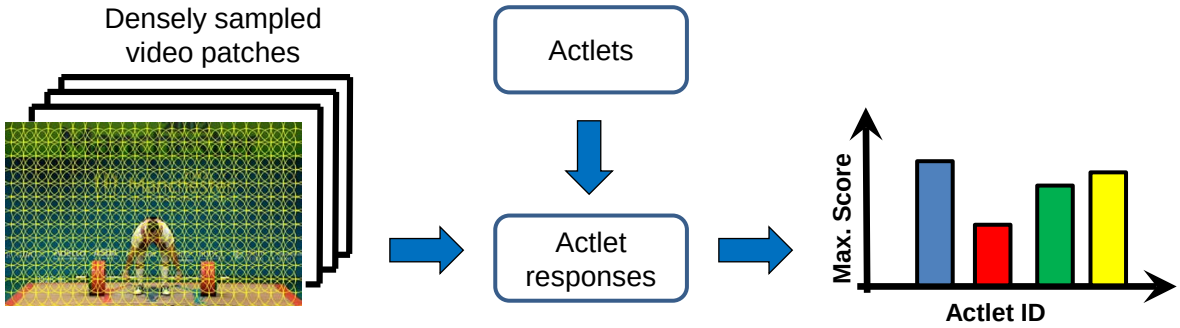


Figure 8: **Illustration de la représentation à base d'*Actlets*.** Les *Actlets* sont appliqués sur une séquence vidéo densément échantillonnée, et la réponse maximale correspondant à chaque classificateur d'*actlet* est par la suite concaténée dans une représentation vectorielle (se référer au texte pour plus d'information).

différents points de vue étant ainsi partitionnées séparément dans le premier cas et conjointement dans le second. Pour sélectionner des clusters significatifs, nous classons tous les *clusters* associés à une articulation (ou paire d'articulations) par ordre décroissant de la somme des distances aux autres clusters et ne gardons que les $n = 50$ premiers. La Figure 7 montre des exemples de tels clusters basés sur 1 ou sur 2 articulations. Afin d'entraîner un *actlet* pour une articulation (ou une paire d'articulations) donnée et pour un type de mouvement, nous extrayons des fragments vidéo dans le voisinage des trajectoires issues d'un groupement. Ces fragments sont utilisés comme échantillons positifs pour l'entraînement de l'*actlet*. Pour obtenir des échantillons négatifs, nous extrayons au hasard 10.000 fragments vidéo synthétiques, correspondant à des trajectoires issues des $n - 1$ *clusters* restants pour la même articulation (ou paire d'articulationss). Nous decrivons les fragments vidéos extraits à l'aide de HOG, HOF et leurs combinaisons, c'est-à-dire les descripteurs HOGHOF [91]. Nous entraînons ensuite un SVM linéaire de type Hellinger sur ces descripteurs. De cette façon, pour chaque type de descripteur, nous obtenons un total de 1000 classificateurs SVM linéaires pour les *Actlets* associés à 1 articulation[4] et 1164 classificateurs SVM linéaires pour les *Actlets* basés sur 2 articulations [5], correspondant aux cas spécifiques à la prise de vue et à celui indépendant de la prise de vue.

[4]Avant: 9 articulation×50 clusters + gauche/droit: 2×5 articulation×50 clusters + indépendant du point de vue: 9 articulation×50 clusters. Nous entraintons des *Actlets* uniquement pour des clusters comptant un minimum de 50 trajectoires.

[5]Avant: 36 paires d'articulations×50 clusters + gauche/droit: 2×10 paires d'articulations×50 clusters + indépendant du point de vue: 36 paires d'articulations×50 clusters. Nous entrainons des *Actlets* uniquement pour des clusters comptant un minimum de 50 trajectoires.

Représentation vidéo à base d'*Actlets*. Étant donnée une vidéo v, nous extrayons de façon dense des fragments vidéos et les représentons par descripteurs HOG, HOF et HOGHOF. Pour chaque type de descripteur et chaque type d'*Actlets* (soit basés sur 1 articulation, soit sur une paire d'articulations), nous obtenons un ensemble de scores pour l'ensemble des *actlet* appris pour ce type de descripteur. De cette façon, nous obtenons un réponse volumique de filtre *actlet*, Ω_{a_k}, pour l'*actlet* a_k. Soit n le nombre total de classificateurs. Nous réalisations une agrégation de ces scores par maximisation sur les n volumes de réponses et concaténons le score maximal de chaque classificateur a_i dans un vecteur de représentation:

$$\left[\max_{(x,y,t)} \Omega_{a_1}, \ldots, \max_{(x,y,t)} \Omega_{a_n}\right], \tag{3}$$

où (x, y, t) dénote le volume spatio-temporel sur lequel s'effectue l'aggrégation, qui est dans ce cas la vidéo entière, comme illustré sur la Figure 8. Suivant la représentation par "banc d'attributs", nous utilisons des grilles spatio-temporelles de 24 niveaux [91] et divisons chaque volume de réponse Ω_{a_i} en 24 différents types de grilles. Pour chaque grille avec m cellules, la représentation vidéo correspondante est formée par la concaténation des attributs d'*actlet* dans chaque cellule c de la grille :

$$\left[\max_{(x,y,t)_c} \Omega_{a_1}, \ldots, \max_{(x,y,t)_c} \Omega_{a_n}\right]_{c=1}^{m}. \tag{4}$$

En conséquence, les représentations vidéos correspondantes (*Actlets1HOG*, *Actlets1HOF*, *Actlets1HOGHOF*, *Actlets2HOG*, *Actlets2HOF*, et *Actlets2HOGHOF*) se composent chacune de 24 canaux spatio-temporels.

Pour la reconnaissance d'actions basée sur les canaux d'*actlet*, nous utilisons un SVM non-linéaire avec noyau RBF. Nous utilisons les représentations vidéo BoF (basées sur les points d'intérêt de type Harris3D [88] et les descripteurs HOGHOF [91]) comme base de référence et nous employons un SVM non-linéaire avec noyau χ^2 pour la classification. Pour combiner différents canaux, nous utilisons une méthode multi-noyaux. [193].

Résultats sur UCF-Sports Le tableau 7 présente les résultats pour la référence à base de BoF ainsi que pour tous les canaux *actlet*. Nous remarquons que les performances de tous les canaux *actlet* sont proches de celles de la référence. Parmi les *Actlets*, les performances des *Actlets* basés sur HOG et sur HOF sont comparables ; tandis que les *Actlets* HOGHOF présentent de meilleurs résultats, suggérant ainsi que les informations sur l'apparence (c'est-à-dire HOG) et celles sur le mouvement (c'est-à-dire HOF) peuvent

Canal %	BoF (réf.)	Actlets1 HOG	Actlets1 HOF	Actlets1 HOGHOF	Actlets2 HOG	Actlets2 HOF	Actlets2 HOGHOF
Précision moyenne	077.25	075.02	074.46	**077.77**	075.63	076.07	076.82
Dive	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
GolfSwing	066.67	077.78	050.00	**088.89**	077.78	066.67	083.33
KickBall	085.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
WeightLift	**100.00**	083.33	083.33	083.33	083.33	083.33	083.33
HorseRide	**066.67**	058.33	050.00	050.00	058.33	041.67	041.67
Run	076.92	**084.62**	053.85	069.23	076.92	061.54	061.54
SwingPommel	085.00	085.00	**100.00**	**100.00**	095.00	**100.00**	**100.00**
Skateboard	016.67	008.33	**033.33**	016.67	016.67	**033.33**	**033.33**
Walk	**090.91**	068.18	081.82	077.27	063.64	081.82	072.73
SwingHighBar	084.62	084.62	**092.31**	**092.31**	084.62	**092.31**	**092.31**

Table 7: Performance en précision pour les différents canaux sur la base de données UCF-Sports.

Canal %	BoF (réf.)	Actlets1 HOG + BoF	Actlets1 HOF + BoF	Actlets1 HOGHOF + BoF	Actlets2 HOG + BoF	Actlets2 HOF + BoF	Actlets2 HOGHOF + BoF	Kläser et al. [78]
Précision moyenne	077.25	079.88	079.22	081.29	082.21	081.90	**083.24**	083.13
Dive	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
GolfSwing	066.67	083.33	072.22	083.33	**088.89**	**088.89**	**088.89**	079.60
KickBall	085.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	083.90
WeightLift	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	071.64
HorseRide	**066.67**	**066.67**	058.33	**066.67**	**066.67**	**066.67**	**066.67**	059.20
Run	**076.92**	**076.92**	061.54	**076.92**	**076.92**	**076.92**	**076.92**	076.00
SwingPommel	085.00	085.00	095.00	090.00	095.00	095.00	**100.00**	095.00
Skateboard	016.67	025.00	025.00	025.00	025.00	008.33	016.67	**083.30**
Walk	090.91	077.27	**095.46**	086.36	077.27	090.91	090.91	082.64
SwingHighBar	084.62	084.62	084.62	084.62	092.31	092.31	092.31	**100.00**

Table 8: Performance en précision pour les différents canaux et combinaisons de canaux sur la base de données UCF-Sports.

être utilement combinées dans l'apprentissage de bons *Actlets*. De plus, les performances obtenues avec Actlet1HOGHOF sont légèrement meilleures que celle de la référence. La Table 8 présente les résultats pour les canaux *actlet* combinés à celui du BoF. Nous voyons que chaque canal *actlet* améliore les performances de base. Les meilleures performances sont atteintes par les canaux Actlet2HOGHOF, en l'occurrence 83,24%, ce qui constitue une amélioration d'environ 6% de la référence BoF (celle-ci étant à 77,25%). Nous comparons également nos résultats avec ceux de Kläser et al. [78] dans le Tableau 8. Nous observons que les canaux *actlet* permettent d'améliorer les résultats pour 7 des 10 classes d'actions (les meilleurs résultats sont marqués en gras).

Résultats sur YouTube-Actions. Le tableau 9 présente les résultats pour la référence ainsi que pour tous les canaux *actlet* individuels. Dans ce cas, les *Actlets* basés sur HOF et sur HOGHOF fonctionnent mieux que le canal BoF de référence. Parmi les

Canal %	BoF (réf.)	Actlets1 HOG	Actlets1 HOF	Actlets1 HOGHOF	Actlets2 HOG	Actlets2 HOF	Actlets2 HOGHOF
Précision moyenne	62.95	56.06	64.57	65.66	58.87	63.27	**67.09**
Bike	71.51	81.08	71.24	**84.46**	81.29	69.24	80.85
Dive	85.00	59.00	**90.00**	80.00	74.00	84.00	81.00
Golf	73.00	88.00	76.00	86.00	**89.00**	77.00	**89.00**
SoccerJuggle	50.00	10.00	**51.00**	36.00	20.00	**51.00**	41.00
TrampolineJump	**74.00**	58.00	64.00	61.00	62.00	64.00	68.00
HorseRide	72.00	71.00	70.00	75.00	**76.00**	69.00	75.00
BasketballShoot	33.67	41.67	41.00	**46.00**	31.67	45.00	**46.00**
VolleyballSpike	73.00	72.00	80.00	82.00	72.00	79.00	**83.00**
Swing	71.00	62.00	**80.00**	76.00	60.00	76.00	**80.00**
TennisSwing	46.00	35.00	46.00	**56.00**	52.00	42.00	**56.00**
Walk	**43.30**	38.94	40.99	39.82	29.61	39.70	38.19

Table 9: Performance en précision pour les différents canaux sur la base de données YouTube-Actions.

Canal %	BoF (réf.)	Actlets1 HOG + BoF	Actlets1 HOF + BoF	Actlets1 HOGHOF + BoF	Actlets2 HOG + BoF	Actlets2 HOF + BoF	Actlets2 HOGHOF + BoF	Liu et al. [101]
Précision moyenne	62.95	67.03	69.89	70.99	66.52	68.56	70.07	**71.21**
Bike	71.51	82.76	81.42	**85.43**	77.40	78.75	82.07	73.00
Dive	85.00	82.00	**90.00**	89.00	86.00	**90.00**	88.00	81.00
Golf	73.00	87.00	87.00	86.00	**91.00**	87.00	89.00	86.00
SoccerJuggle	50.00	49.00	**59.00**	55.00	48.00	57.00	57.00	54.00
TrampolineJump	74.00	75.00	74.00	75.00	73.00	72.00	75.00	**79.00**
HorseRide	72.00	71.00	73.00	**75.00**	69.00	70.00	73.00	72.00
BasketballShoot	33.67	39.67	39.67	40.67	34.33	40.67	41.67	**53.00**
VolleyballSpike	73.00	84.00	85.00	**87.00**	79.00	82.00	85.00	73.30
Swing	71.00	72.00	**77.00**	**77.00**	76.00	**77.00**	**77.00**	57.00
TennisSwing	46.00	48.00	54.00	60.00	55.00	53.00	56.00	**80.00**
Walk	43.30	46.90	48.70	50.83	43.03	46.70	47.03	**75.00**

Table 10: Performance en précision pour les différents canaux et combinaisons de canaux sur la base de données YouTube-Actions.

Actlets, les gains en performance s'échelonnent de la façon suivante: HOG-based<HOF-based<HOGHOF-based. Les meilleures performances sont obtenues par les canaux Actlet2HOGHOF. Le Tableau 10 présente ensuite les résultats obtenus en combinant les canaux *actlet* avec le canal BoF. Nous remarquons que chaque combinaison améliore les performances de la référence. Les meilleures performances sont obtenues par les canaux Actlet1HOGHOF, soit 70,99% de précision, ce qui représente une amélioration d'environ 8% par rapport à la base BoF (qui est, elle, à 62,95%). Nous comparons également nos résultats avec ceux des travaux de Liu et al. [101]où la base de données fut introduite (Table 10). Nous pouvons observer que les canaux *actlet* ont permis d'obtenir une amélioration des résultats de classification pour 7 des 11 classes d'action (les meilleurs résultats sont marqués en gras).

5. Conclusion

Dans ce travail, nous avons exploré de nouvelles représentations locales pour la reconnaissance d'actions humaines dans des données vidéo réelles. Nous avons développé en particulier des représentations locales supervisées qui sont peu couteuses à calculer et permettent d'améliorer le modèle de référence à base de "sac de caractéristiques" (BoF). Nous proposons ainsi plusieures types de descriptions vidéo à caractère discriminant. Leur complémentarité est exploitée dans un cadre de classification par combinaison de noyaux. Les évaluations empiriques sur plusieurs bases de données montrent que ces représentations enrichissent le modèle de référence BoF grâce à l'apport d'une supervision automatique à base de MoCap.

Contents

Chapter 1

Introduction

Contents

Technological advancement, over the past few decades, has revolutionized our lives to the extent that this era can be regarded as the era of "technological revolution". In particular, recent advances in computers, digital cameras, and Internet have contributed in the proliferation of multimedia, especially videos. For instance, YouTube alone uploads about 60 hours of video every minute, and streams 4 billion online videos every day worldwide[1] [128]. Moreover, humans are predominantly the main focus in video, as we are majorly interested in 'ourselves'. We humans can easily interpret a video, based on its visual content. We can easily distinguish between different actions being performed, such as fighting, running, walking, driving, and so on. Nevertheless, neuroscience and related fields are still unclear about how this performance is achieved. While an automated recognition of human actions in video is fascinating, computer vision systems are far behind the capabilities of human visual system.

An automated system for human action recognition would have many practical applications, as for instance:

[1] As of January 2012, source: `http://www.reuters.com`

- **Content-based video search**

  With the explosion of electronic devices (such as tablets, digital cameras, smart phones, etc.), Internet usage, and online publishing, we now have access to a tremendous amount of video data, and it is rising on a massive scale. However, the possibilities to effectively analyze such a huge collection are rather limited. Currently, web search engines (e.g., Google, Yahoo, Bing, etc.) majorly rely on text-based descriptions or captions, in order to retrieve the relevant videos. Automated human action recognition could be used to extract more information directly from videos, which can help to index and categorize them automatically.

- **Smart user interfaces**

  As electronic devices are becoming more and more ubiquitous in our lives, new ways for humans to interact with these devices are being sought. For example, Microsoft's Kinect gaming platform[2] allows users to play controller-free video games. Users can interact in a virtual world using their full bodies in a natural way. The platform achieves this capability by combining information from multiple sensors: a video camera, a depth sensor (based on infrared patterns), and a multi-array microphone. Automated human action recognition can be helpful in developing intelligent user interfaces.

- **Assisted living**

  Automated human action recognition has the potential to be employed for assisted living in smart homes, hospitals, and elderly care centers. For instance, elderly people who are dependent on others in their everyday needs, can be well monitored and assisted through the automatic recognition of their actions. Other application areas include medical diagnosis as well as analysis and optimization of movements in athletics or in dance choreography.

The main focus in this thesis is the automated recognition of human actions in realistic video data, such as movies and online videos, for instance. Human actions in such video data expose large variations due to changes in person appearance and action styles, scale and view-point changes, dynamic backgrounds, illumination conditions, and other factors. Consequently, vision-based human action recognition is not trivial on such an unconstrained and challenging video data. Recently, local video representations based on space-time features have been demonstrated to be effective in realistic settings.

[2] `http://www.xbox.com/en-US/kinect`

The success of local space-time features can be attributed to their mild assumptions about the data and robustness to certain variations in the video. However, such video representations are typically based on order-less collections of low-level video features, also referred to as the *Bag-of-Features* representation. Being purely local in nature, local features yield limited discriminative power, which results in an ambiguous video representation. Moreover, local features are sensitive to the large variations in appearance and motion. To address such limitations, we in this thesis, extend current methods and develop supervised statistical representations for improving human action recognition in realistic and challenging video data, such as Hollywood movies and YouTube videos.

1.1 Problem statement

The area of human action recognition is closely related to other research fields which analyze human motion from images and videos. The recognition of human movements can be performed at various levels of abstraction. Different taxonomies have been proposed in the literature and here we adopt the hierarchy proposed by Moeslund *et al.* [118]: *action primitive* (or *movement*), *action*, and *activity*. An *action primitive* is a basic and atomic movement that can be described at the body-limb level. An *action* is comprised of *action primitives* and describes a (possibly cyclic) whole-body movement. Finally, an *activity* is a larger scale event, which involves a number of subsequent *actions*. *Activities* are often related to the context and environment in which the *actions* are being performed. For instance, the "long jump" can be considered as an *activity*, which involves the subsequent actions: "running", "taking off", "flying", and "landing". The "running" action can be further decomposed into the *action primitives*: "right leg forward", "right arm bend", "right arm forward", "left leg backward", "left arm backward", etc. This thesis is concerned with the recognition of *actions*, which can be defined by action verbs (such as run, walk, eat, fight, etc.), and are typically performed by one or two people. Moreover, the recognition is based only on visual observations, typically by means of one or more video cameras. But of course, actions can also be recognized through other sensory channels, such as audio.

In this thesis, the expression "action recognition" is used as an equivalent to "action classification". Therefore, *action recognition* is the process of naming actions, in the simple form of action verbs, using visual observations. More precisely, given an input video sequence, the objective is to assign it with one or more class labels from a set of known action categories, based only on the visual content.

Action categories which might seem clearly defined to us, such as kicking, punching, waving, etc., can expose very large variability when performed in practice. In particular, when performed by different subjects of different gender and size, and with different speed and style. Furthermore, the background environment heavily influences the visual observation of actions. Therefore, it is utmost important to design an action model, which identifies for each action the distinguishing features, while maintaining an appropriate invariance to all forms of visual variations. This thesis addresses such problems by developing supervised statistical representations, aiming to improve action recognition in challenging video data.

1.2 Challenges

The task of human action recognition is particularly challenging in realistic video data, such as movies and web videos, for instance. Action categories in such video data exhibit a diverse range of variations in their visual appearance, due to many factors. In this section, we discuss the inherent characteristics of realistic video data, which pose major challenges for any artificial action recognition system.

Intra/inter-class variations

The problem of large *intra-class* differences is pertinent in relatively unconstrained and realistic video data, such as movies and online videos. Instances of the same action class can vary a lot in their visual appearance, due to many factors. Figure 1.1 illustrates a large variety of intra-class variations within the same action classes. One important source is the anthropometric differences among individuals performing actions. For instance, walking movements can differ in speed and stride length (see the action "Walking" in Figure 1.1 (b)). Moreover, actions are adapted to the context of their environment. For example, the telephone model (see the action "Answering phone" in Figure 1.1 (a)) drastically affects the way a person uses it. A good action recognition system should be able to generalize over variations within one class and distinguish between actions of different classes (i.e., the *inter-class* variations). For increasing numbers of action classes, this will become more challenging as the overlap between different classes will be higher.

Recording conditions

The recording environment has a major impact on the visual observation of actions. For instance, different lighting and illumination conditions can influence the appearance of the person performing the action. Moreover, person localization might prove harder in

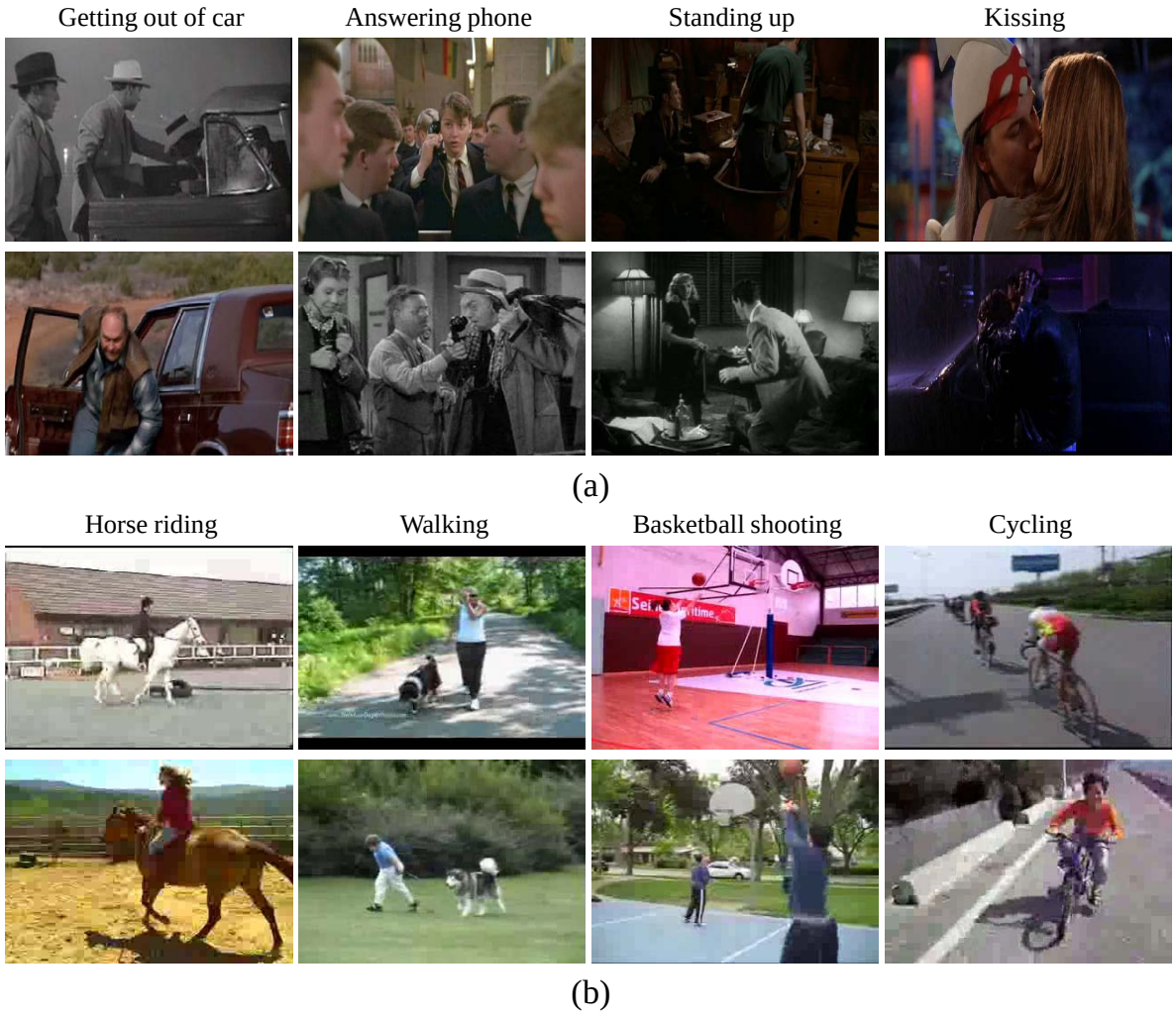


Figure 1.1: **Illustration of action variations in realistic video data**. (a) Sample action categories from Hollywood movies; and (b) sample actions from YouTube videos.

cluttered or dynamic environments, with multiple background motions. Also, parts of the person might be occluded in the recording, which may lead to difficulties in the interpretation of the action being performed.

Furthermore, the same action when observed from different view points, can lead to very different visual observations. When multiple cameras are employed, problems related to view point changes as well as occlusion, can be tackled. Moreover, the scale at which an action is being recorded, is an additional source of visual variation. Camera motion and shake further complicates the visual interpretation of actions in realistic video data. The recording quality can also turn out to be challenging, especially in the low-resolution videos available on the Internet (see Figure 1.1 (b)). A vision-based human action

recognition system should be able to deal with all of these problems.

Temporal variations

It is often assumed that the actions are pre-segmented into video clips, each showing a single action from start to finish, both for training as well as testing. In practice, however, actions are not temporally segmented as the temporal (as well as spatial) action segmentation is a hard problem [121, 116, 101, 183, 190]. Moreover, there can be substantial variation in the rate of performance of an action. The rate at which an action is recorded has an important effect on the temporal extent of the action, which consequently affects the motion estimation. A robust human action recognition system should provide invariance to different rates of execution.

Obtaining and labeling training data

An important limitation is the lack of sufficient amount of training and evaluation video data, spanning the aforementioned realistic variations. Earlier work on human action recognition (e.g., [13, 157, 11, 31, 121]) is evaluated on simple video data (e.g., KTH-actions [157] and Weizmann [11] datasets). Such datasets are mainly shot with static cameras, having simple and homogeneous backgrounds, and humans fully visible. Recently, more realistic datasets have been introduced (e.g., Hollywood-2 [111], UCF-sports [148], YouTube-actions [101], etc.). These contain labeled video sequences from movies, sports broadcasts or web videos. While these datasets address common variations in realistic scenarios, they are still limited in the number of training and test sequences. More recently, there have been attempts (e.g., [82]) to address such shortcomings.

A related issue is the labeling of video sequences. Several automatic approaches have been proposed in the literature. Such approaches rely on web image search results [68], video subtitles [59], and subtitle to movie script matching [25, 32, 91]. Nonetheless, often manual verification is required. Moreover, performance of an action might be perceived differently. For instance, a small-scale experiment shows significant disagreement between human labeling and the assumed ground-truth on a common dataset [131].

1.3 Main contributions

The contributions of this thesis can be categorized into two parts. The first part investigates several methods which represent local information in video, under a common evaluation framework. The second part is concerned with developing new features and

integrating additional supervision into Bag-of-Features based representations. These contributions are summarized below.

- We perform a systematic evaluation and comparison of several of the available local space-time features and descriptors under a common Bag-of-Features based action recognition framework. In total, we investigate four different feature detectors and six feature descriptors on a total of 25 action classes distributed over three datasets with varying difficulty. This work provides a comprehensive evaluation and comparison of the popular local space-time features and descriptors.

- We propose an improvement in the standard Bag-of-Features representation using non-local region level information in video. We integrate additional supervision with the Bag-of-Features representation by utilizing pre-trained region detectors. We furthermore investigate combination of different complementary video representations in a kernel combination framework and demonstrate promising results on a challenging dataset.

- We investigate an attribute-based approach to integrate high-level information with Bag-of-Features representation. The proposed Attribute Bank representation is capable of detecting characteristic attributes (e.g., objects, static actions, and poses) in video, and provides complementary high-level information to the low-level features. The Attribute Bank representation is based on pre-trained detectors, which have been trained on large number of static images. Empirical evaluation demonstrates the promise of the proposed method.

- We propose *Actlets*, a novel approach to represent discriminative local motion patterns in video. To train such specialized detectors, we create a relatively large synthetic dataset of avatars, performing different human actions. We then devise a method which successfully utilizes Actlets for human action recognition in video, and demonstrate promising results on two challenging datasets.

1.4 Outline

The presentation of the rest of this thesis is organized in Chapters 2 – 7. The content of these chapters is summarized below.

Chapter 2: Literature review: Several methods and benchmark datasets for human action recognition have been proposed in the computer vision literature, over the past

few years. This chapter reviews related work in human action recognition, and presents several benchmark datasets used in this thesis for performance evaluation.

Chapter 3: Evaluation of local space-time features: Local space-time features have become a popular video representation for action recognition, over the last decade. Several methods for feature localization and description have been proposed, with promising results demonstrated on different human actions datasets. Their comparison, however, is limited, owing to the different experimental settings and various recognition frameworks employed. This chapter presents a systematic evaluation of several recent local space-time feature detectors and descriptors under a common evaluation framework.

Chapter 4: Bag-of-Features with non-local cues: A major factor which limits the performance of Bag-of-Features based video representations is the inherently limited discriminative power of local features. This chapter proposes to improve the basic Bag-of-Features representation by exploiting non-local region-level information and by integrating additional supervision.

Chapter 5: Attribute Bank for action recognition: This chapter investigates an attribute-based representation for human action recognition in video. The proposed Attribute Bank representation employs simple object detectors as well as discriminative human pose and action detectors. Such video representation is shown to capture high-level information in video, which offers complementary information to low-level features.

Chapter 6: Actlets: action-characteristic local motion descriptors: This chapter introduces a novel video representation based on Actlets. Actlets are discriminatively-trained detectors of human body parts undergoing specific patterns of motion. The chapter first demonstrates how to train Actlets from a large pool of automatically annotated synthetic videos, derived from the motion-capture data. It then presents a method which successfully employs Actlets for human action recognition in video.

Chapter 7: Conclusion and future perspectives: This chapter concludes the thesis with a discussion. It also sheds light upon some future perspectives.

1.5 Publications

This thesis is partly based on the following publications:

- H. Wang, M. M. Ullah, A. Klaŝer, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. British Machine Vision Conference (BMVC)*, UK, 2009.

- M. M. Ullah, S. N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *Proc. British Machine Vision Conference (BMVC)*, UK, 2010.

- M. M. Ullah and I. Laptev. Actlets: A novel local representation for human action recognition in video. In *Proc. IEEE International Conference on Image Processing (ICIP)*, USA, 2012.

Chapter 2

Literature review

Contents

This chapter reviews related work in human action recognition. It starts by presenting an overview of the visual classification problem in Section 2.1. Related work in object recognition is briefly discussed in Section 2.2. Section 2.3 then reviews some state-of-the-art methods of human action recognition presented in the literature. Finally, Section 2.4 presents some standard benchmark datasets, used to evaluate and compare different action recognition techniques in the literature and in this thesis.

2.1 Image and video classification

Classification, in literal terms, is the act or process of dividing things into groups according to their type[1]. In computer vision, visual classification is the process of dividing images or videos into *semantic* categories, typically based on their visual content. Consider, for instance, all the images belonging to the class 'vehicle'. The vehicle images can be further divided into sub-classes: bicycle, motorbike, car, truck, bus, tank, aeroplane, and so on. All of these are semantically well-defined categories, as each corresponds to a different object having specific structure as well as appearance. Now suppose, we want to build a visual classification system able to discriminate between the different types of vehicle images. A typical way is to proceed by collecting a representative set of training images for each category. Next, discriminative features (e.g., based on shape, color, texture, etc.) are computed from the training images corresponding to each category. Following that, a machine learning technique is employed to learn a model on the features extracted for each category. During the test phase, the learned model is expected to accurately classify a novel vehicle image, which has not been used during the training phase. Likewise for video, the objective is to build an action classification system, able to differentiate between different actions performed in videos (e.g., running, fighting, walking, kissing, etc.).

However, not as straightforward as it sounds, visual classification is an open, highly challenging and active research area as demonstrated by performance evaluations, e.g., in PASCAL VOC [37] and TRECVID [162] competitions. The visual content in images as well as videos is greatly affected by many factors. For instance, view changes, background clutter, occlusion, and illumination conditions are the primary sources of variations in the visual appearance. Moreover, another issue is the large *intra-class* variation in certain classes. For example, there is a diverse range of model styles available within the visual class 'cell phone'. Furthermore, the additional *temporal* dimension in the video domain poses additional challenges for visual action classification. For instance, a variety of action styles and speeds, background motion, camera shake and motion, etc., make the visual classification of actions in videos even harder.

Classification problem from the statistical point of view is described in Appendix A, wherein, we briefly present the Support Vector Machines classifier. We explain the Bag-of-Features based visual classification in Section 2.1.1.

[1]Cambridge dictionaries online: `http://dictionary.cambridge.org`

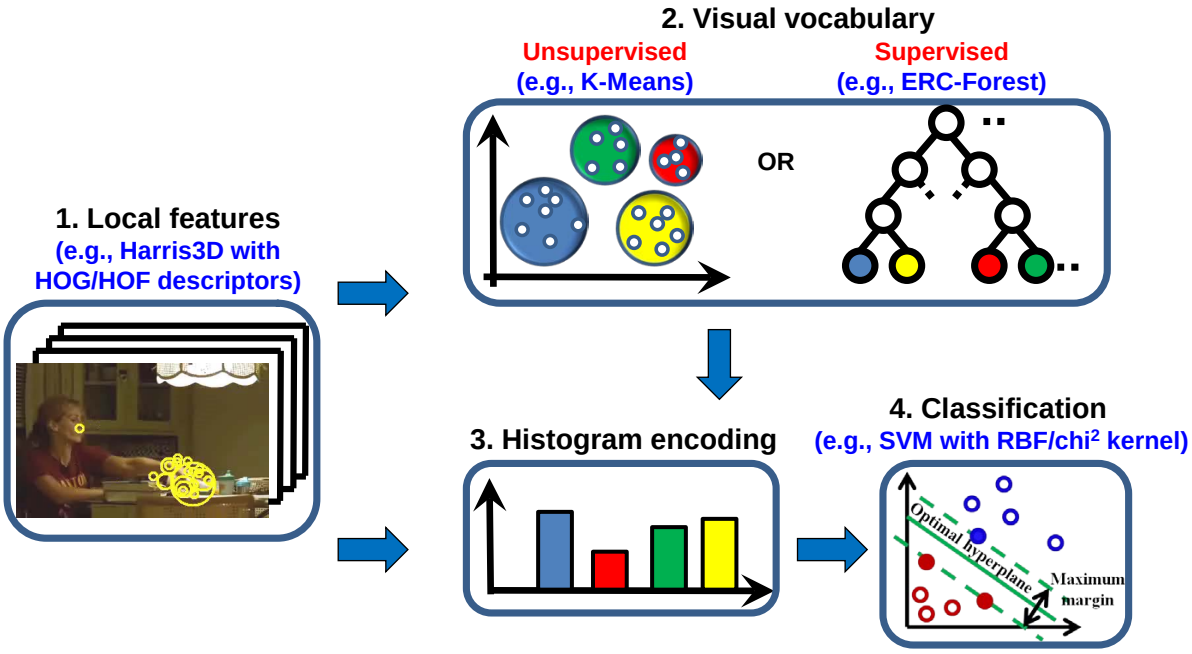


Figure 2.1: **Illustration of Bag-of-Features (BoF) classification**. Refer to the text for further details about each step of the pipeline.

2.1.1 Bag-of-Features classification

Bag-of-Features (BoF) has been a popular visual representation over the last decade. Historically, BoF is inspired by the success of Bag-of-Words (BoW) representation in text retrieval systems. The basic idea behind the BoW model is to describe textual documents as occurrence frequency distributions over discriminative words. This representation has been extensively applied in text retrieval domain [153].

In the field of computer vision, [28], [161], [27], [160] are among the first to extend the BoW model to BoF with applications for texture classification, object/scene retrieval, image categorization, and object localization, respectively. Whereas, [157], [31], and [121] propose the first extensions to action recognition in video. Consequently, *words* are replaced by *visual words* or *features* in the proposed BoF model.

A typical BoF based visual classification is comprised of the following main steps, each of which is schematically shown in Figure 2.1:

Local features

The first step is to compute local features in an image or a video. Local features describe the visual observation at characteristic local regions or patches, and comprised of the following two steps: (a) feature extraction, and (b) feature description. Feature extraction

is the process of detecting interest points (or keypoints) in the input image or video, such that the same points can be detected again even under different transformations (e.g., scale/view changes, rotations, etc.). In case of an image, common types of interest points include blobs, corners, and edges (e.g., Harris-Laplace [115], Difference-of-Gaussian [106, 105], etc.). Whereas, in a video, the space-time interest points (i.e., STIPs) are the locations in space-time where sudden changes in movement occur (e.g., Harris-3D detector [88]). Feature description then summarizes an image or a video patch in a vector representation that is ideally invariant to background clutter, appearance and occlusions, and possibly to rotation and scale. SIFT [106, 105] is a common descriptor for images, whereas, HOG/HOF descriptor [91] is typically used to represent STIPs in videos. Figure 2.1 illustrates an example of detected STIPs.

Visual vocabulary construction

Once feature extraction/description has been done on the training and test set, the next step is to learn a visual vocabulary. The rationale behind learning a visual vocabulary is to be able to give a compact and discriminative representation to an image or a video sequence, which can be efficiently used in the subsequent training and classification stages. The typical idea is to partition the local descriptor space into informative regions, whose internal structure can be disregarded or parameterized linearly. These regions are also called *visual words*, and a collection of visual words is called a *visual vocabulary* (or codebook). Here, we discuss two approaches to construct a visual vocabulary.

k-means is probably the most common way of constructing *unsupervised* visual vocabularies. Given a set $x_1, \ldots, x_n \in \Re^N$ of n training descriptors, k-means seeks K vectors $\mu_1, \ldots, \mu_K \in \Re^N$ and data-to-means assignments $q_1, \ldots, q_n \in \{1, \ldots, K\}$ such that the cumulative approximation error $\sum_{i=1}^{n} \|x_i - \mu q_i\|^2$ is minimized. We consider the standard Lloyd's algorithm [104] for k-means clustering, which is an optimization method that alternates between seeking the best means given the assignments ($\mu_k = \text{avg}\{x_i : q_i = k\}$), and seeking then the best assignments given the means:

$$q_{ki} = \text{argmin}_k \|x_i - \mu_k\|^2 \tag{2.1}$$

Extremely randomized clustering forest (ERC-Forest), in contrast to k-means, is a *supervised* approach to clustering [119], which has been previously employed for image classification tasks [122, 96]. ERC-Forest is an ensemble of randomly created clustering trees. It predicts class labels y from local feature descriptors x. It benefits from labeled training set $J = \{(x_i, y_i)\}_{i=1}^{n}$ with n descriptors x associated with class

labels y and recursively builds random trees in a top-down manner. At each node, the labeled training set is divided into two halves such that the classes are separated well by maximizing the Shannon entropy:

$$S_c\left(J,T\right) = \frac{2 \cdot I_{C,T}\left(J\right)}{H_C\left(J\right) + H_T\left(J\right)} \ , \tag{2.2}$$

where H_C denotes the entropy of the class distribution in J, H_T is the split entropy of the test T which splits the data into two partitions, and $I_{C,T}$ is the mutual information of the split. Let the ERC-Forest consists of M random trees, each of K leaf nodes, which are treated as visual words. During quantization, each local descriptor x_i traverses each tree from the root down to a leaf. Each tree assigns a unique leaf index to the visual descriptor. As a result, for each descriptor x_i, the ERC-Forest returns $M \times q_{ki}$ leaf indices, one for each tree, corresponding to the associated visual word (see [119] for further details).

Histogram encoding

Given a visual vocabulary, an image or a video can be represented by local features assigned to visual words. A conventional approach is the histogram encoding, introduced in [27, 95, 161]. As the name suggests, histogram encoding is a histogram of the quantized local descriptors. Given a set of descriptors $x_1, \ldots, x_n$, let q_{ki} be the assignments of each descriptor x_i to the corresponding visual word, as given by Eq. 2.1. The histogram encoding of the set of local descriptors is the non-negative vector $H \in \Re^K$, such that $[H]_k = |\{i : q_{ki} = k\}|$. In the case of ERC-Forest, the histogram encoding is the non-negative vector $H^{'} \in \Re^{M \times K}$, such that $[H^{'}] = [H_1 \ldots H_M]$, corresponding to each random tree. Irrespective of the type of clustering involved, such histograms only contain global statistics about the type of descriptors found in an image or a video sequence. Any information about the spatial or temporal relations between the descriptors is ignored.

As described above, histogram encoding computes a histogram of visual words. Recently, several approaches are proposed to improve the histogram encoding by replacing the hard quantization of descriptors involved with alternative encodings that retain more information about the original descriptors. This has been achieved either by expressing descriptors as *combinations* of visual words (e.g., soft quantization [173], local linear encoding [177]), or by recording the *difference* between the descriptors and the visual words (e.g., Fisher encoding [132], super-vector encoding [195]). We refer the reader to [24] for a comprehensive evaluation of recent encoding methods.

Classification

The final step is classification, which involves training a classifier on the labeled training histograms, and subsequent classification of the test histograms. A non-linear SVM with χ^2 kernel is a frequent choice of a classifier, that has been employed in different state-of-the-art methods [157, 31, 91, 184, 171].

2.2 Object recognition

In this section, we briefly review related state-of-the-art work in object recognition. The two standard tasks in object recognition are (a) image classification, and (b) object detection. The goal in classification is to identify the presence of an object (e.g., face, person, horse, aeroplane, etc.) in an image, and to classify it to one of the known object categories. Object detection, on the other hand, localizes its position in the image, and possibly estimates its pose as well.

A popular way of object classification is bag-of-features classification (e.g., [27, 127, 193]). First, local patches are extracted from all the training images and quantized into a visual vocabulary. Each image is then represented by a histogram, indicating the number of occurrences of each visual word. A classifier is then trained to predict the presence/absence of an object in novel images, which are also described by histograms of visual word occurrences. The main advantage of bag-of-features approach is its simplicity and the relatively small amount of supervision involved. Labelling the training data only requires indicating the presence/absence of an object in the image. No manual object segmentation or bounding box specification is needed.

An extension to bag-of-features classification is the *Object Bank* representation by Li *et al.* [98, 99], which has been used for image scene classification. The authors employ a large number of pre-trained generic object detectors (e.g., water, sky, boat, bear, etc.) on an input image at multiple scales. The resulting response map for each object is max-pooled, and the corresponding maximum response values are concatenated into a vector representation, encoding the image. The Object Bank representation has been shown to capture high-level information from scene images. We use the Object Bank technique in Chapters 5 and 6.

A usual way of object detection is the *sliding window* approach. This approach involves training a classifier, which for a fixed size image patch, decides whether the desired object (e.g., a face) is present. Given a test image, such a classifier is then applied within a

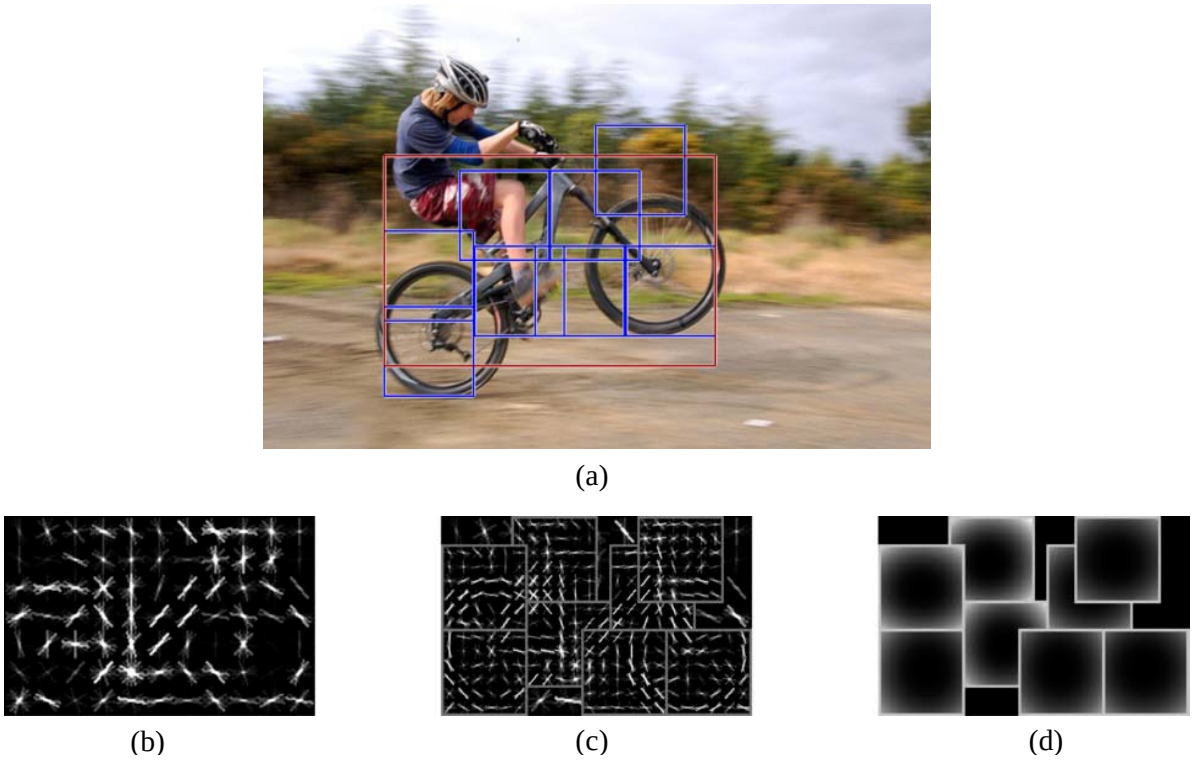


Figure 2.2: **Illustration of discriminatively trained star-structured part-based model**. (a) Detections obtained with a single component bicycle model; (b) the model is defined by a coarse root filter; (c) several higher resolution part filters; and (d) spatial model for the location of each part relative to the root (figure reprinted from [40]).

sliding window, over a range of translations and scales. Training the classifier typically requires many cropped training images, with both object present and absent. The task of the classifier is to capture the intra-class variations in the training object instances.

An example of a sliding window approach is the outstanding work by Viola and Jones [174]. The authors propose a very fast frontal face detector. The features are based on sums of pixel values in rectangular image regions, which can be computed very efficiently using an integral image. The authors employ a cascade of detectors with increasing complexity, where only image windows likely to contain faces, are passed to more complex classifiers further down the cascade. Each stage of the cascade is a classifier, which is trained using AdaBoost.

Deformable part models based on *pictorial structures* [34] have also been investigated for object detection (e.g., [42, 41, 142]). Pictorial structures represent objects by a collection of parts, arranged in a deformable configuration. Each part captures local appearance properties of an object, while the deformable configuration is characterized by spring-like

connections between certain pairs of parts. In contrast to bag-of-features based models, deformable part models can localize objects in images. Recently, Felzenszwalb *et al.* [40] introduce a state-of-the-art object detection method based on mixtures of deformable part models (see Figure 2.2). These models are trained using a discriminative method that only requires bounding boxes for the objects in an image. The main features of their approach are: (i) strong low-level features based on histograms of oriented gradients (HOG), (ii) efficient matching algorithms for deformable part-based models, and (iii) discriminative learning with latent variables (latent SVM). The approach leads to efficient object detectors that achieve state-of-the-art results on PASCAL VOC 2007 and 2009 challenges. We use the object detection scheme in Chapters 4 and 5.

Moreover, Bourdev and Malik [17] recently propose a novel approach to body part localization, called *poselets*. Poselets are body-part detectors, trained on a relatively large amount of annotated static images, and invariant to distracting variations in still images. Poselets achieve state-of-the-art results on PASCAL VOC 2007-2010 challenges for the person category. We employ poselets in Chapter 5.

2.3 Human action recognition

Vision-based human action recognition, in broader sense, can be regarded as a combination of feature extraction/representation and subsequent classification of image representations. Consequently, vision-based techniques for human action recognition can be categorized according to many different criteria. For instance, according to the body parts involved (facial expressions, hand gestures, leg movements, upper-body gestures, full-body motions, etc.); the extracted image features (landmarks, edges, silhouettes, optical flow, interest points, trajectories, etc.); and the class of statistical models used for learning and recognition (Nearest Neighbors, Support Vector Machines, Markov Models, Bayesian Networks, Conditional Random Fields, etc.) [182]. As the scope of this thesis is feature representation in action recognition, we classify the existing methods based on the type of features used to model and recognize human actions. In this regard, existing methods of human action recognition are categorized into the following three main classes:

- *Body landmark based methods* represent structure of actions by employing positions as well as movements of landmark points on the human body. For example, body-joints can serve as landmark points. This class of methods is briefly presented in Section 2.3.1.

- *Holistic appearance and motion based methods*, in contrast to body landmark based methods, localize humans in video. An action model is subsequently learnt, which captures characteristic holistic body shape and/or motion, irrespective of any notion of body parts or landmark points. These methods are reviewed in Section 2.3.2.

- *Local patch based methods* describe the visual observation of human actions as a collection of independent video patches, without any prior knowledge about human position as well as his/her body part localization. Such methods are detailed in Section 2.3.3.

Several surveys within the area of vision-based human motion analysis and recognition exist in the literature. Early surveys on human motion analysis include [22, 3, 47]. Moeslund *et al.* [117, 118] survey vision-based methods for human motion capture and analysis. Hu *et al.* [67] review action recognition in the context of visual surveillance. Surveys by Forsyth *et al.* [44] and Poppe [138] focus on the recovery of human poses and motion from image sequence. Surveys on human or pedestrian detection (e.g., [46, 35, 50]) are also related, where the task is to localize persons within an image sequence. Broader surveys covering the aforementioned topics, including human action recognition, include [12, 178, 4]. Krüger *et al.*[81] highlight the importance of context in visual action recognition, whereas, Turaga *et al.* [170] focus on the higher-level recognition of human activity. Surveys that exclusively target vision-based human action recognition, are presented by Weinland *et al.* [182] and Poppe [137].

2.3.1 Body landmark based methods

In this section, we review methods which represent actions by modeling the human body. Usually, certain landmarks on human body are used to estimate pose in each frame of the observed video stream. Consequently, an action is represented with the help of the recovered poses. This is an intuitive approach to action recognition, which is also supported by psychophysical work on visual interpretation of biological motion [72].

The classic experiment by Johansson [72] shows that humans can recognize actions merely from the motion of a few moving light displays (MLDs) attached to the human body (Figure 2.3). MLDs consist of bright spots attached to the joints of an actor dressed in black, and moving in front of a dark background. The collection of spots carry only 2D information and no structural information, as they are not connected to each other. While a set of static spots remain meaningless to observers, their relative movement

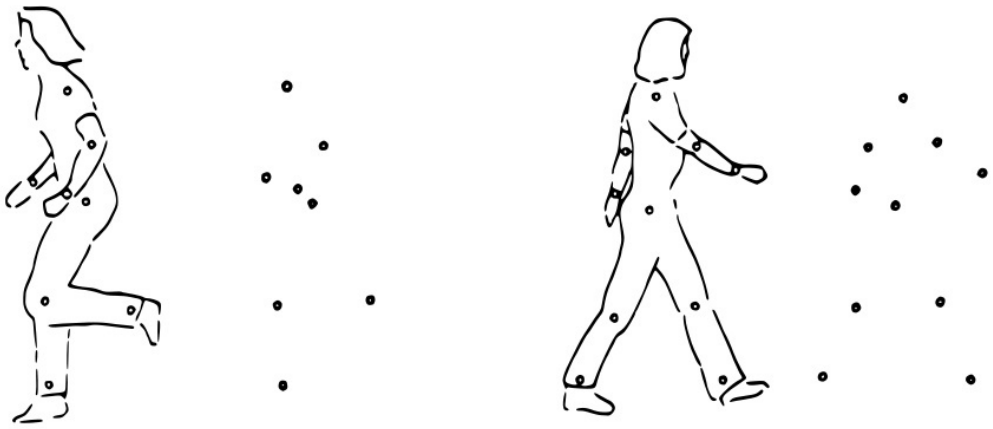


Figure 2.3: **Illustration of Johansson's moving light displays (MLDs) experiment**. Example movements that can easily be recognized by humans with only a few MLDs attached to the human body (figure reprinted from [72]).

create a vivid impression of a person walking, running, and dancing etc. The gender of a person, and even the gait of a friend can be recognized based solely on the movement of these spots [7]. Our easy interpretation of MLDs would indicate that we can directly use body landmark movements as a means for action recognition. Nonetheless, it has been shown that the inverted (upside-down) recordings of MLDs are usually not recognized by humans, even for some simple movements [166]. This would suggest that humans have a strong prior model in their perception [166, 54], i.e., an inverted movement is not natural nor familiar; humans expect people walking upright and can not easily adapt to strong transformations.

Over several decades, Johansson's findings inspired many techniques in human action recognition. Generally, two approaches about the interpretation of MLDs type stimuli, have been advocated in the literature [118]. In the first, relative motion information in the MLDs is used to recover the 3D structure of human body, which is subsequently used for action recognition (*recognition by reconstruction*). In the second approach, the 2D motion information is directly used to perform recognition, without any 3D structure recovery (*direct recognition*).

Recognition by reconstruction first estimates a 3D model of the human body, typically represented as a kinematic joint model, from the 2D motion information. Then, action recognition is performed based on 3D joint trajectories. Two major difficulties, however, are the large number of degrees-of-freedom of the human body and the high variability of their shapes. Consequently, a parametric model of the human body must be

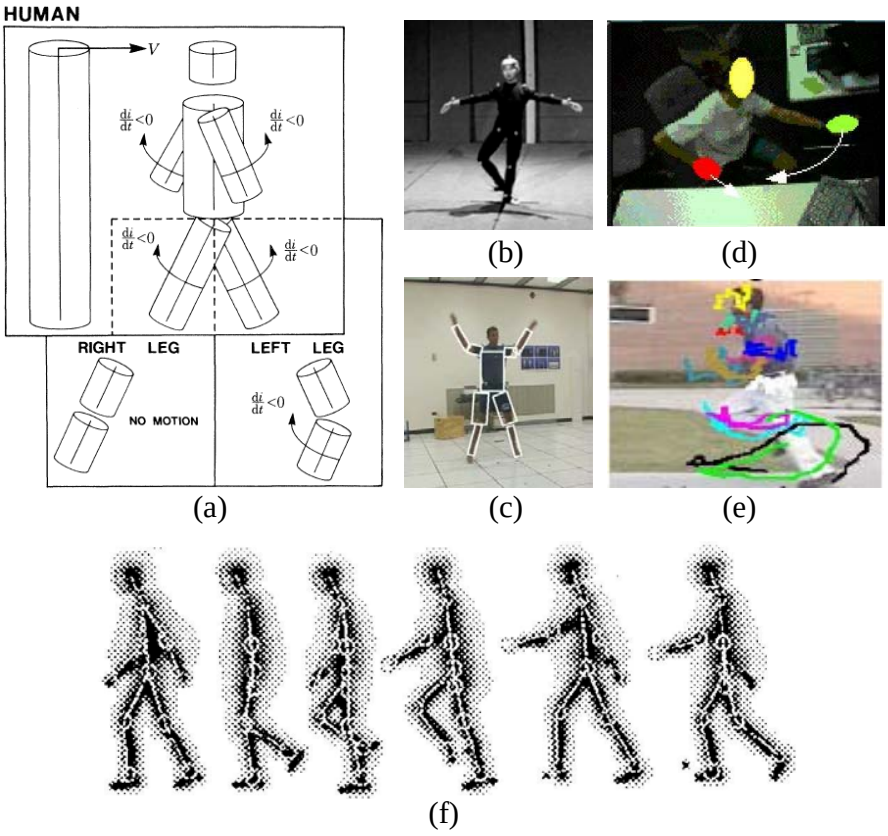


Figure 2.4: **Illustration of body landmark based models**. (a) Hierarchical 3D model based on cylindrical primitives [110]; (b) ballet dancer with special markers attached to the body [21]; (c) body model based on rectangular patches [143]; (d) blob model [18]; (e) 2D trajectories of landmark points [189]; (f) stick figure model [58] (figures reprinted from the respective papers).

carefully selected and calibrated to support a wide range of variations in action styles as well as physiques. A large variety of parametric models have been proposed in the literature (see Figure 2.4 for some examples). Marr and Nishihara [110] propose a theoretical body model consisting of a hierarchy of cylindrical primitives (see Figure 2.4 (a)). Such a model is later adopted in several methods to recognize human movements, e.g., [66, 149]. Gavrila and Davis [48] propose a more general body model based on super-quadrics, in a multi-view approach. Green and Guan [57] propose an even more flexible model by approximating body parts in 3D through a textured spline model. A bottom-up approach is used in [143], which first tracks body parts in 2D, using rectangular appearance patches, and then lifts the tracked 2D configuration into 3D (see Figure 2.4 (c)). Motion capture (MOCAP) techniques which require special markers attached to the human body, have also been used for action recognition. For instance, Campbell and Bobick [21] compute a

joint model from 14 marker points attached to a ballet dancer's body (see Figure 2.4 (b)).

Direct recognition approaches deal with the direct use of 2D motion information, as our easy interpretation of MLDs would suggest. Typically, these methods work from 2D models of the human body, i.e., labeled body parts, without lifting them into 3D. Common 2D representation are stick figures and 2D anatomical landmarks, similar to Johansson's MLDs. For instance, Goddard [55] investigates the use of MLDs for human action recognition. Similarly, Yilmaz and Shah [189] employ the 2D trajectories of landmark points on the human body, to recognize actions under camera movement as well as view-point change (see Figure 2.4 (e)). Guo *et al.*[58] recover a 2D stick figure from the skeleton of a person's silhouette (see Figure 2.4 (f)), whereas, Niyogi and Adelson [123] detect a stick figure from the space-time volume spanned by an image sequence of a walking person. Other direct recognition approaches employ coarse 2D body representations based on blobs and patches. For instance, Starner and Pentland [164] detect the hands of a person facing the camera using skin tone based color segmentation, and track them over time, for American sign language recognition. Moreover, Brand *et al.*[18] use the head and hand trajectories for action recognition in a hidden Markov models (HMM) framework (see Figure 2.4 (d)).

Notwithstanding the most intuitive and biologically plausible approach to action recognition, body landmark based methods are often limited in their applicability to real-world scenarios, owing to many factors. Estimating a 3D parametric body model from an image sequence is a hard problem in itself, and is sensitive to noise. Multiple cues like motion, specularities, textures, etc. are needed. Moreover, 3D reconstruction alone is not sufficient for robust and accurate recognition of actions. On the other side, localization of body parts is a challenging task in realistic and less constrained video data due to background clutter, occlusion, multiple movements, and lighting conditions etc. Some methods (e.g., [2, 150, 163, 172]) achieve relatively better results by using strong prior models assuming particular types of movements (e.g., walking, running, etc.), and thus impose strong constraints on the type of possible body configuration. Such restriction, however, reduces the search space of possible pose estimates, which limits their application to action recognition [133]. Pose estimation from RGB images and video is still a very hard and active research area (e.g., [159, 188, 180, 73, 154]).

2.3.2 Holistic appearance and motion based methods

In this section, we review methods which represent videos by their global appearance and/or motion, instead of relying on the detection and labeling of individual body parts. Holistic representations are obtained in a top-down fashion, wherein a person is localized first in the image using methods of e.g., background subtraction, person detection, tracking or their combinations. Then, the region of interest (ROI) around the person is encoded as a whole, which results in the image descriptor. Holistic representations are in general much simpler compared to representations based on parametric body models or information about body parts. As a result, holistic representations can be computed more efficiently and robustly.

Holistic methods can be roughly classified into three main categories. The first category employs the silhouette information or contours of the person performing the action. The second category is based on the computation of optical flow or gradient in an image sequence. Finally, the third category combines techniques from the first two categories.

Silhouette based methods

These methods represent actions with the help of silhouette information in a video sequence. The silhouette of a person in an image sequence can be obtained by using background subtraction. One of the earliest methods employing silhouettes is by Yamato *et al.* [187] (see Figure 2.5 (a)). The authors divide the extracted silhouette into a regular grid. For each cell, they compute the ratio of black and white pixels within the underlying cell region, as features. These features are used to learn a visual vocabulary, and the quantized tennis actions are subsequently learned using HMMs.

Bobick and Davis [13] integrate silhouettes over time in so-called *motion energy images* (MEI) and *motion history images* (MHI), as illustrated in Figure 2.5 (b). MEI is a binary mask which indicates regions where motion occurs, whereas, MHI represents these regions as a recency function over time (the more recent, the higher the pixel intensity). Two templates are then compared using Hu moments. Their method is the first to introduce the idea of *temporal templates* for human action recognition.

A 3D space-time volume (STV) can be formed by stacking multiple silhouette images. Blank *et al.* [11] and Gorelick *et al.* [56] stack silhouettes over a given sequence to form an STV (see Figure 2.5 (c)). Then, the solution of the Poisson equation is used to derive local space-time saliency and orientation features. Global features for a given temporal range (i.e., 10 frames) are obtained by calculating weighted moments over these local

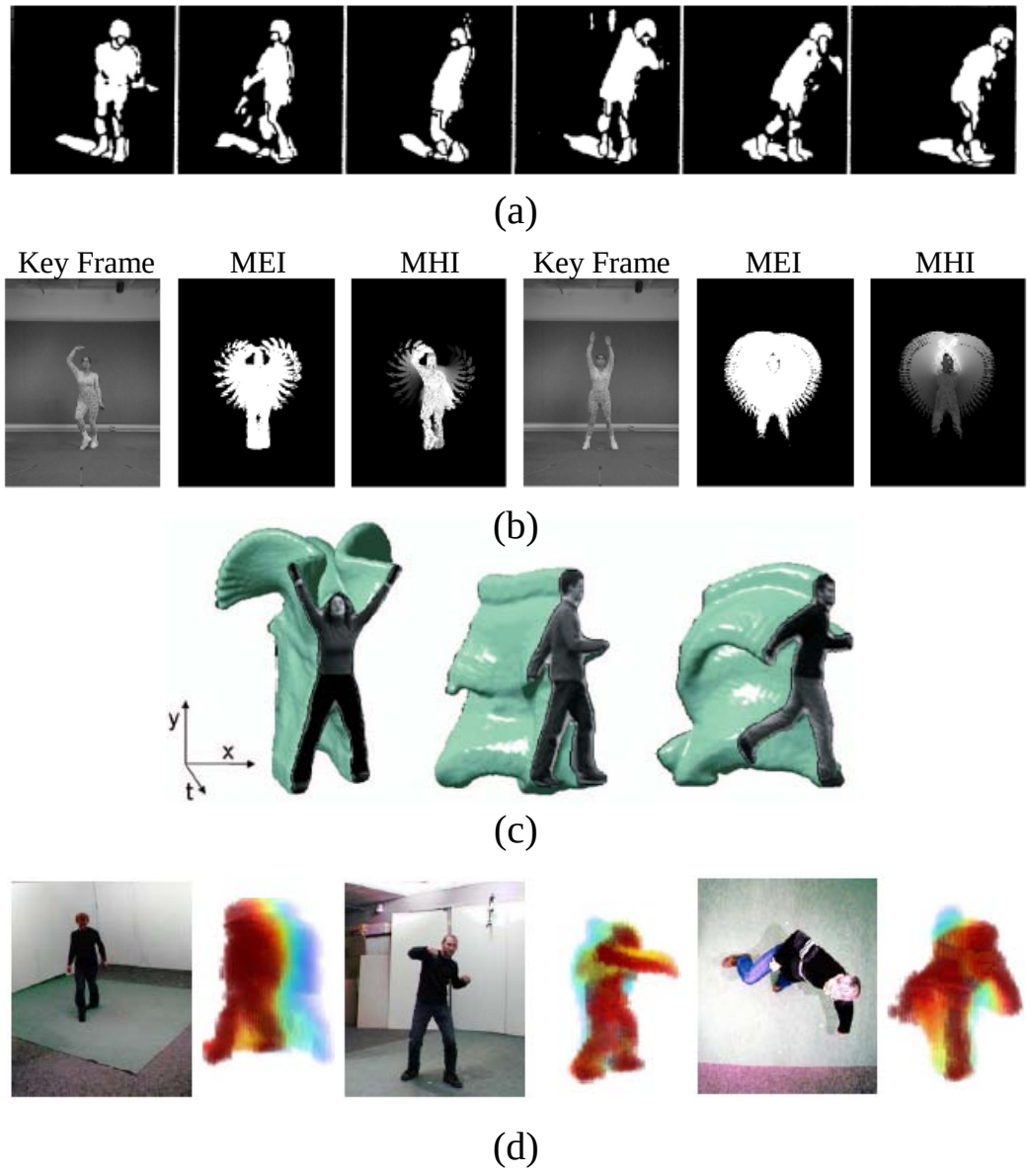


Figure 2.5: **Illustration of silhouette based representations**. (a) Silhouette shape masks for representing tennis actions [187]; (b) silhouette based motion energy images (MEI) and motion history images (MHI) [13]; (c) space-time volumes (STV) [11]; (d) motion history volumes (MHV) [191] (figures reprinted from the respective papers).

features, and represented in a high-dimensional feature vector. During classification, these feature vectors are matched in a sliding window fashion to STVs in the test sequences. Later, Achard *et al.* [1] propose to use a set of STVs for each video sequence, each of which covers only a part of the temporal dimension. Their approach, therefore, helps to deal with action performances of different temporal durations.

When multiple cameras are employed, silhouettes can be obtained from each. Weinland *et al.* [191] combine silhouettes from multiple cameras into a 3D voxel model. They use *motion history volumes* (MHV), which is an extension of the MHI [13] to 3D (see Figure 2.5 (d)). Such a representation is informative enough but requires accurate camera calibration. View-invariant matching is performed by aligning the MHV using Fourier transforms on the cylindrical coordinate system around the medial axis. Even though the representations of STV [11] and MHV appear similar (see Figure 2.5), the former is viewed from a single camera, whereas, the latter is viewed from multiple cameras and shows a recency function over reconstructed 3D voxel models.

Weinland and Boyer [181] propose an orderless representation for action recognition based on a set of silhouette exemplars. The authors represent a video sequence with a vector of minimum distances between silhouettes in the set of exemplars and in the sequence. Classification is then performed using Bayes classifier with Gaussians to model action classes. Moreover, the authors employ the Chamfer distance measure to match the silhouette exemplars directly to edge information in the test sequences, thereby eliminating the need for background subtraction.

Ragheb *et al.* [140] propose to transform an STV to Fourier domain. The authors first compute an STV (similar to [11]) for a given video sequence. Then, each STV is divided into space-time sub-volumes (STSV), wherein the corresponding mean frequency responses are used as a feature vector. Classification is based on a weighted Euclidean distance measure, where the representation is shown to cope with camera view changes as well as silhouette imperfection and noise.

Silhouettes provide strong cues for action recognition, and are insensitive to color, texture, and contrast changes. Nonetheless, reliable person segmentation in realistic settings is still a very challenging problem due to failures of background subtraction, occlusions, unreliable person detection and tracking. Silhouettes of the person are also not capable of capturing certain actions generating signal on the interior of the person, e.g., drinking for frontal person views.

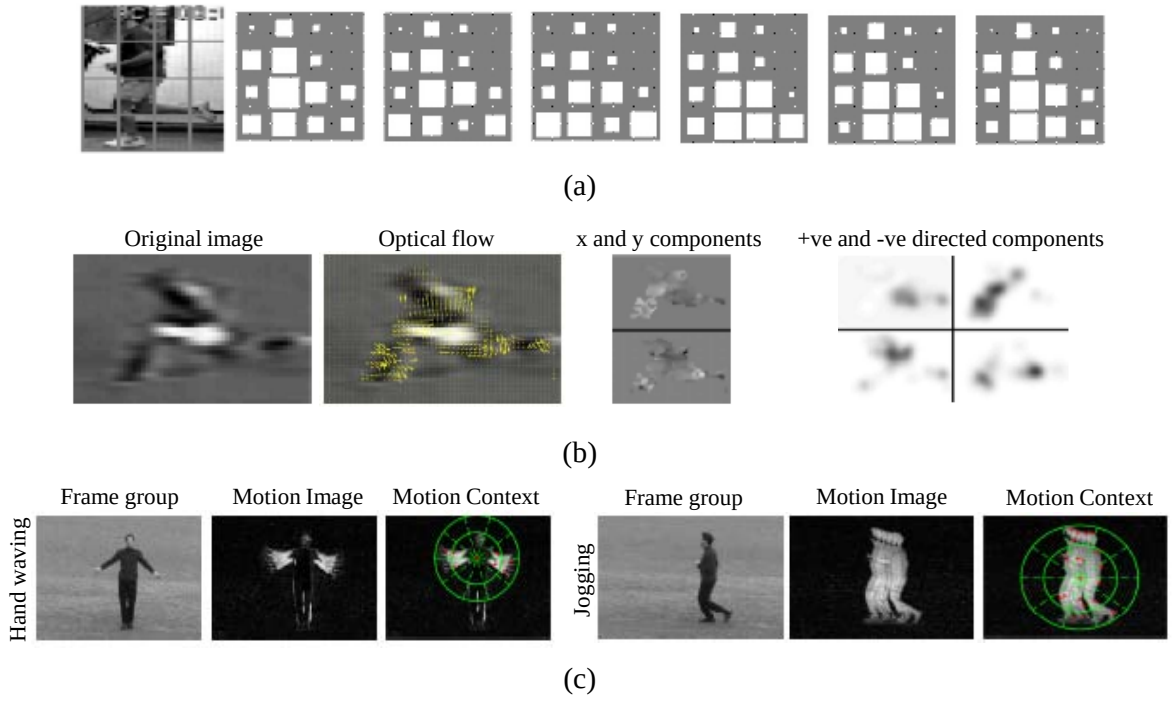


Figure 2.6: **Illustration of motion based methods**. (a) A human-centered grid of optical flow magnitudes to describe actions [136]; (b) motion descriptor using optical flow [33]; (c) motion images are computed over groups of images; the Motion Context descriptor is computed over consistent regions of motion [194] (figures reprinted from the respective papers).

Optical flow and gradient based methods

The observation within the ROI can be represented with motion information and/or gradient information. A substantial body of research in action recognition is based on optical flow, which measures pixel-wise displacements in the image plane. Optical flow can be used when background subtraction cannot be performed. Polana and Nelson [135] are one of the first to use optical flow for motion recognition. They propose to use *temporal-textures*, i.e., first and second order statistics based on the direction and magnitude of normal flow, to recognize events such as motion of trees in wind or turbulent motion of water. Later, Polana and Nelson [136] propose to use optical flow for human action recognition. They first track the person to get the ROI. Then, optical flow is computed, and the flow magnitudes are accumulated in a regular spatio-temporal grid of non-overlapping bins (see Figure 2.6 (a)). The flow based descriptor is computed for periodic motion patterns (e.g., walking, running, swimming, skiing, etc.). Classification is based on matching the descriptors in test sequences to reference motion templates of

known periodic actions.

Another approach in this direction is presented by Efros *et al.* [33]. They track soccer players in sports footage, where persons in the image are very small, and calculate optical flow in person-centered images. The result is blurred as optical flow can result in noisy displacement vectors. To make sure that oppositely directed vectors do not cancel out, the horizontal and vertical components are divided into positively and negatively directed, yielding four distinct channels (see Figure 2.6 (b)). Classification is then performed by frame-wise aligning a test sequence to a database of annotated actions, and matching the four channels separately. The proposed representation is later used in [147, 179], whereas, Ahad *et al.* [5] use the four flow channels to solve the problem of self-occlusion in a MHI approach.

Fablet and Bouthemy [38] propose a probabilistic approach to design nonparametric motion models for characterizing motion content within image sequences. The proposed temporal multi scale Gibbs models, computed from co-occurrence statistics of optical flow based measurements, are shown to capture both spatial and temporal aspects of the underlying motion. Recognition results on a wide variety of dynamic contents (e.g., wind blown grass, gentle sea waves, moving escalator, person walking, etc.) show promise of the nonparametric motion modeling. Later, Piriou *et al.* [134] present a probabilistic framework wherein, camera motion is explicitly modeled using affine motion models. Whereas, low-level local motion features are used to model the scene motion. The approach is successfully demonstrated for the classification of a wide range of sport actions (see [79] for an overview of sports-related indexing and retrieval work).

A somewhat different approach is proposed by Zhang *et al.* [194]. The authors compute foreground shape masks based on motion information in chunks of video data. Then, a motion context descriptor is computed over consistent regions of motion by using a polar grid (see Figure 2.6 (c)). Each cell in the grid is described with a histogram over quantized SIFT descriptors. The final descriptor for a video sequence is the sum over all the chunk descriptors. Classification is performed using SVM as well as different models for probabilistic latent semantic analysis (PLSA).

Rodriguez *et al.* [148] propose to use flow features in a template matching framework. They compute spatio-temporal cubes over regularity flow information. Regularity flow shows improvement over optical flow as it globally minimizes the overall sum of gradients in the image sequence. The cuboid templates are learned by aligning training samples via correlation. For classification, test sequences are correlated with the learned cuboid

templates using generalized Fourier transform, which allows for vectorial values.

Ali and Shah [6] derive a number of kinematic features from the optical flow. These include divergence, vorticity, symmetric and anti-symmetric flow fields, second and third principal invariants of flow gradient and rate of strain tensor, and third principal invariant of rate of rotation tensor. Principal component analysis (PCA) is applied on the spatio-temporal volumes of the kinematic features to determine the dominant kinematic modes. For classification, the authors propose to use multiple instance learning (MIL), in which each action video is represented by a bag of kinematic modes. Each video is then embedded into a kinematic mode based feature space, and the coordinates of the video in that space are used for classification using the nearest neighbor algorithm.

Optical flow based representations do not depend on background subtraction, which makes them more practical than silhouettes in many settings. However, they rely on the assumption that image differences can be explained as a result of movement, rather than changes in dynamic backgrounds, such as changes in material properties, illumination, etc. Also, camera movement results in observed motion, which can be compensated for by tracking the person.

An important class of image features is based on gradient, which is a directional change in the intensity or color of an image. Gradient based representations have gain popularity in particular with local sparse features (see Section 2.3.3). However, there are several approaches which employ gradient globally. Zelnik-Manor and Irani [192] propose to construct the *temporal pyramid* by blurring and sub-sampling a video sequence along the temporal direction only. The temporal pyramid is comprised of three levels, corresponding to three temporal scales. The authors then compute the space-time gradient at each space-time point in each of the three pyramid cubes. Two sequences are matched by comparing gradient measurements across the corresponding pyramid cubes.

A popular gradient based representation is the histogram of oriented gradients (HOG) descriptor, which has been very successfully applied to person and object detection [29]. Lu and Little [107] present a simultaneous tracking and action recognition framework using the *PCA-HOG* descriptor. They track soccer or ice-hockey players and represent each frame by a descriptor using histograms of oriented gradients. PCA is then applied to reduce the descriptor dimensionality. An HMM with a few states is employed to model actions such as running and skating etc.

Thurau and Hlavac [168] extend the HOG descriptor [29] for human action recognition in videos or still images. Instead of computing a single gradient histogram per frame,

the authors divide an image into regularly spaced overlapping blocks, and compute a histogram within each of those blocks. Action classes are then represented by histograms of *poses primitives*. Action recognition is based on the nearest neighbor algorithm.

Gradient based representations share many characteristics with those of optical flow. In particular, they do not depend on background subtraction, but likewise are sensitive to material properties, texture, and lighting, etc. [77]. In contrast to optical flow, gradients are discriminative for both moving and non-moving regions, which is advantageous in certain situations, whereas, disadvantageous in others. For instance, static non-moving body parts can also provide strong cues for an action, yet might be easily confused with still object in the background with strong gradients.

Hybrid methods

Only one type of features may not be able to capture the full dynamics of an action in a video, and thus could result in sub-optimal recognition performance. In order to cope with the discrepancy associated with using only a single type of features, researchers have attempted to combine different types of features, and demonstrated superior performance. Common hybrid representation combine optical flow with gradient (i.e., appearance) information, or silhouettes with optical flow. For instance, Schindler and Gool [156] use optical flow information and Gabor filter responses in a human-centric framework. For each frame, both types of information are weighted and concatenated. PCA is applied over all pixel values to learn the most discriminative feature information. The authors employ a majority voting scheme to yield the final class label for a full video sequence in multi-class experiments. Results are reported on the KTH actions and Weizmann datasets.

In another hybrid approach, Laptev and Perez [92] demonstrate the localization of drinking actions in movies by learning a cuboid classifier that combines a set of appearance (histograms of oriented gradients) and motion features (histograms of optical flow), as illustrated in Figure 2.7. To avoid an exhaustive spatio-temporal search and to improve performance for action localization, the authors propose to pre-filter possible action localizations with a human key-pose detector, trained on keyframes of the action.

Tran and Sorokin [169] propose a metric learning approach to human action recognition. The authors propose to capture local motion and appearance in each frame by combining optical flow with silhouette mask, in a human-centric approach. Moreover, motion context is introduced by appending a summary of the motion (i.e., histograms of optical flow and silhouette) around each frame. The proposed method is capable of rejecting unseen

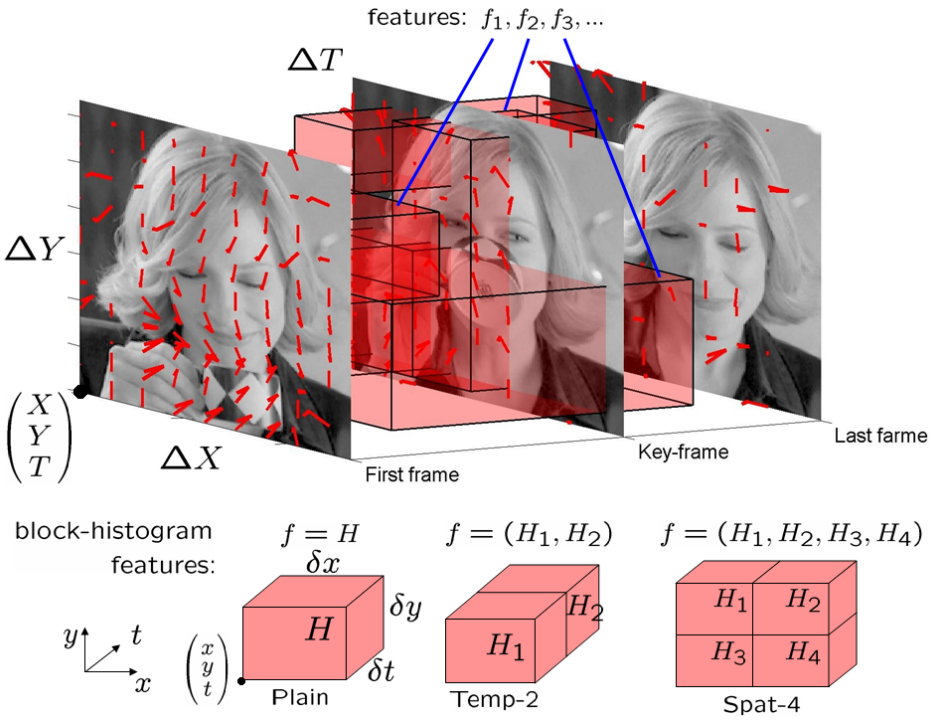


Figure 2.7: **Illustration of a drinking action with different histogram features**. (Top) Action volume in space-time is represented by a set of basic motion and appearance features; (bottom) three types of features with different arrangements of histogram blocks (figure reprinted from [92]).

actions, and can learn from few training instances. The method is shown to perform well on noisy YouTube videos.

Holistic methods rely on assumptions such as tracking, detection, etc., which are hard to meet with current methods in realistic video data. On the other hand, template-based methods might be too rigid and require much annotation to address a wide range of action classes. Such limitations limit the applicability of holistic methods in realistic settings.

2.3.3 Local patch based methods

In this section, we discuss local patch based methods, which describe actions by orderless collections of video patches. Such approach is also referred to as the *Bag-of-Features* representation (introduced in Section 2.1.1). This class of methods proceeds in a bottom-up fashion, wherein space-time interest points are first detected, and local patches around these points are subsequently summarized in descriptor representations. Local patch

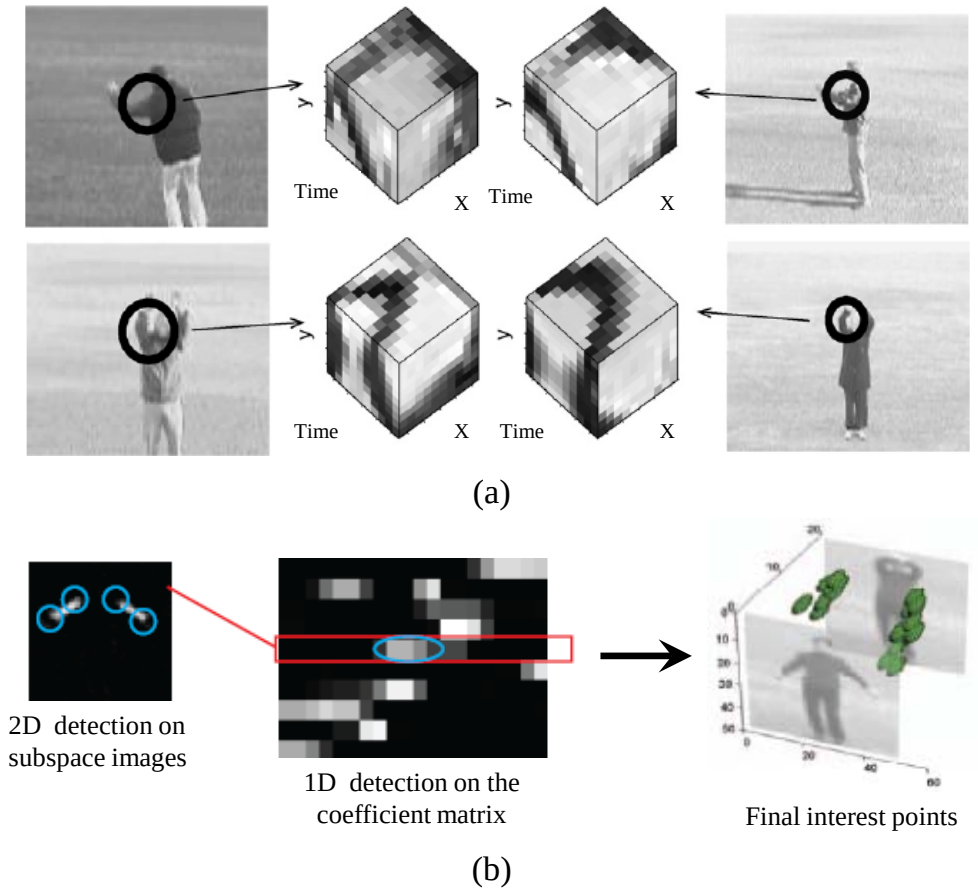


Figure 2.8: **Illustration of space-time interest points (STIPs)**. (a) Extraction of space-time cuboids at interest points from similar actions performed by different persons [89]; (b) detection of STIPs using global information [185] (figures reprinted from the respective papers).

based methods, owing to their local nature, are less sensitive to appearance variations, e.g., partial occlusions, view-point changes, etc. Moreover, local representations are straightforward to compute, and do not require background subtraction nor tracking. Nevertheless, local features are sensitive to severe variations in appearance and motion.

We first review a variety of available local space-time interest point detectors. We then discuss few local descriptors, proposed to describe local patches around space-time interest points. Following that, we briefly review methods based on local feature trajectories. Finally, we discuss few attempts to improve the Bag-of-Features approach by modeling the spatio-temporal relationships among local features.

Space-time interest point detectors

Space-time interest points are the characteristic locations where the local neighborhood

has a significant variation in both the spatial and the temporal domain. In other words, these are the locations in space and time where sudden changes of movement occur in the video. It is assumed that these locations are most discriminative for human action recognition in video. Laptev and Lindberg [90, 88] are the pioneers of introducing a space-time interest point detector based on a 3D spatio-temporal extension of the Harris corner detector [62]. The corness criterion is based on the eigenvalues of a spatio-temporal second-moment matrix at each point in video. Local maxima indicate points of interest. The authors propose to automatically select the scale of the neighborhood for space and time individually, as spatial and temporal extents of actions are in general independent. Later, the work is extended to compensate for relative camera motions in [89]. Figure 2.8 (a) illustrates the detection of Harris3D interest points and associated cuboid patches in some video sequences.

Harris3D [88] detects relatively sparse amount of space-time interest points. However, Dollár *et al.* [31] argue that in certain cases, true spatio-temporal corner points (according to the Harris criterion) are relatively rare, while enough characteristic motion is still present. Therefore, they design their interest point detector to yield relatively denser coverage in videos. Their method employs spatial Gaussian kernels and temporal Gabor filters. Like for Harris3D, local maxima give final interest points. The number of interest points is adjusted by changing the spatial and temporal size of the neighborhood in which local maxima are selected. Rapantzikos *et al.* [144] use the responses after applying discrete wavelet transforms in each of the three directions of a video volume. Responses from low-pass and high-pass filters for each dimension are used to select salient points in space and time. In addition to intensity and motion cues, Rapantzikos *et al.* [145] also incorporate color. They compute saliency as the solution of an energy minimization process which involves proximity, scale, and feature similarity terms.

Oikonomopoulos *et al.* [126] extend the work on 2D salient point detection by Kadir and Brady [75] to 3D space and time. The entropy within a cylindric cuboid around a given space-time position of a video sequence is calculated. The centers of the entropies with local maximum energy are selected as interest points. The scale of each interest point is determined by maximizing the entropy values.

Willems *et al.* [184] propose a 3D spatio-temporal extension of the Hessian saliency measure applied for blob detection in images [9]. The authors attempt to design a rather dense, scale-invariant, and computationally efficient interest point detector. Saliency of interest points is measured using the determinant of the 3D Hessian matrix. An integral video structure allows to speed up computations by approximating derivatives

with box-filter operations. A non-maximum suppression algorithm selects joint extrema over space, time, and different scales. Another attempt to contain the computational complexity is presented by Oshin *et al.* [129]. The authors train randomized ferns to approximate the behavior of interest point detectors.

Instead of determining the saliency of an interest point with respect to its local neighborhood, Wong and Cipolla [185] suggest to determine interest points by considering global information. The authors first detect subspaces of correlated movement in a video volume. These subspaces correspond to large movements such as an arm wave. Within these subspaces, local 2D saliency detection as well as temporal maxima in their coefficient matrix determine a sparse set of globally salient points (see Figure 2.8 (b)). Similarly, Bregonzio *et al.* [19] first compute the difference between subsequent frames to estimate the focus of attention. Then, Gabor filtering is used to detect salient points within these regions.

The presented space-time interest point detectors mainly differ in the type of saliency function as well as the sparsity of selected points. Moreover, majority of them are extensions of 2D image detectors to 3D in space and time, such as the Harris3D [88] and Hessian3D [184] detectors.

Local descriptors

Local descriptors capture shape and motion information in a local neighborhood patch surrounding interest points. Local descriptors summarize a video patch in a representation that is ideally invariant to background clutter, appearance and occlusions, and possibly to rotation and scale. The spatial and temporal size of a patch is usually determined by the scale of the interest point. Laptev and Lindeberg [93] are among the pioneers of designing local descriptors for videos. The authors develop and compare different descriptor types, including single and multi-scale higher-order derivatives (called local jets), histograms of optical flow, and histograms of spatio-temporal gradients. Histograms for optical flow and gradient components are computed in each cell of a $M \times M \times M$ grid layout, describing the local neighborhood of an interest point. Empirically, descriptors based on histograms of optical flow and spatio-temporal gradients are demonstrated to perform the best.

In a similar work, Dollár *et al.* [31] evaluate different local space-time descriptors based on brightness, gradient, and optical flow information. The authors investigate different descriptor variants: simple concatenation of pixel values, a grid of local histograms, and a single global histogram. Moreover, PCA is applied to reduce the dimensionality of each

descriptor variant. Overall, descriptors based on concatenated gradient information are shown to give the best performance.

Scovanner *et al.* [158] propose an extension of the image SIFT descriptor [105] to 3D in space and time. For a set of randomly sampled positions in a video sequence, spatio-temporal gradients are computed in the local neighborhood of each position. Each pixel in the neighborhood is weighted by a Gaussian centered on the given position and votes into a $M \times M \times M$ grid of histograms of oriented gradients. For orientation quantization, the gradients are represented in spherical coordinates ϕ, ψ, that are divided into a 8×4 histogram. To be rotation-invariant, the axis corresponding to $\phi = \psi = 0$ is aligned with the dominant orientation of the local neighborhood.

The histograms of oriented gradients (HOG) and histograms of optical flow (HOF) descriptors have been proposed by Laptev *et al.* [91]. To characterize local motion and appearance, HOG and HOF are combined in a late fusion approach. The histograms are accumulated in the space-time neighborhood of detected interest points. Each local region is subdivided into a $N \times N \times M$ grid of cells, wherein for each cell, 4-bin HOG histogram and 5-bin HOF histogram are computed. The normalized cell histograms are concatenated into the final HOG and HOF descriptors.

Kläser *et al.* [77] propose an extension of the HOG descriptor to 3D, referred to as the histograms of spatio-temporal gradient orientations (HOG3D). Their approach is based on a memory-efficient algorithm to compute 3D gradients for arbitrary scales and a generic 3D orientation quantization based on regular polyhedrons. Descriptor parameters are optimized for action recognition using Bag-of-Features representation.

Willems *et al.* [184] extend the image SURF descriptor [8] to video, called the extended SURF (ESURF) descriptor. Like the previous approaches, the authors divide 3D patches into a grid of local $M \times M \times M$ histograms. Each cell is represented by a vector of weighted sums of uniformly sampled responses of Haar-wavelets along the three axes.

The presented descriptors are mainly based on spatio-temporal gradients and optical flow. The HOG/HOF descriptors [91] are similar in concept to the SIFT descriptor [105], and combine both appearance and motion information in the final descriptor. The HOG3D [77] and SIFT3D [158] descriptors are similar, and both are extensions of the SIFT descriptor to 3D in space and time. The ESURF descriptor [184], however, is an extension of the image SURF descriptor [8], and is based on Haar-wavelets.

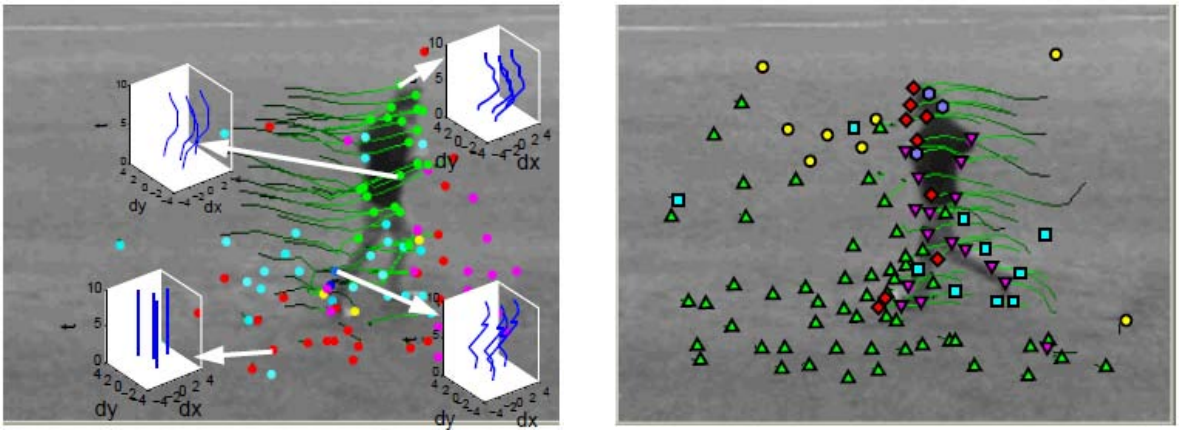


Figure 2.9: **Illustration of feature trajectories**. Trajectories are obtained by detecting and tracking spatial interest points, and are quantized to a library of trajectons, which are then used for action classification (figure reprinted from [112]).

Feature trajectories

In contrast to space-time interest points, feature trajectories are based on spatial interest points which are tracked over time. The shapes of trajectories encode the information about local motion patterns. Consequently, feature trajectories can be directly used as local features. Messing *et al.* [114] propose to represent feature trajectories of varying length as sequences of log-polar quantized velocities. Human activities are then modeled using a generative mixture of Markov chain models.

Hervieu *et al.* [65] propose a statistical trajectory-based HMM framework for analyzing sport video content, such as Formular One car racing and skiing. The target objects are tracked to compute the motion trajectories. The motion trajectories are described by the local differential features, which combine curvature and motion magnitude. HMMs then model the temporal causality of the local features and consequently, represent the motion trajectory. The proposed method has the potential to detect unexpected events in video.

In another approach, Matikainen *et al.* [112, 113] employ feature trajectories of a fixed length in a Bag-of-Features framework for human action classification, as illustrated in Figure 2.9. Feature trajectories computed in a video sequence are clustered together. For each cluster center, an affine transformation matrix is calculated. In addition to a velocity-based vector, the final trajectory descriptor contains elements of the affine transformation matrix for its assigned cluster center.

Feature trajectories are typically extracted using the KLT tracker or matching SIFT descriptors between frames. However, the quality as well as quantity of these features

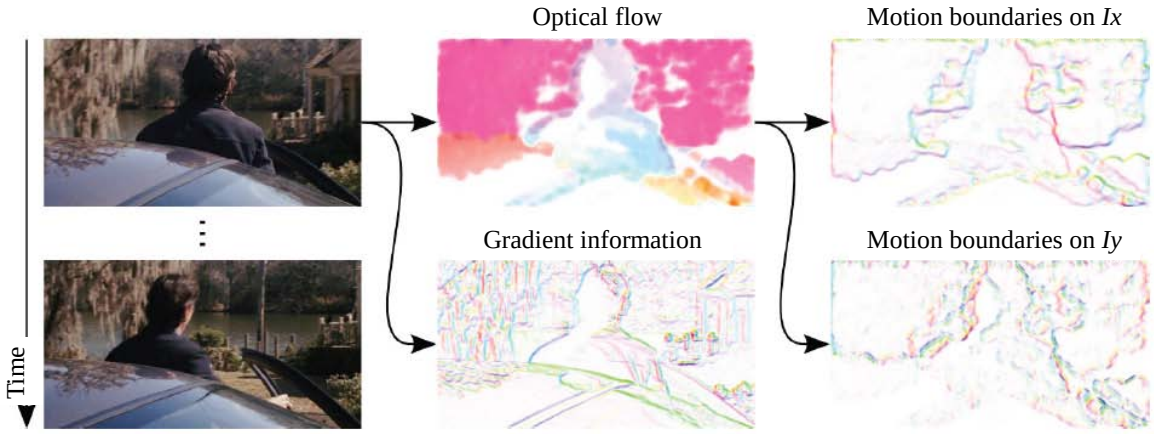


Figure 2.10: **Illustration of the information captured by HOG, HOF, and MBH descriptors**. Motion boundaries are computed as gradients of the x and y optical flow components separately. Contrary to optical flow, motion boundaries suppress most camera motion in the background and highlight the foreground motion. Unlike gradient information, motion boundaries eliminate most texture information from the static background. (figure reprinted from [175]).

is often not sufficient. Wang *et al.* [175] attempt to overcome such limitations, and propose to represent videos by rather dense trajectories. The authors sample dense points from each frame and track them based on displacement information from a dense optical flow field. Moreover, they introduce a novel descriptor based on motion boundary histograms (MBH), which is robust to camera motion (see Figure 2.10). In a comprehensive empirical evaluation, the proposed descriptor is consistently shown to outperform other state-of-the-art descriptors in a Bag-of-Features approach to human action classification.

Spatio-temporal relationship modeling

The basic BoF model represents a video sequence as an orderless collection of local features, and is therefore limited due to the lack of any geometrical relationship among features. However, there are a number of attempts to overcome the limitation by exploiting correlation between local features for selection or construction of higher-level features. Laptev *et al.* [91] include weak geometric relationship among local features by overlaying pre-defined spatio-temporal grids on video volumes (see Figure 2.11). In the spatial dimensions, a 1x1 grid (corresponding to the standard BoF representation), a 2x2 grid, a horizontal h3x1 grid as well as a vertical v1x3 grid is used. Moreover, the authors implement a denser 3x3 grid and a center-focused o2x2 grid where neighboring cells overlap by 50% of their width and height. For the temporal dimension, they subdivide the

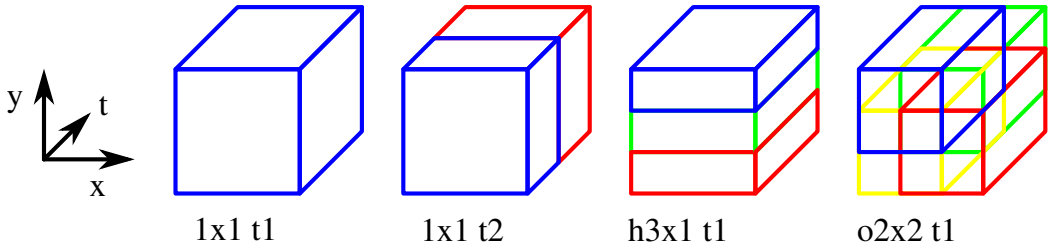


Figure 2.11: **Illustration of spatio-temporal grids**. Weak geometric information among local features can be incorporated in the Bag-of-Features model by overlying coarse spatio-temporal grids on video sequences (figure reprinted from [91]).

video sequence into 1 to 3 non-overlapping temporal bins resulting in t1 (standard BoF), t2 and t3 binnings. They also implement a center-focused ot2 binning. The combination of six spatial grids with four temporal binnings results in 24 possible spatio-temporal grids. The 24-level spatio-temporal grid layout is combined with shape and motion descriptors in a kernel fusion framework using a non-linear SVM. A greedy optimization strategy learns the best combination of grids and feature types per action class.

Savarese *et al.* [155] introduce correlations that describe co-occurrences of visual words within spatio-temporal neighborhoods. The codebook size strongly influences the classification performance. Too few entries do not allow for good discrimination, while too many visual words are likely to introduce noise due to sparsity of the histograms. Liu and Shah [103] attempt to solve this issue and determine the optimal size of the codebook using maximization of mutual information. Their method merges two codebook entries if they have comparable distributions. They additionally use spatio-temporal pyramid matching to exploit temporal information.

Gilbert *et al.* [51] propose to mine the compound features from dense spatio-temporal corners. The authors first detect spatio-temporal Harris corners on (x, y),(x, t),(y, t) planes. For each corner, they determine the relative spatial arrangement of all other corners in each video frame. This results in an extremely large number of features. Data mining techniques are then employed to discriminatively select those feature combinations that are informative of a class. Later, Gilbert *et al.* [52] introduce a hierarchical approach to combine Harris corner features. Frequent feature combinations that occur in a local spatio-temporal neighborhood are learned. These features are combined again in a hierarchical manner. In addition, the authors propose a voting scheme to localize actions in video sequences.

Another hierarchical approach based on SIFT feature trajectories is suggested by Sun *et*

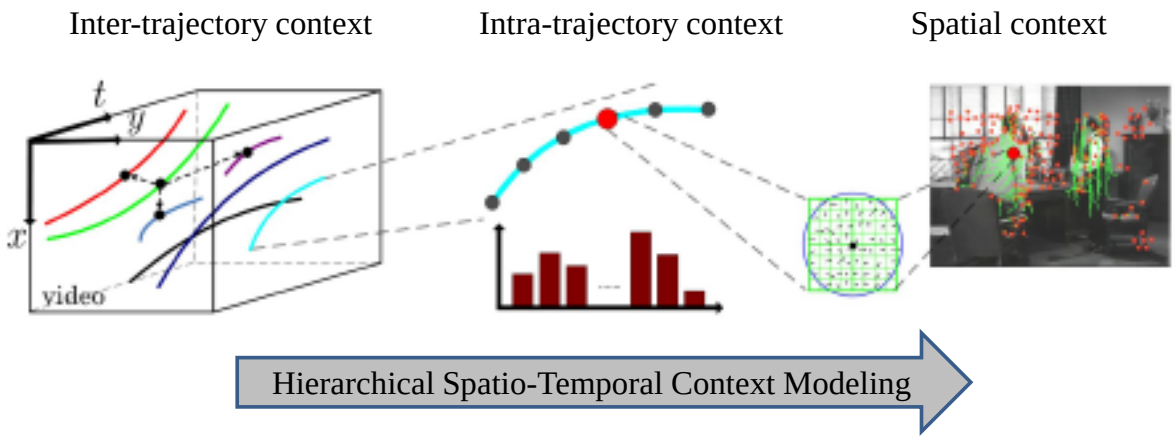


Figure 2.12: **A hierarchical approach to spatio-temporal context modeling**. The three levels of spatio-temporal context residing with SIFT-based trajectories are: (i) the point-level context (SIFT average descriptor), (ii) intra-trajectory context (trajectory transition descriptor), and (iii) inter-trajectory context (trajectory proximity descriptor) (figure reprinted from [167]).

al. [167]. The authors introduce different levels of context information: (i) point-level context encodes the local spatial neighborhood of a trajectory with an average SIFT descriptor; (ii) intra-trajectory context models the trajectory transition information; (iii) inter-trajectory context captures the relation among adjacent trajectories (see Figure 2.12). In order to capture dynamics of the last two levels, they employ stationary Markov distribution vectors. Furthermore, Multiple Kernel Learning (MKL) is proposed to prune the kernels towards speedup in algorithm evaluation.

Liu *et al.* [101] propose to combine motion and static appearance features to recognize realistic actions from YouTube videos. The authors mine the most informative features by applying the PageRank algorithm on the feature co-occurrence graph. Furthermore, a divisive information-theoretic algorithm is employed to construct compact yet discriminative visual vocabularies by grouping semantically related features. AdaBoost is used to integrate all the complementary features for action recognition.

Han *et al.* [60] propose to combine different local features with varying layouts and types: histograms of oriented gradients, histograms of optical flow, histograms of oriented spatio-temporal gradients. The authors suggest to combine multiple kernels using the Gaussian processes. In addition, they employ various object detectors (for full body, upper body, chairs, cars) to include information about the absence or presence of objects in the video sequences.

Kovashka and Grauman [80] propose to learn the shapes of space-time feature neighborhoods that are most discriminative for a given action category. Given a set of training videos, the authors construct a hierarchy of codebooks using neighborhoods of spatio-temporal feature points. The neighborhoods themselves are feature-centered, and their variable shape in the space and time dimensions is automatically learned. The selected shapes allow to capture varying extents of appearance and motion cues.

Matikainen *et al.* [113] present a method for representing pairwise spatio-temporal relationships between features in the Bag-of-Features framework. Instead of naively expanding codewords to include all possible pairs and relationships between features, their method produces an output whose size is proportional to the number of base codewords rather than to its square, which reduces the likelihood of overfitting and makes it more computationally efficient. The authors demonstrate their method to improve action classification performance with appearance as well as trajectory based features.

Local bag-of-features based methods have been a good choice because of their simplicity and robustness to certain variations in video. A wide variety of local space-time interest point detectors and descriptors is available. However, a fair comparison of these methods lacks, particularly due to the different experimental settings and various recognition methods employed. We overcome this limitation in Chapter 3 by performing a systematic evaluation of several local space-time feature detectors and descriptors under a common bag-of-features recognition framework. Moreover, local features and descriptors may provide limited discriminative power, implying ambiguity among features and sub-optimal recognition performance. To cope with this weakness, we in Chapter 4, propose to disambiguate local space-time features and to improve action recognition by integrating additional non-local cues with bag-of-features representation. For this purpose, we employ pre-trained object and action detectors (presented in Section 2.2) as well as spatio-temporal grids [91] to segment video into region classes and augment local features with corresponding region-class labels. Furthermore, local bag-of-features model offers limited semantics, as the representation is merely based on the statistics of local patches. In Chapter 5, we propose to represent video based on high-level semantically meaningful visual attributes. Our framework employs pre-trained detectors ([40],[17]) to predict the presence of characteristic objects, actions as well as poses in video. Significant changes of view points and appearance affects local descriptors and, therefore, introduces distraction to local representations. To address this problem, we in Chapter 6, propose a supervised approach to learn local motion descriptors from a large pool of annotated video data. The main motivation behind our approach is to construct action-characteristic representations

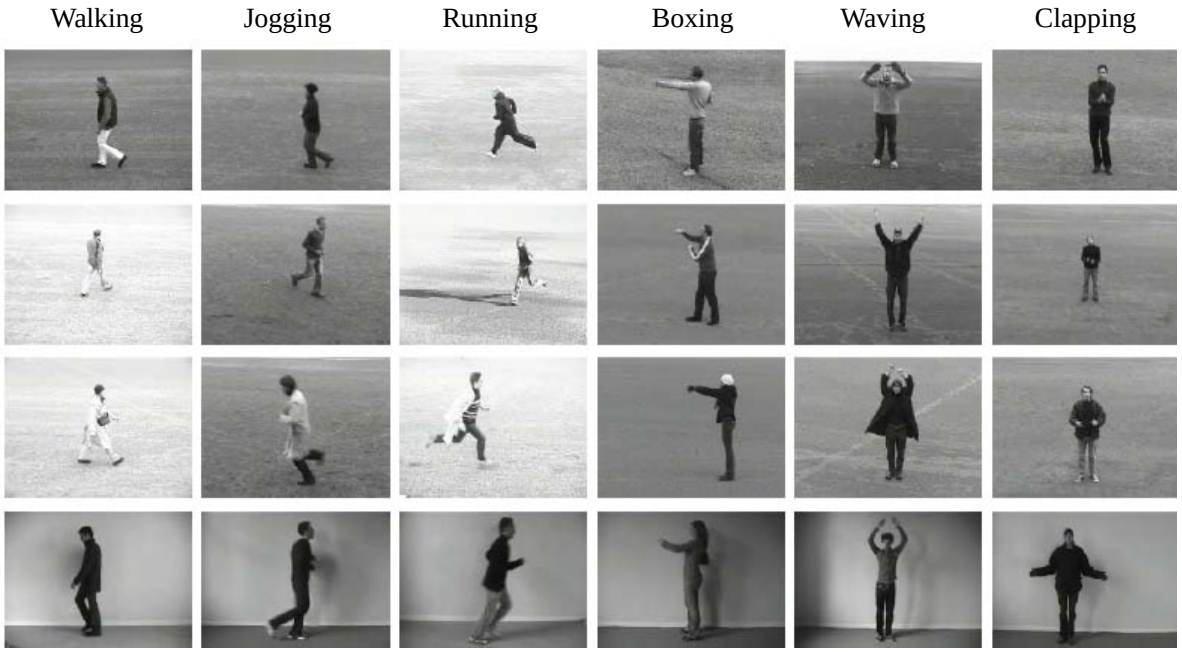


Figure 2.13: **Illustration of KTH-Actions dataset**. Sample frames for all the six action classes (column-wise) recorded under different scenarios (row-wise).

of body-joints undergoing specific motion patterns while learning invariance with respect to changes in camera views, lighting, human clothing, and other factors. We show that the proposed representation is discriminative as well as complimentary to bag-of-features representation.

2.4 Benchmark datasets

In this section, we present a detailed description of some of the benchmark datasets proposed in the literature over the past few years. All the experiments, in the rest of the thesis, are based on these datasets.

Section 2.4.1 presents the KTH-Actions dataset, which has been extensively used in the literature. The dataset, however, is comprised of simple actions with homogeneous background. The UCF-Sports dataset, presented in Section 2.4.2, has been collected from broadcast TV sports, such as BBC and ESPN. The dataset contains a variety of sport actions in high-resolution videos, while limited in its size. The relatively challenging and extensive YouTube-Actions and Hollywood-Actions datasets are described in Section 2.4.3 and Section 2.4.4 respectively. These two datasets offer relatively unconstrained and realistic variations in human actions.

2.4.1 KTH-Actions

The KTH-Actions dataset[2] has been introduced by Schüldt *et al.*[157]. It consists of six different human action classes: *walking, jogging, running, boxing, waving,* and *clapping* (see Figure 2.13). Each action class is performed several times by 25 subjects. The sequences are recorded in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. The background is homogeneous and static in most sequences. Apart from the zooming scenario, some of the scenes are recorded with a slightly shaking camera. Moreover, there is considerable variation in the performance and duration of actions, and somewhat in the view-point. Overall, the dataset consists of 2391 video sequences. In the original experimental setup proposed by its authors, the sequences are divided into test set (9 subjects: 2, 3, 5, 6, 7, 8, 9, 10, and 22) and training set (the remaining 16 subjects). Classification performance on this dataset is evaluated as average accuracy over all classes.

Most approaches that evaluate on the KTH-Actions dataset are based on bag-of-features framework. The original paper of the dataset [157] report 71.7% recognition rate. Recently, several approaches report recognition rates above 90% (e.g., [52, 80, 186]). More recently, the Action Bank representation of S. Sadanand *et al.*[152] achieves up to 98.2% recognition accuracy.

2.4.2 UCF-Sports

The UCF-Sports dataset[3] has been published by M. D. Rodriguez *et al.* [148]. It contains ten different types of 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* (see Figure 2.14). The dataset consists of 150 video sequences, which show a large intra-class variability. For most action classes, there is considerable variation in human appearance, action performance, camera movement, view-point, illumination, and background. The original setup proposed by its authors employs *leave-one-out* for testing, and the performance criterion for the multi-class classification is average accuracy over all classes.

The authors of the dataset [148] employ a template matching approach and report 69.2% performance accuracy. Other methods which evaluate on this dataset include [78] and [80], which achieve recognition performance of 86.7% and 87.3% respectively.

[2]Available at: `http://www.nada.kth.se/cvap/actions`
[3]Available at: `http://www.cs.ucf.edu/vision/public_html`

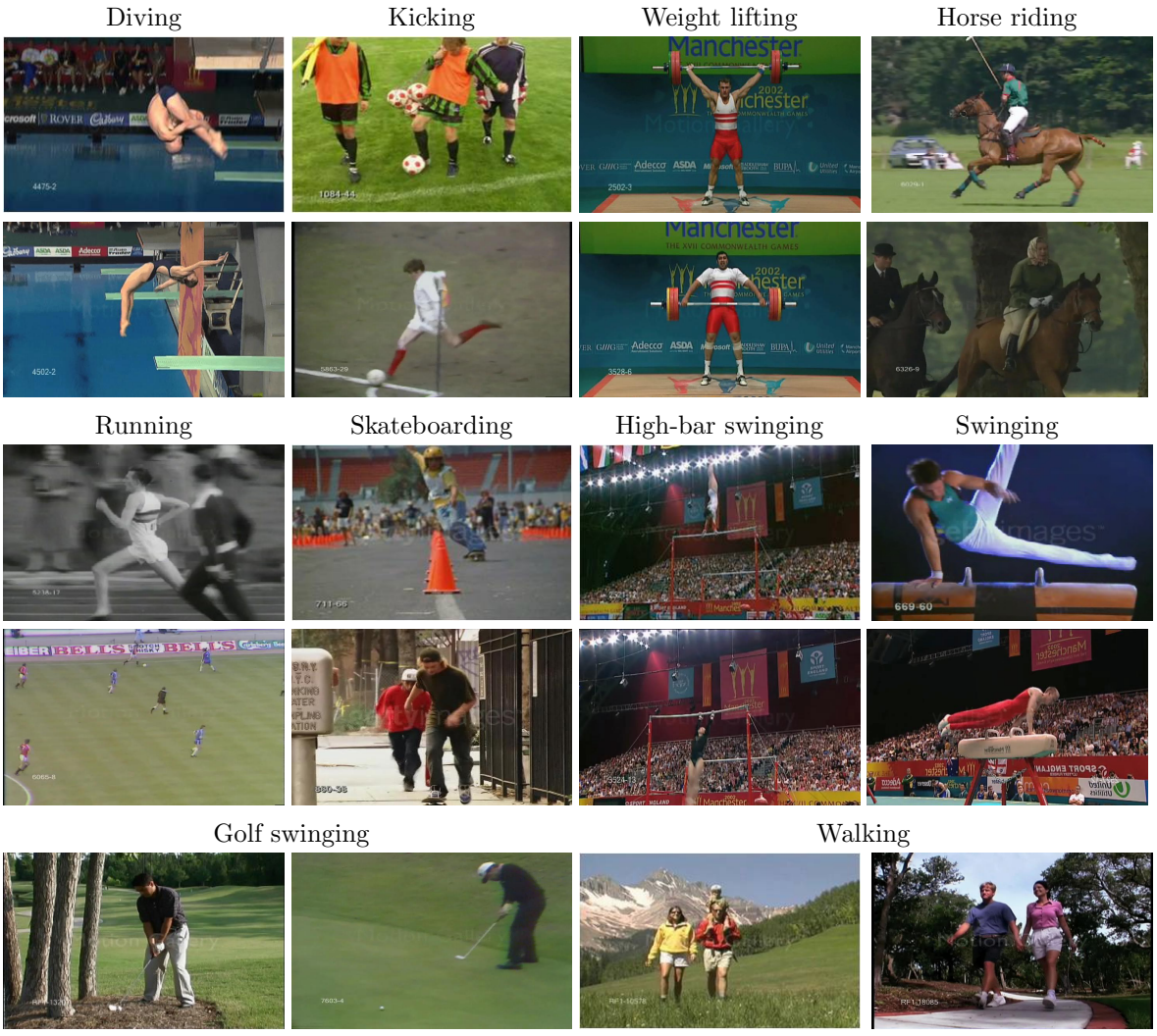


Figure 2.14: **Illustration of UCF-Sports dataset**. Two sample frames from all the ten action classes are shown.

Recently, H. Wang *et al.*[175] employ dense trajectories in a bag-of-features framework and achieve 88.2% recognition accuracy. More recently, S. Sadanand *et al.*[152] achieve 95.0% recognition rate with their Action Bank representation.

2.4.3 YouTube-Actions

The YouTube-Actions dataset[4] has been proposed by Liu *et al.*[101]. It is comprised of 11 action categories: *basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking* and *walking with a dog* (see Figure 2.15). This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, view-point, cluttered background and illumination conditions etc. The dataset contains a total of 1168 sequences. In the original setting, evaluation is carried out using cross validation for a set of 25 folds, which is defined by the authors. Average accuracy over all classes is used as the performance measure for multi-class classification.

The authors of this dataset [101] employ both static and motion features in a bag-of-features framework and report 71.2% recognition accuracy. Moreover, N. Ikizler-Cinbis *et al.*[69] propose a multiple instance learning (MIL) framework to integrate multiple feature channels, and achieve 75.2% recognition rate. Recently, H. Wang *et al.*[175] achieve 84.2% recognition rate, using dense trajectories in a bag-of-features framework.

2.4.4 Hollywood-Actions

The Hollywood-Actions dataset is comprised of two versions, namely Hollywood-1 [91] and Hollywood-2 [111]. In both cases, the authors use movie scripts to avoid exhaustive manual annotation of several hundreds of hours of movie data. Movie scripts provide textual description of the movie content, such as scenes, characters, transcribed dialogues, and human actions. A two-step process is employed to retrieve action samples. In the first step, scripts are aligned to movie subtitles, since they usually lack the time information. In the second step, classifiers are trained on a bag-of-words representation of the scene description for different action classes. Several features are used: bag-of-words over single words, over adjacent pairs of words, as well as over pairs of words in a small neighborhood. This allows to cope with significant variations in the text description. The classifiers are subsequently used to retrieve action samples from the movie data. The

[4]Available at: `http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html`

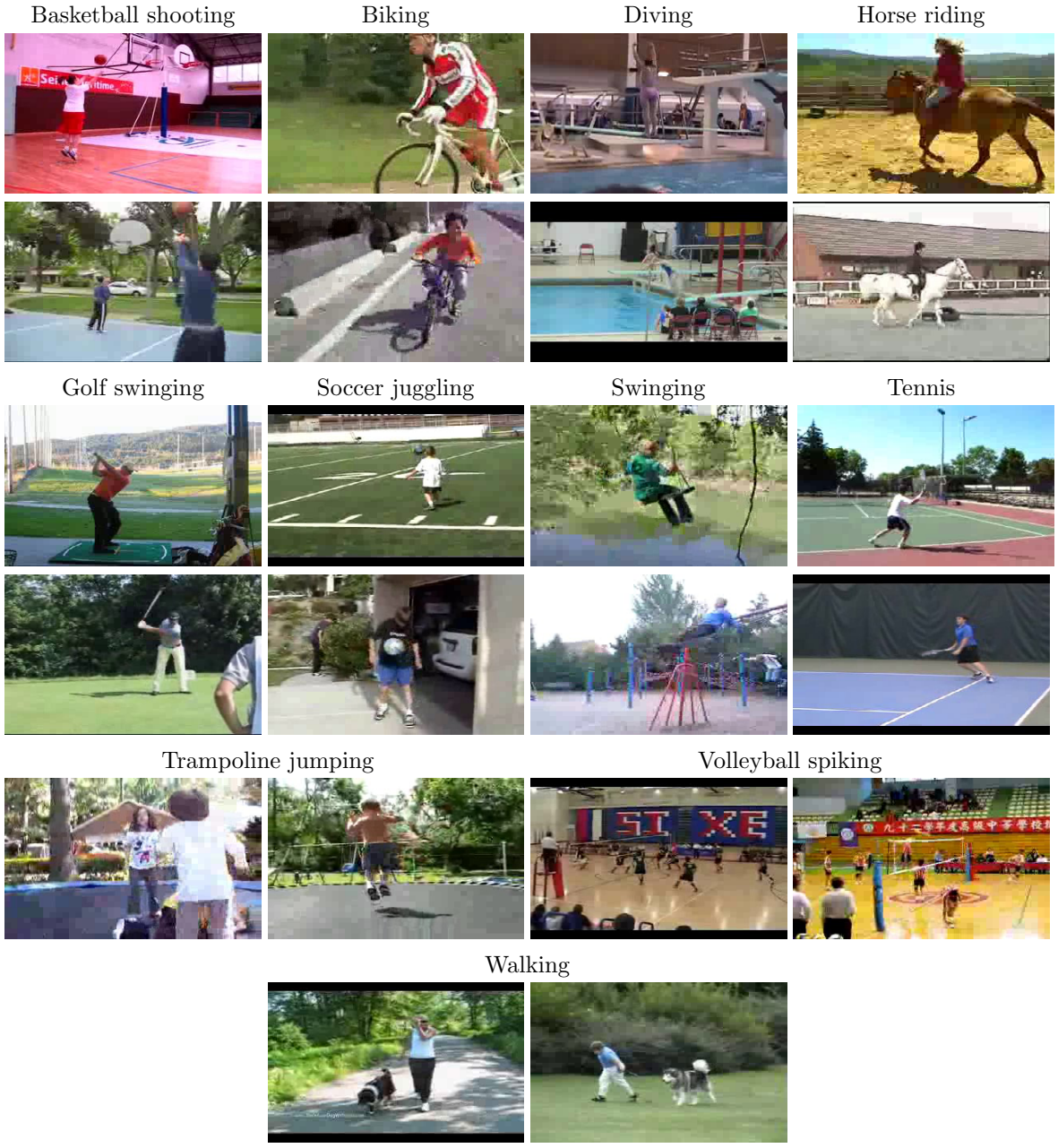


Figure 2.15: **Illustration of YouTube-Actions dataset**. Two sample frames from each of the eleven action classes are shown.

Figure 2.16: **Illustration of Hollywood-Actions dataset**. Two sample frames from each of the twelve action classes are shown.

authors manually correct the automatic class labels and provide the *clean* train and test set. Additionally, the authors provide the automatically labelled *noisy* train set.

The first version[5], i.e., Hollywood-1, has been published by I. Laptev *et al.*[91]. It contains eight different action classes: *answering the phone, getting out of car, hand shaking, hugging, kissing, sitting down, sitting up,* and *standing up.* The action samples have been collected from 32 different Hollywood movies. The full dataset consists of 663 video samples, divided into a clean train set (219 sequences) and a clean test set (211 sequences), where train and test sequences are obtained from different movies. The additional noisy train set contains 233 video sequences.

The second and extended version[6], i.e., Hollywood-2, has been introduced by M. Marszałek *et al.*[111]. In total, it is comprised of samples from 69 different Hollywood movies. The initial eight action classes are extended by adding four additional classes: *driving a car, eating, fighting,* and *running.* Figure 2.16 illustrates sample frames from all the twelve action classes. In total, the dataset is comprised of 2517 video samples, split into a manually cleaned train set (823 sequences) and a test set (884 sequences). The noisy train set consists of 810 sequences. Train and test sequences are obtained from different movies.

The actions in the Hollywood dataset are performed by professional actors, involving a wide range of realistic variations in action style, view-point, occlusion, camera movement, and background etc. This dataset is very challenging and involves inter-actions with people (fight-person, hand-shake, hug-person, kiss) and objects (answer-phone, drive-car, getout-car). Most of the samples in this dataset are at the scale of the upper-body, but some record the entire body or a close-up of the face. The performance measure for both, Hollywood-1 and Hollywood-2, is calculated by computing the average precision (AP) for each of the action classes and reporting the mean AP over all the classes (i.e., mAP). Note that this follows the evaluation procedure established by the Pascal Visual Object Class Challenge (2007) [36].

The authors of Hollywood-1 [91] employ the Harris3D features in combination with a set of spatio-temporal grids, and report 38.4 mAP using the clean evaluation setup. The current state-of-the-art performance on Hollywood-1 is by A. Gilbert *et al.*[53], i.e., 53.5 mAP. The authors propose a hierarchical data mining approach to group simple 2D Harris points, and use a simple voting scheme for classification. On Hollywood-2, they

[5]Available at: `http://www.di.ens.fr/~laptev/download.html`
[6]Available at: `http://www.di.ens.fr/~laptev/download.html`

achieve 50.9 mAP. Recently, H. Wang *et al.*[175] report 58.3 mAP on Hollywood-2.

The presented datasets vary in terms of appearance, background, lighting, actions, and styles, etc. The KTH-Actions dataset is the simplest with homogeneous background. It has been extensively used for bag-of-features based methods with up to 98.2% recognition accuracy achieved [152]. UCF-Sports and YouTube-Actions mainly contain sport actions, and offer relatively unconstrained settings. In particular, YouTube-Actions dataset presents relatively realistic variations, as it has been collected from YouTube. Hollywood-Actions dataset is the most challenging, and has been collected from Hollywood movies. Up to 58.3 mAP has been achieved [175] on the Hollywood-2 version of this dataset. Several other datasets are available, such as the HMDB dataset [82]. HMDB dataset provides a large-scale testing environment with up to 51 action categories.

Chapter 3

Evaluation of local space-time features

Contents

As discussed in Chapter 2, local image and video representations have been shown successful for many recognition tasks such as object and scene recognition [42, 95] as well as human action recognition [157, 91]. Many different space-time feature detectors [88, 31, 184, 70, 185, 126] and descriptors [91, 184, 77, 158, 93] have been proposed in the past few years (see Section 2.3.3). Feature detectors usually select spatio-temporal locations and scales in video by maximizing specific saliency functions. The detectors usually differ in the type and the sparsity of selected points. Feature descriptors capture shape and

motion in the neighborhoods of selected points using image measurements such as spatial or spatio-temporal image gradients and optical flow.

While specific properties of detectors and descriptors have been advocated in the literature, their justification is often insufficient due to the limited and non-comparable experimental evaluations used. For example, results are frequently presented for different datasets such as the KTH-Actions dataset [157, 77, 91, 184, 31, 185, 70], the Weizman dataset [11, 158] or the aerobic actions dataset [126]. For the common KTH-Actions dataset [157], results are often non-comparable due to the different experimental settings used. Furthermore, most of the previous evaluations are reported for actions in controlled environments such as in KTH-Actions and Weizman datasets. It is therefore unclear how these methods generalize to action recognition in realistic setups [91, 148].

Several evaluations of local space-time features have been reported in the past. Laptev [87] evaluates the repeatability of space-time interest points as well as the associated accuracy of action recognition under changes in spatial and temporal video resolution as well as under camera motion. Similarly, Willems *et al.* [184] evaluate repeatability of detected features under scale changes, in-plane rotations, video compression and camera motion. Local space-time descriptors are evaluated by Laptev et al. [93], where the comparison includes families of higher-order derivatives (local jets), image gradients and optical flow. Dollár *et al.* [31] compare local descriptors in terms of image brightness, gradient and optical flow. Scovanner *et al.*. [158] evaluate 3D-SIFT descriptor and its two-dimensional variants. Jhuang *et al.* [70] evaluate local descriptors in terms of the magnitude and orientation of space-time gradients as well as optical flow. Kläser *et al.*[77] compare space-time HOG descriptor with HOG and HOF descriptors [91]. Willems *et al.* [184] evaluate the extended SURF descriptor. However, evaluations in these works are usually limited to a single detection or description method as well as to a single dataset.

In this chapter, we overcome the above-mentioned limitations and provide an extensive comparison for a number of local space-time detectors and descriptors. We evaluate performance of three space-time interest point detectors and six descriptors along with their combinations on three datasets with varying degree of difficulty. Moreover, we introduce and evaluate dense features obtained by regular sampling of local space-time patches, motivated by excellent results recently obtained by dense sampling in the context of object recognition [97, 74]. We, furthermore, investigate the influence of spatial video resolution and shot boundaries on the performance. We also compare methods in terms of their sparsity as well as the computational speed of available implementations. All the experiments are reported for the same bag-of-features recognition framework.

The rest of the chapter is organized as follows. In Section 3.1, we give a detailed description of the local spatio-temporal features included in our comparison. Section 3.2 then presents the evaluation framework based on the bag-of-features approach. Finally, Section 3.3 compares the results obtained for different features while Section 3.4 concludes the chapter with a discussion.

3.1 Local space-time video features

This section describes local feature detectors and descriptors used in the evaluation. Methods are selected based on their use in the literature as well as the availability of the implementation. In all cases, we use the original implementation and parameter setting provided by the respective authors.

3.1.1 Detectors

Harris3D detector: It is proposed by Laptev and Lindeberg in [88], as a space-time extension of the Harris detector [62]. The authors compute a spatio-temporal second-moment matrix at each video point $\mu(\cdot;\sigma,\tau) = g(\cdot; s\sigma, s\tau) * (\nabla L(\cdot;\sigma,\tau)(\nabla L(\cdot;\sigma,\tau))^T)$ using independent spatial and temporal scale values σ, τ, a separable Gaussian smoothing function g and space-time gradients ∇L. They define locations of space-time interest points as local maxima of $H = \det(\mu) - k\,\mathrm{trace}^3(\mu)$, $H > 0$. The authors propose an optional mechanism for spatio-temporal scale selection. This is not used in our experiments, but we use points extracted at multiple scales based on a regular sampling of the scale parameters σ, τ. This has shown to give excellent results in [91]. We use the original implementation available on-line[1] and standard parameter settings $k = 0.0005$, $\sigma^2 = 4, 8, 16, 32, 64, 128$, $\tau^2 = 2, 4$. Figure 3.2 (2nd row) illustrates interest point detections by the Harris3D detector on example frames of a video sequence.

Cuboid detector: It is proposed by Dollár *et al.* [31] and is based on temporal Gabor filters. The response function has the form: $R = (I * g * h_{ev})^2 + (I * g * h_{od})^2$, where $g(x, y; \sigma)$ is the 2D Gaussian smoothing kernel, applied only along the spatial dimensions, and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally, defined by $h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$. The authors set $\omega = 4/\tau$, effectively giving the response function R two parameters σ and τ , corresponding roughly to the spatial and temporal scales of the detector. Interest points

[1] Available at: `http://www.irisa.fr/vista/Equipe/People/Laptev/download.html\#stip`

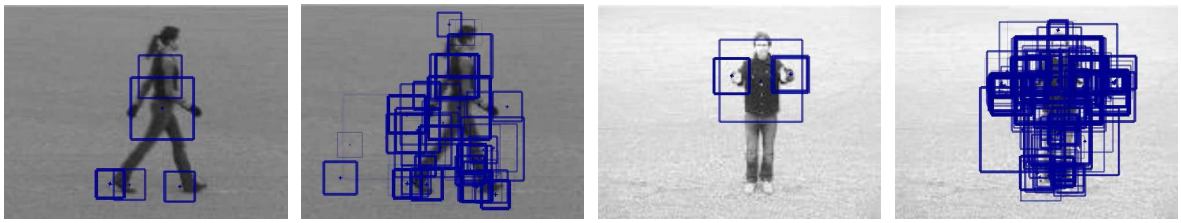


Figure 3.1: **Illustration of space-time interest points detected using the Hessian3D detector**. Interest points are shown for different threshold values (figure reprinted from [184]).

are detected at the local maxima of the response function R. We use the code from the authors' website[2] and detect features using standard scale values $\sigma = 2, \tau = 4$. Figure 3.2 (3rd row) shows interest point detections by the Cuboid detector on some video frames.

Hessian3D detector: It is proposed by Willems *et al.* [184] as a spatio-temporal extension of the Hessian saliency measure used in [9, 100] for blob detection in images. The authors use the determinant of the 3D Hessian matrix to measure the saliency. The position and scale of the interest points are simultaneously localized without any iterative procedure. In order to speed up the detector, approximative box-filter operations are used on an integral video structure. Each octave is divided into 5 scales, with a ratio between subsequent scales in the range $1.2 - 1.5$ for the inner 3 scales. The determinant of the Hessian is computed over several octaves for both the spatial and temporal scales. A non-maximum suppression algorithm, then, selects joint extrema over space, time and scales. Figure 3.1 presents some interest point detections for different thresholds. We use the executables from the authors' website[3] and employ the default parameter setting. Figure 3.2 (4th row) presents example detections by the Hessian3D detector on some video frames.

Dense sampling: Video blocks at regular positions and scales in space and time are extracted. There are 5 dimensions to sample from: (x, y, t, σ, τ), where σ and τ are the spatial and temporal scales, respectively. In our experiments, the minimum size of a 3D patch is 18×18 pixels and 10 frames. Spatial and temporal sampling are done with 50% overlap. Multi-scale patches are obtained by multiplying σ and τ by a factor of $\sqrt{2}$ for consecutive scales. In total, we use 8 spatial and 2 temporal scales, since we consider the spatial scale to be more important than the time scale. We consider all combinations of spatial and temporal scales, i.e., we sample an image 16 times with different σ and

[2]Available at: `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`
[3]Available at: `http://homes.esat.kuleuven.be/~gwillems/research/Hes-STIP`

τ parameters. Figure 3.2 (last row) illustrates dense sampling on example frames of a video sequence.

3.1.2 Descriptors

For each given sample point (x, y, t, σ, τ), a feature descriptor is computed for a 3D video patch centered at (x, y, t). Its spatial size $\Delta_x(\sigma), \Delta_y(\sigma)$ is a function of σ and its temporal length $\Delta_t(\tau)$ a function of τ. We consider the following descriptors:

Cuboid descriptor: It is proposed along with the Gabor detector by Dollár *et al.* [31]. The size of the descriptor is given by $\Delta_x(\sigma) = \Delta_y(\sigma) = 2 \cdot ceil(3\sigma) + 1$ and $\Delta_t(\tau) = 2 \cdot ceil(3\tau) + 1$. We follow the authors' setup and concatenate the gradients computed for each pixel in the patch into a single vector. Then, principal component analysis (PCA) is used to project the feature vector to a lower dimensional space. We download the code from the authors' website and use the default settings (e.g., the size of descriptor after PCA projection is 100). The PCA basis is computed on the training samples.

HOG/HOF descriptors: They are introduced by Laptev *et al.* in [91]. To characterize local motion and appearance, the authors compute histograms of spatial gradient and optic flow accumulated in space-time neighborhoods of detected interest points. For the combination of HOG/HOF descriptors with interest point detectors, the descriptor size is defined by $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma$ and $\Delta_t(\tau) = 8\tau$. Each 3D patch volume is subdivided into a (n_x, n_y, n_t) grid of cells; for each cell, 4-bin histograms of gradient orientations (*HOG*) and 5-bin histograms of optic flow (*HOF*) are computed. Normalized histograms are concatenated into HOG, HOF as well as HOGHOF descriptor vectors (see Figure 3.3) and are similar in spirit to the well known SIFT descriptor. In our evaluation, we use the grid parameters $n_x, n_y = 3$, $n_t = 2$, as suggested by the authors. We notice low dependency of results for different choices of the scale factor for σ, τ in general. We use the original implementation available on-line.

When computing the HOG/HOF descriptors for the Hessian3D detector, we optimize the mappings $\sigma = \alpha\sigma^h$ and $\tau = \beta\tau^h$ w. r. t. α and β for the HOG/HOF scale parameters σ, τ and the scale parameters σ^h, τ^h returned by the Hessian3D detector. For the Cuboid detector, (computes at low space-time scale values), we fix the scales of HOG/HOF descriptors to $\sigma^2 = 4$ and $\tau^2 = 2$.

HOG3D descriptor: It is proposed by Kläser et al. [77]. It is based on histograms of 3D gradient orientations and can be seen as an extension of the popular SIFT descriptor [105] to video sequences. Gradients are computed using an integral video

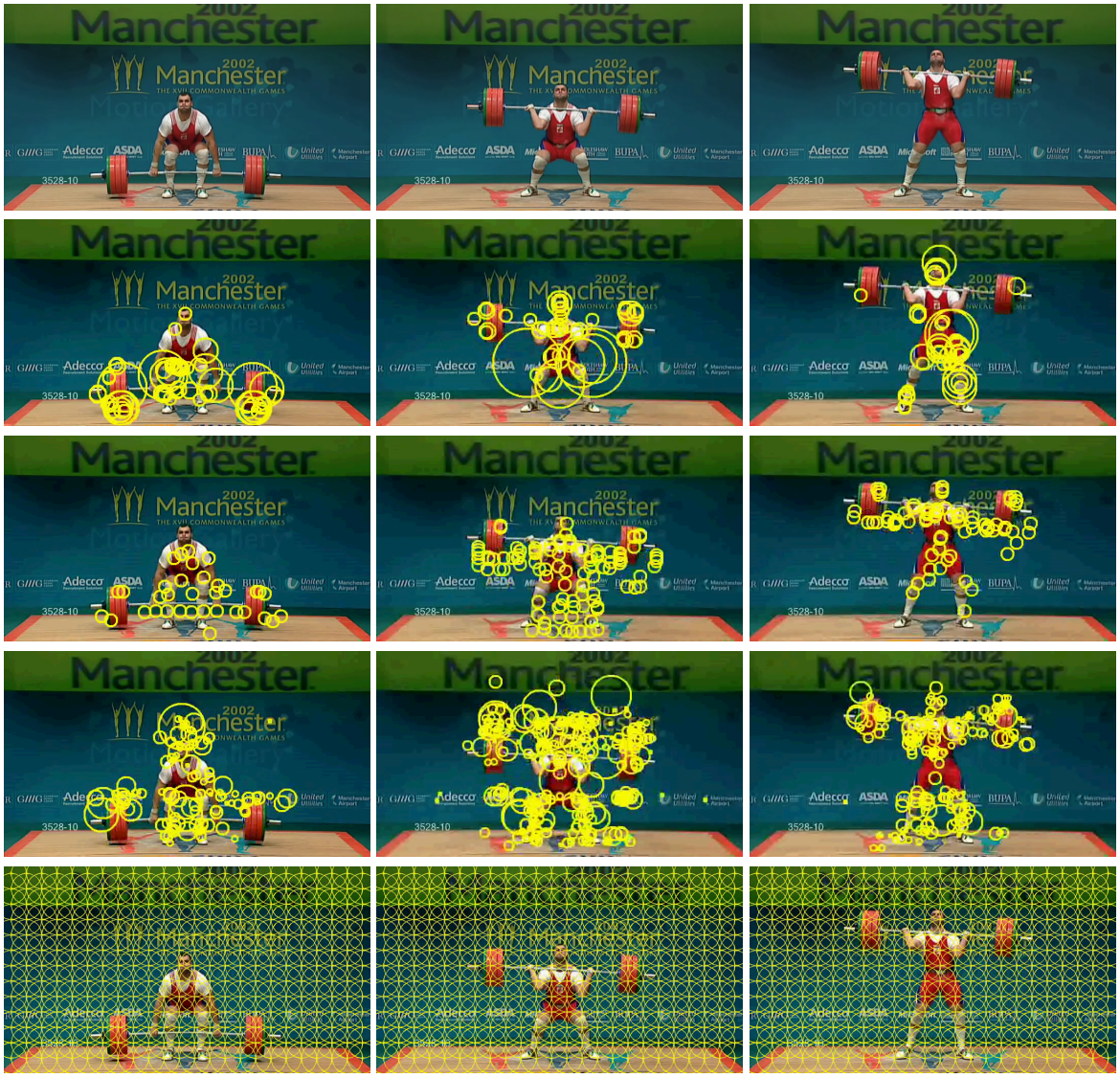


Figure 3.2: **Visualization of interest points detected by the different detectors on subsequent frames of a video sequence**. Harris3D (2nd row), Gabor (3rd row), Hessian3D (4th row) and Dense sampling (5th row).

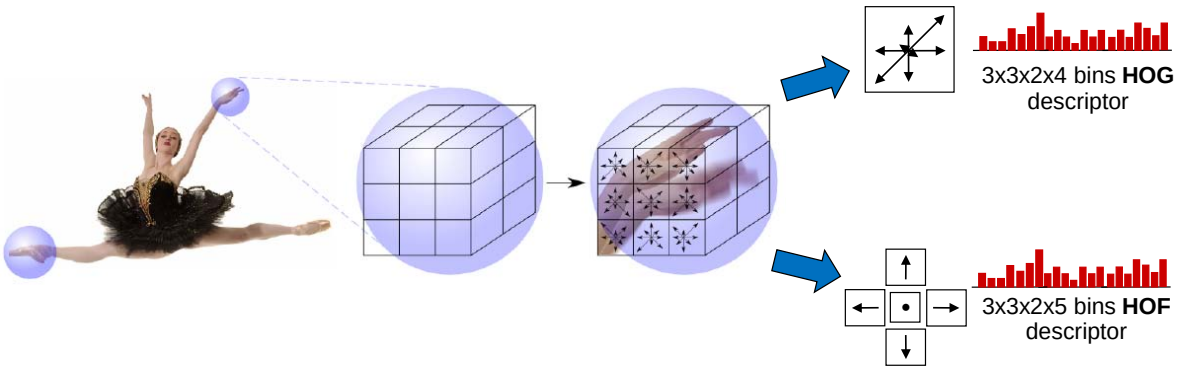


Figure 3.3: **Illustration of the HOG/HOF descriptor**. An interest region is described by a cuboid volume, divided into a grid of cells. For each cell, a histogram of oriented spatial gradients (HOG) as well as histogram of optical flow (HOF) is computed. The final descriptor is the concatenation of all the HOG and HOF histograms, corresponding to each grid cell. (figure reprinted from [91]).

representation. Regular polyhedrons are used to uniformly quantize the orientation of spatio-temporal gradients. The descriptor, therefore, combines shape and motion information at the same time. A given 3D patch is divided into $n_x \times n_y \times n_t$ cells. The corresponding descriptor concatenates gradient histograms of all cells and is then normalized (see Figure 3.4). We use the executable from the authors' website[4] and apply their recommended parametric settings for all feature detectors: descriptor size $\Delta_x(\sigma) = \Delta_y(\sigma) = 8\sigma$, $\Delta_t(\tau) = 6\tau$, number of spatial and temporal cells $n_x = n_y = 4$, $n_t = 3$, and icosahedron as the polyhedron type for quantizing orientations.

Extended SURF (ESURF) descriptor: It is proposed by Willems *et al.* [184], and extends the image SURF descriptor [8] to videos. Like for previous descriptors, the authors divide 3D patches into $n_x \times n_y \times n_t$ cells. The size of the 3D patch is given by $\Delta_x(\sigma) = \Delta_y(\sigma) = 3\sigma$, $\Delta_t(\tau) = 3\tau$. For the feature descriptor, each cell is represented by a vector of weighted sums $v = (\sum d_x, \sum d_y, \sum d_t)$ of uniformly sampled responses of the Haar-wavelets d_x, d_y, d_t along the three axes. We use the executables from the authors' website with the default parameter setting.

3.2 Evaluation framework

Our evaluation framework is based on the bag-of-features (BoF) representation and Support Vector Machines (SVM) classification, as described in Section 2.1. Here, we

[4]Available at: `http://lear.inrialpes.fr/software`

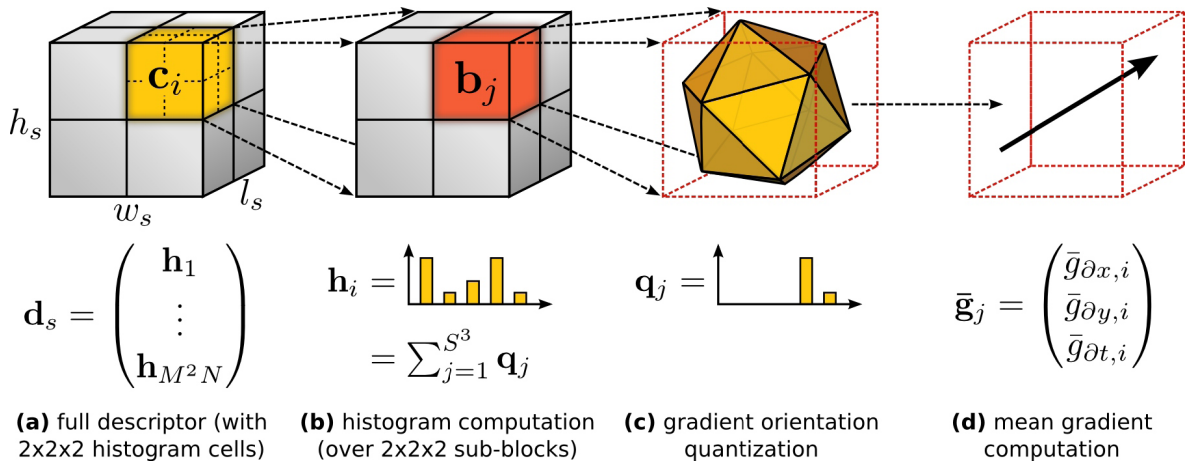


Figure 3.4: **Illustration of the HOG3D descriptor**. (a) The region of interest is divided into a grid of oriented gradient histograms; (b) each histogram is computed over a grid of mean gradients; (c) each gradient orientation is quantized using regular polyhedrons; (d) each mean gradient is computed using integral videos. (figure reprinted from [77]).

follow Section 2.1.1, and use k-means clustering to construct visual vocabularies. We set the number of visual words k to 4000 which has shown to empirically give good results for a wide range of datasets. To limit the complexity, we cluster a subset of 100,000 randomly selected training features. To increase precision, we initialize k-means 8 times and keep the result with the lowest error. Features are assigned to their closest vocabulary word using Euclidean distance. The resulting histograms of visual word occurrences are used as video sequence representations.

For classification, we use a non-linear Support Vector Machine [23] with a χ^2-kernel [91]. For multi-class classification, we use the *one-against-all* approach.

3.3 Experiments

We carry out experiments on three action datasets; KTH-Actions, UCF-Sports and Hollywood-2 actions (see Section 2.4 for the detailed description). In this section, we present experimental results for various detector/descriptor combinations. Recognition results are presented in the order of different datasets in Section 3.3.1–3.3.3. Section 3.3.4 evaluates different parameters for dense sampling. The computation complexity of tested methods is evaluated in Section 3.3.5

Due to high memory requirements of some descriptor/detector codes, we sub-sample

[%]	HOG3D	HOG/HOF	HOG	HOF	Cuboid	ESURF
Harris3D	89.0	91.8	80.9	**92.1**	-	-
Cuboid	90.0	88.7	82.3	88.2	89.1	-
Hessian3D	84.6	88.7	77.7	88.6	-	81.4
Dense	85.3	86.1	79.0	88.0	-	-

Table 3.1: Average accuracy for various detector/descriptor combinations on the KTH-Actions dataset.

original UCF-Sports and Hollywood-2 sequences to half spatial resolution in all our experiments. This enables us to compare all methods on the same data. We evaluate the effect of subsampling for the Hollywood-2 dataset in Section 3.3.3. The ESURF and Gradient descriptors are not evaluated for other detectors than those used in original papers. Unfortunately, separate implementations of these descriptors were not available at the evaluation time. Note that due to random initialization of k-means clustering used for vocabulary generation, we observe a standard deviation of approximately 0.5% in our experiments.

3.3.1 KTH-Actions dataset

KTH-Actions [157] is to date the most common dataset in evaluations of action recognition. Among recently reported results, Laptev *et al.* [91] obtain 91.8% using a combination of HOG and HOF descriptors while Kläser *et al.* [77] get 91.4% with the HOG3D descriptor. Both methods use Harris3D detector and follow the original experimental setup of [157]. Adopting the Gabor detector, Liu and Shah [103] report 94.16%, and Bregonzio *et al.* [19] obtain 93.17% with a 2D Gabor filter based detector. Note, however, that these results are obtained for a simpler Leave-One-Out Cross-Validation (LOOCV) setting and are not directly comparable to the results in this chapter.

Our results for different combinations of detectors and descriptors evaluated on the KTH-Actions dataset are illustrated in Table 3.1. The best results are obtained for Harris3D + HOF (92.1%) and HOG/HOF (91.8%). These results are comparable to 91.8% reported in [91] for Harris3D + HOG/HOF. For Harris3D + HOG3D, we only reach 89.00%, about 2.5% lower than the original result in [77]. This could be explained by the different strategy of vocabulary generation (i.e., random sampling) used in [77]. For the Gabor detector, the best result 90.0% is obtained with HOG3D descriptor. The performance of Hessian3D and Dense detectors are below Harris3D and Gabor. The low performance of dense sampling on KTH-Actions may be explained by the large number of

[%]	HOG3D	HOG/HOF	HOG	HOF	Cuboid	ESURF
Harris3D	79.7	78.1	71.4	75.4	-	-
Cuboid	82.9	77.7	72.7	76.7	76.6	-
Hessian3D	79.0	79.3	66.0	75.3	-	77.3
Dense	**85.6**	81.6	77.4	82.6	-	-

Table 3.2: Average accuracy for various detector/descriptor combinations on the UCF-Sports dataset.

features corresponding to the non-informative background. When comparing performance of different descriptors, we note that HOG features alone show low performance which highlights the importance of motion information for action recognition. Moreover, HOG/HOF and HOF give the best results in combination with Harris3D, Hessian3D and Dense features.

3.3.2 UCF-Sports dataset

The results for different combinations of detectors and descriptors evaluated on the UCF-Sports actions are illustrated in Table 3.2. The best result 85.6% over different detectors is obtained by the dense sampling. We note that dense features outperform sparse features for each of the descriptor. This can be explained by the fact that dense features capture background which may provide useful context information. Scene context indeed may be helpful for sports actions which often involve specific equipment and scene types. The second-best result 82.9% is obtained for the Gabor detector. Also above 80% are dense points in combination with HOG/HOF and HOF. Harris3D and Hessian3D detectors perform similar at the level of 80%. Among different descriptors, HOG3D provides the best results for all detectors except Hessian3D. HOG/HOF gives second-best result for UCF-Sports. The authors of the original paper [148] report 69.2% for UCF-Sports. Their result, however, does not correspond to the version of UCF-Sports dataset available on-line used in our evaluation.

3.3.3 Hollywood-2 dataset

Finally, evaluation results for Hollywood-2 actions are presented in Table 3.3. As for the UCF-Sports dataset, the best result 47.4% is obtained for dense sampling while interest point detectors demonstrate similar and slightly lower performance. We assume dense sampling again benefits from a more complete description of motions and the rich context information. Among different descriptors, HOG/HOF performs the best. Unlike

[mAP]	HOG3D	HOG/HOF	HOG	HOF	Cuboid	ESURF
Harris3D	43.7	45.2	32.8	43.3	-	-
Cuboid	45.7	46.2	39.4	42.9	45.0	-
Hessian3D	41.3	46.0	36.2	43.0	-	38.2
Dense	45.3	**47.4**	39.4	45.5	-	-

Table 3.3: Mean AP for various detector/descriptor combinations on the Hollywood-2 dataset.

[mAP]	HOG3D	HOG/HOF	HOG	HOF
Reference	43.7	45.2	32.8	43.3
Without shot boundary features	43.6	**45.7**	35.3	43.4
Full resolution videos	45.8	**47.6**	39.7	43.9

Table 3.4: Comparison of the Harris3D detector on (top) videos with half spatial resolution, (middle) with removed shot boundary features and (bottom) on the full resolution videos.

in results for KTH-Actions, here the combination of HOF and HOG improves HOF with about 2 percent. The HOG3D descriptor performs similar to HOF.

Shot boundary features: Since action samples in Hollywood-2 are collected from movies, they contain many shot boundaries, which cause many artificial interest points. To investigate the influence of shot boundaries on recognition results, we compare in Table 3.4 the performance of the Harris3D detector with and without shot boundary features. Results for HOG/HOF and HOG demonstrate 0.5% and 2% improvement respectively when removing shot boundary features while the change in performance for other descriptors is minor. We conclude that shot boundary features do not influence our evaluation significantly.

Influence of subsampling: We also investigate the influence of reduced spatial resolution adopted in our Hollywood-2 experiments. In Table 3.4 recognition results are reported for videos with full and half spatial resolution using the Harris3D detector. The performance is consistently and significantly increased for all tested descriptors for the case of full spatial resolution. Note that for full resolution, we obtain approximately 4 times more features per sequence than for half resolution.

3.3.4 Dense sampling parameters

Given the best results obtained with dense sampling, we further investigate the performance as a function of different minimal spatial sizes of dense descriptors (see Table 3.5).

Spatial size	Hollywood-2 [mAP]				UCF-Sports [%]			
	HOG3D	HOG/HOF	HOG	HOF	HOG3D	HOG/HOF	HOG	HOF
18×18	**45.3**	47.4	39.4	45.5	**85.6**	**81.6**	**77.4**	82.6
24×24	45.1	**47.7**	**39.4**	**45.8**	82.0	81.4	76.8	**84.0**
36×36	44.8	47.3	36.8	45.6	78.6	79.1	76.5	82.4
48×48	42.8	46.5	35.8	45.5	78.8	78.6	73.9	79.0
72×72	39.7	45.2	32.2	43.0	77.8	78.8	69.6	78.4

Table 3.5: Average accuracy for dense sampling with varying minimal spatial sizes on the Hollywood-2 and UCF-Sports dataset.

	Harris3D + HOG/HOF	Hessian3D + ESURF	Cuboid-detector + Cuboid-descriptor	Dense + HOG3D	Dense + HOG/HOF
Frames/second	1.6	4.6	0.9	0.8	1.2
Features/frame	31	19	44	643	643

Table 3.6: Average speed and average number of generated features for different methods.

As before, further spatial scales are sampled with a scale factor of $\sqrt{2}$. As in Sections 3.3.3 and 3.3.2, we present results for Hollywood-2 and UCF-Sports videos with half spatial resolution. We observe no significant improvements for different temporal lengths, therefore we fix the temporal length to 10 frames. The overlapping rate for dense patches is set to 50%. We can see that the performance increases with smaller spatial size, i.e., when we sample denser.

3.3.5 Computational complexity

Here, we compare the tested detectors by their speed and the number of detected interest points. The comparison is performed on a set of videos from the Hollywood-2 dataset with spatial resolution of 360×288 pixels (i.e., half resolution) and about 8000 frames length in total. The run-time estimates are obtained on a Dell Precision T3400 Dual core PC with 2.66 GHz processors and 4GB of RAM. Table 3.6 presents results for the three detectors and dense sampling in terms of frames per second and average number of features per frame. Note that feature computation is included in the run time. Among the detectors, Gabor extracts the densest features (44 features/frame) and it is the slowest one (0.9 frames/second). Hessian3D extracts the sparsest features (19 features/frame) and is consequently the most efficient (4.6 frames/second). As for the dense sampling, since there is no feature detection as such, the overall computational time is mainly spent on the feature description. Obviously, dense sampling extracts many more features than interest point detectors. Note that the time of descriptor quantization is not taken into account in this evaluation.

3.4 Discussion

This chapter presents a comprehensive evaluation and comparison of several local space-time detectors and descriptors under a common bag-of-features based action recognition framework. Among the main conclusions, we note that dense sampling consistently outperforms all the tested interest point detectors in realistic video settings, but performs worse on the simple KTH-Actions dataset. This indicates both (i) the importance of using realistic experimental video data as well as (ii) the limitations of current interest point detectors. We argue that the choice of sparse detectors seems to be less important as their performance is often similar. On the contrary, the introduced dense features consistently outperform sparse feature detectors. Note, however, that dense sampling also produces a very large number of features (usually 15-20 times more than feature detectors). This is more difficult to handle than the relatively sparse number of interest points. Across the datasets, Harris3D performs better on KTH-Actions dataset, while the Gabor detector gives better results for UCF-Sports and Hollywood-2 actions datasets.

Among the tested descriptors, the combination of gradient based and optical flow based descriptors performs relatively better. The combination of dense sampling with the HOG/HOF descriptors provides the best results for the most challenging Hollywood-2 dataset. On the UCF-Sports dataset, the HOG3D descriptor performs the best in combination with dense sampling.

Chapter 4

Bag-of-Features with non-local cues

Contents

In the previous chapter, local space-time features integrated within a Bag-of-Features (BoF) video representation have been shown to provide promising results for action recognition in realistic video data. Local features and descriptors, however, are often ambiguous, implying their limited discriminative power and sub-optimal performance in action recognition. For instance, Figure 4.1 shows matching of local features in pairs of video sequences. As can be seen, local features alone may not always provide sufficient information for correct matching of similar events. Therefore, in this chapter, we propose to disambiguate local space-time features and to improve action recognition by integrating additional non-local cues within the BoF representation.

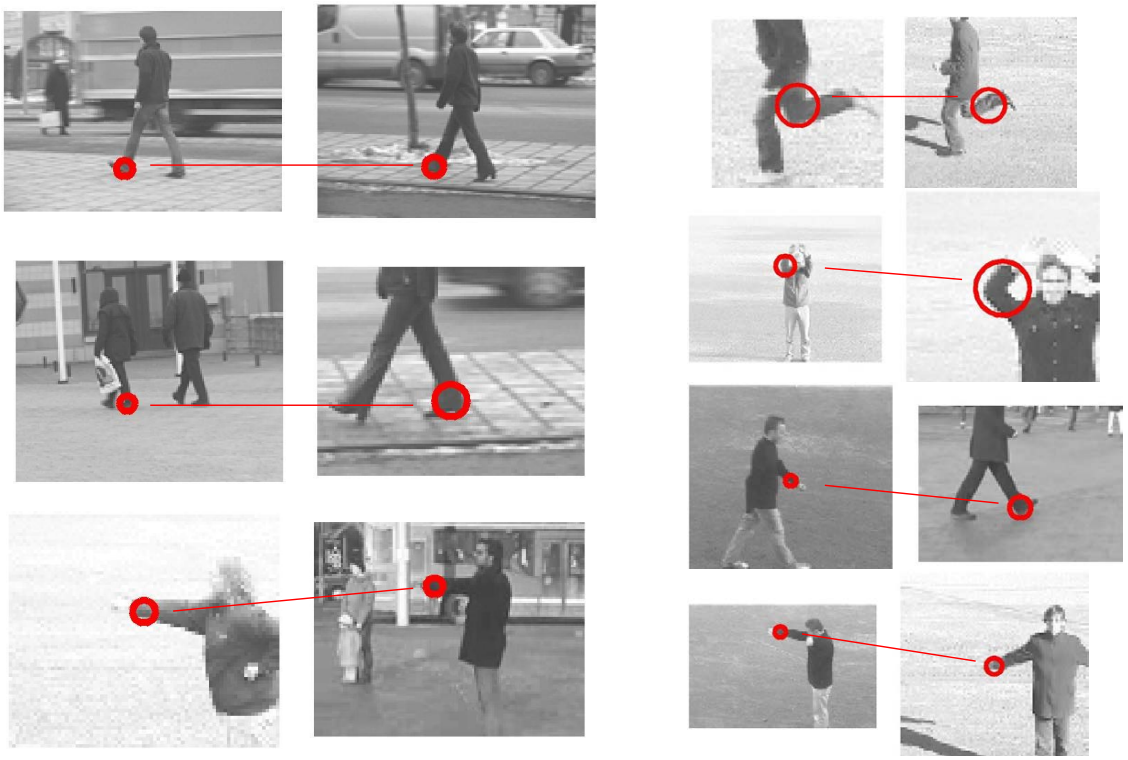


Figure 4.1: **Illustration of local feature matches.** While local features often provide correct matching of events in video, pure local information is not always sufficient to separate semantically different events; e.g., the two examples in the bottom-right are incorrect matches. Such ambiguities occur due to local similarity of different events in shape and motion (figure courtesy of Ivan Laptev).

We argue that a video is mostly comprised of certain semantic regions. For instance, the video illustrated in Figure 4.2 can be divided into three regions, namely parking lot, road and side walks. We believe that decomposing a video into such regions can be helpful in disambiguating local space-time features. For example, the regions of a *parking lot* and *side walks* in Figure 4.2 are likely to correlate with specific actions such as *opening a trunk* and *running*. Propagating region labels to the local feature level in this example is therefore expected to increase discriminative power of local features with respect to particular actions.

To decompose a video into region classes, we in this chapter, resort to multiple and readily-available segmentation methods. In particular, we investigate unsupervised and supervised video segmentation using (i) motion-based foreground separation, (ii) person detection, (iii) static action detection and (iv) object detection. While such segmentation methods might be imperfect, they provide complementary region-level information to

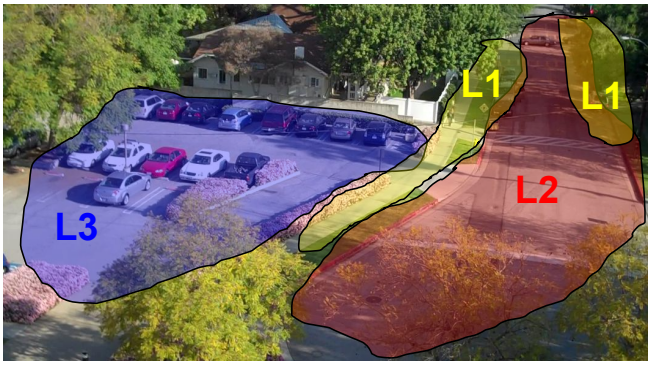


Figure 4.2: Regions in video such as road, side walk and parking lot frequently co-occur with specific actions (e.g., driving, running, opening a trunk) and may provide informative priors for action recognition.

local features. Moreover, segmentation methods trained on additional training data (e.g., person and object detection) introduce additional supervision into our extended BoF framework and potentially increase its discriminative power. We furthermore, employ the ERC-Forest (described in Section 2.1.1) approach to learn supervised visual vocabulary, aiming to introduce more supervision into our extended BoF framework, to further improve action recognition performance.

Using different types of regions, we construct alternative video representations from the original set of local spatio-temporal features. We moreover, exploit complementarity of such representations and combine them within a multi-channel SVM framework [193]. We evaluate our method on the challenging Hollywood-2 human actions dataset [111] and demonstrate significant improvement with respect to the state-of-the-art.

The rest of the chapter is organized as follows. Section 4.1 describes the proposed extension in the BoF framework. Section 4.2 presents details of alternative video segmentation methods used. Section 4.3 presents results while section 4.4 concludes the chapter with a discussion.

4.1 Extended BoF representation

Our baseline BoF framework is essentially the same as presented in Section 3.2. We compute the BoF representation using the Harris-3D feature points [88] together with the HOG/HOF descriptors [91], and use k-means for visual vocabulary.

While k-means is a simple and *unsupervised* approach to construct visual vocabularies, previous methods (e.g., [45, 119]) have attempted to improve image classification

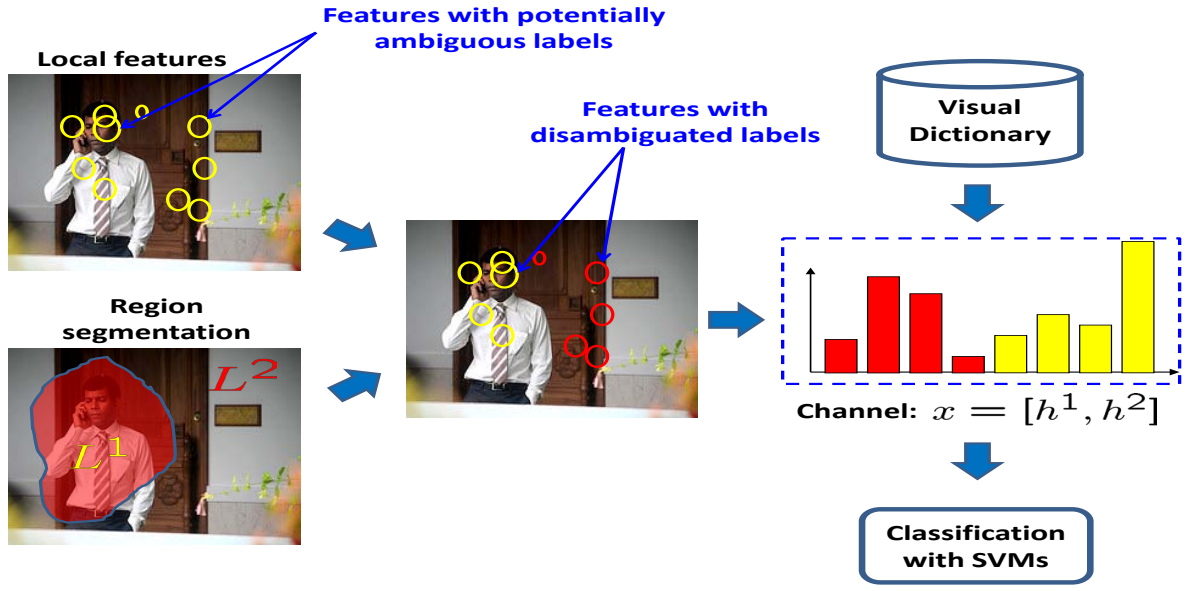


Figure 4.3: An illustration of our approach to disambiguate local descriptors with the help of semantic video segmentation.

tasks by constructing *supervised* visual vocabularies. One of such approaches is the *Extremely Randomized Clustering Forest* (ERC-Forest) by Moosmann *et al.* [119] (refer to Section 2.1.1 for more detail). Here, we use ERC-Forest to construct supervised visual vocabularies, aiming to improve action recognition in realistic video data. We construct $M = 5$ multiple trees with 1000 leaf nodes each, and assign M labels to each feature descriptor according to each tree. In this case, the resulting histogram of feature labels corresponding to M trees, is used as the final video representation. We demonstrate in Section 4.3.1 that the supervised ERC-Forest outperforms the unsupervised k-means on the challenging Hollywood-2 actions dataset.

We propose to extend the BoF representation (presented in Section 2.1.1) and to decompose video into a set of regions r assigned to labels l, $l \in \{L^1, \ldots, L^M\}$. A separate BoF histogram h^i is accumulated from quantized features within all regions with labels L^i. Following the terminology of [94], a video signature, i.e., a *channel* is then constructed by concatenating BoF histograms for all region labels, i.e., $x = [h^1, \ldots, h^M]$ as illustrated in Figure 4.3. In this chapter, we investigate different types of channels obtained with alternative video segmentation methods described in section 4.2.

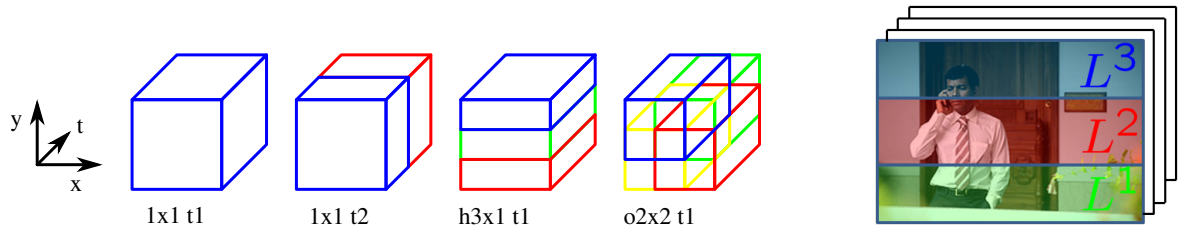


Figure 4.4: (Left) examples of spatio-temporal grids [91], (right) illustration of video decomposition according to $h3 \times 1$ $t1$ grid.

4.2 Video segmentation

In this section, we describe alternative methods for decomposing video into region classes, thereby providing means for disambiguating local Harris3D features.

4.2.1 Spatio-temporal grids

Spatio-temporal video grids are introduced in [91] and show promising results for action recognition. The basic idea is to divide a video into a set of predefined spatio-temporal regions. We follow the same approach and define 24 different spatio-temporal grids. Each of these 24 grids divides a video in up to M = 27 regions with unique region labels. The feature histograms corresponding to each spatio-temporal grid region are then concatenated into one vector and normalized to make a channel. Spatially, we use a 1×1 grid (corresponding to the standard BoF representation), a 2×2 grid, a horizontal $h3 \times 1$ grid, a vertical $v1 \times 3$ grid, a denser 3×3 grid and a center-focused $o2 \times 2$ grid where neighboring cells overlap by 50% of their width and height. Temporally, a video sequence is divided into 1 to 3 non-overlapping temporal bins, resulting in $t1$, $t2$ and $t3$ binnings, where $t1$ represents the standard BoF approach. There is also a center-focused $ot2$ grid. In the following, we refer to these 24 spatio-temporal grid channels as *STGrid-24*. Figure 4.4 illustrates some of the grids which show good performance in [91].

4.2.2 Foreground/background motion segmentation

Segmenting local descriptors based on the foreground (FG) and background (BG) motions in video can be valuable in order to separate foreground features which are more likely to belong to the action from background features which can help action recognition by

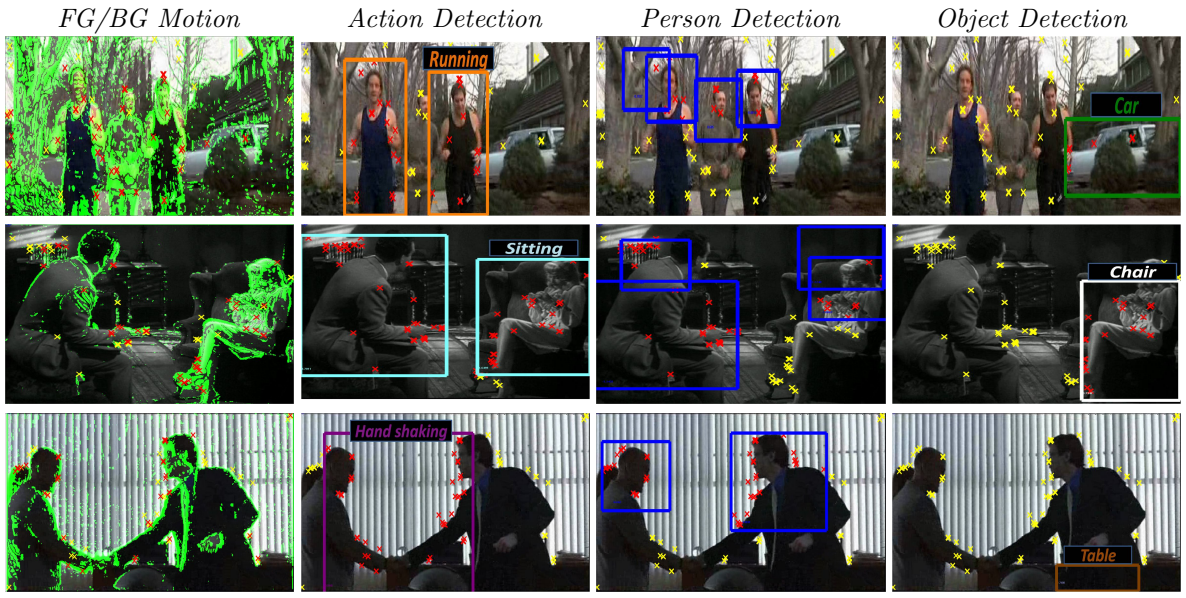


Figure 4.5: **Illustration of proposed semantic region extraction in video according to (from left to right): motion region segmentation, action detection, person detection and object detection**. Correct segmentation separates local features into meaningful groups denoted by yellow and red crosses. We also illustrate failures of automatic segmentation due to false negative detections (see e.g., missed running action in the first row) and false positive detections (see e.g., incorrect table detection in the third row).

capturing scene context. We use the Motion2D library [124] [1] to estimate 2D parametric motion model in a video sequence. We then threshold (with four threshold values: 127, 150, 170, 200) the motion estimations and generate FG/BG masks. We use these masks to segment local descriptors into FG and BG classes. Figure 4.5 (1st column) shows the FG masks in green together with the segmented features. By separating features and building feature histograms according to FG and BG regions as well as for four different threshold values, we obtain 8 channels. We refer to these eight channels as *Motion-8*.

4.2.3 Action detection

The ability to localize action in a video can be helpful in separating action specific descriptors. Of course, all the remaining descriptors that belong to the background of action, can form another complementary channel by capturing the context information.

The idea is to train an action specific detector on still images collected from the Internet and perform action detections on the Hollywood-2 video sequences. Depending upon

[1] Available at: `http://www.irisa.fr/vista/Motion2D`

the availability of sufficient amount of action samples on the Internet, we investigate the idea for the following action classes: answering phone, hugging, hand shaking, kissing, running, eating, driving a car, and sitting on sofa/chair. The last class corresponds to the action classes: *sitting down, standing up,* and *sitting up*. Figure 4.6 presents sample images collected from the Internet. We train Felzenszwalb's object detector [40] for each action class (using 100-170 positive and approximately 9000 negative images for training) and run detector on the frames of Hollywood-2 videos (see Figure 4.5, 2nd column). The returned bounding boxes segment video into FG/BG corresponding to action/non-action regions. We then perform the following steps:

1. Threshold bounding boxes with six threshold values θ and divide each corresponding FG region into a 1×1 or 2×2 grid.

2. Compute 12 channels for six threshold values and two types of grid, i.e., $x_{\theta,1\times1} = [h^1, h^2]$ and $x_{\theta,2\times2} = [h^1, h^2, h^3, h^4, h^5]$.

We refer to the 12 obtained channels as *Action-12* for each of the eight aforementioned action classes.

4.2.4 Person detection

Separation of local descriptors on the basis of person/non-person region segmentation not only helps to disambiguate them but also provide a compact BoF representation for an action (as actions are related to persons). We use the Calvin upper-body detector [2] which is a combination of the Felzenszwalb's object detector [40] and the Viola-Jones' face detector [174]. This detector returns bounding boxes fitting the head and upper half of the torso of the person (see Figure 4.5, 3rd column), which segment video into FG/BG corresponding to person/non-person regions. Following the steps of Section 4.2.3, we generate 12 channels. We refer to these channels as *Person-12*.

4.2.5 Object detection

Objects can provide a valuable context information in recognizing actions in video. For instance, the object *car* can be helpful to recognize the actions *driving a car* and *getting out of a car*, and the objects *chair* and *sofa* can be helpful for the classes *sitting down* and *standing up*. We investigate this concept by using Felzenszwalb's object detectors

[2] Available at: `http://www.vision.ee.ethz.ch/calvin/calvin_upperbody_detector`

AnswerPhone DriveCar Eat HandShake HugPerson Kiss Run Sitting

Figure 4.6: Sample images collected from the Internet used to train the action detectors.

Channels	Performance (mean AP)
BoF with k-means	0.481
BoF with ERC-Forest	**0.482**
STGrid-24 with k-means	0.509
STGrid-24 with ERC-Forest	**0.525**

Table 4.1: Classification performance of the baseline channels in the Hollywood-2 dataset [111].

[40] [3] on the following object classes: *car, chair, table* and *sofa*, and perform separate detections on the Hollywood-2 sequences (see Figure 4.5, 4th column). The returned bounding boxes divide video into FG/BG corresponding to object/non-object regions. Again following the steps of Section 4.2.3, we compute 12 channels per object class. We refer to the corresponding 12 channels for each object class as *Objects-12*.

4.3 Experiments

For action classification, we follow the evaluation setup proposed in Section 3.2 and use a non-linear SVM with χ^2 kernel. To investigate combination of different video channels, we use the multi-channel kernel [193], which is presented in Appendix A. All the experiments are performed on the Hollywood-2 actions dataset (see Section 2.4.4 for details).

4.3.1 Baseline performance

To get a baseline, we perform experiments with (i) the standard BoF method, and (ii) STGrid-24 channels using the k-means as well as ERC-Forest generated visual vocabularies. ERC-Forests have been previously used for image classification tasks (e.g.,

[3] We use object detectors trained by the authors on the PASCAL VOC 2008 dataset.

Video channels	Performance (mean AP)
Motion-8	0.503
Person-12	0.496
Objects-12	0.490
Action-12	**0.526**
STGrid-24 + Motion-8	0.533
STGrid-24 + Person-12	0.535
STGrid-24 + Objects-12	0.530
STGrid-24 + Action-12	**0.560**
STGrid-24 + Motion-8 + Action-12 + Person-12 + Objects-12	**0.553**

Table 4.2: Overall performance of individual channels and their different combinations.

[119, 122, 96]), and here we want to evaluate ERC-Forest [119] for action recognition in realistic video data. Table 4.1 compares their mean average precisions. It turns out that STGrid-24 channels improve upon the standard BoF approach, which is consistent with the findings in [91]. Moreover, the performance improvement in BoF with ERC-Forest is marginal, whereas, an improvement of about 2% is observed in the case of STGrid-24. Therefore, in the rest of this chapter, we only present results obtained with the supervised ERC-Forest vocabulary. Note that our baseline result for BoF with k-means (mAP 0.481) is comparable to the best result (mAP 0.476) previously reported on this dataset in [176].

4.3.2 Improvements with channel combination

The performance by STGrid-24 channels (0.525 mAP) serves as a strong baseline result here. Table 4.2 (1st portion) reports results for the new channels (introduced in Section 4.2), with Action-12 channels having the highest mAP (i.e., 0.526). While most of our new channels do not outperform the baseline, the advantage of all new channels becomes apparent when combined with the baseline STGrid-24 channels. As can be seen from Table 4.2, new channels combined with STGrid-24 not only improve upon their individual performance but also improve the baseline result up to 0.560. This can be explained by the complementarity of channels, adding different information to the BoF representation. Note, however, that the integration of Action-12, Person-12 and Objects-12 channels implies the use of additional training data which makes the corresponding results not directly comparable to previous results reported on Hollywood-2 dataset. By combining all the four new channels with STGrid-24 channels, we obtain 0.553 mAP, which is a

Channels	BoF	STGrid-24 (Baseline)	Action-12	STGrid-24 +Action-12	STGrid-24 +Motion-8 +Action-12 +Person-12 +Objects-12
mean AP	0.482	0.525	0.526	**0.560**	0.553
AnswerPhone	0.157	0.259	0.207	**0.299**	0.248
DriveCar	0.874	0.859	0.869	0.865	**0.881**
Eat	0.548	0.607	0.574	0.593	**0.614**
FightPerson	0.739	0.749	0.758	0.760	**0.765**
GetOutCar	0.331	0.447	0.383	0.457	**0.473**
HandShake	0.200	0.285	0.457	**0.497**	0.383
HugPerson	0.378	**0.461**	0.408	0.452	0.446
Kiss	0.516	0.569	0.552	0.590	**0.615**
Run	0.711	0.698	0.732	0.719	**0.743**
SitDown	0.594	0.589	0.595	**0.625**	0.613
SitUp	0.207	0.202	0.227	**0.275**	0.250
StandUp	0.533	0.574	0.556	0.588	**0.604**

Table 4.3: Per-class AP performance by different channels/channel-combinations.

significant improvement over the baseline (0.525 mAP). We also note that the channel combination STGrid-24+Action-12 (0.560 mAP) slightly outperforms the combination of all channels. This behavior highlights the need for more sophisticated methods for kernel combination compared to the simple multi-channel approach (product of kernels) considered in this work. We have tried learning kernel combination using Multiple Kernel Learning (MKL) framework [141]. However, similar to the previous findings [49], MKL did not improve results in our case.

In table 4.3, we present per-class average precision values corresponding to the baseline channels as well as the best performing new channels and their combinations. We note improvement of eleven out of twelve action classes (APs are marked in bold in the last two columns) when combining new channels with the baseline channels. Distribution of the best class APs across three columns (corresponding to different channel(s)) points out the need to devise some sophisticated technique for class-specific channel(s) selection. Moreover, although the mean AP performance by the final channel combination (i.e., 0.553) is slightly lower than that by the STGrid-24+Action-12 channels (i.e., 0.560), yet it achieves the best results for seven action classes (APs are marked in bold in the last column).

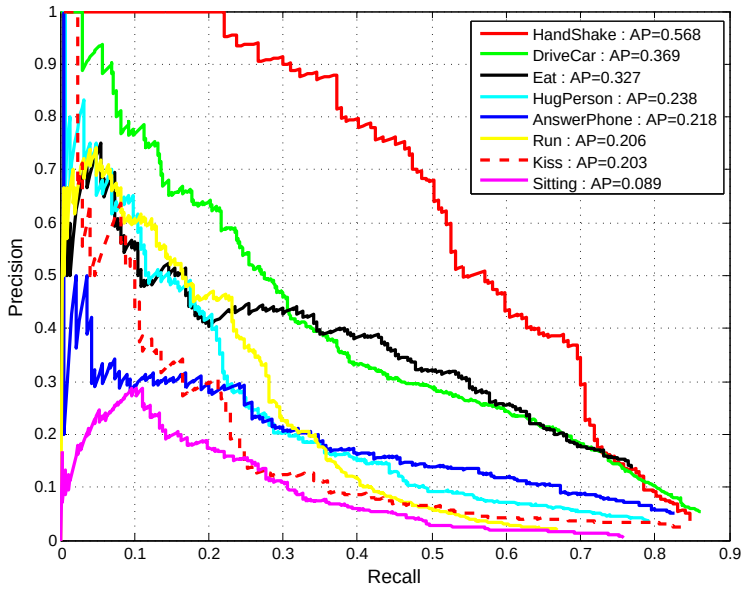


Figure 4.7: Detection performance by the eight action detectors on a subset of Hollywood-2 test sequences.

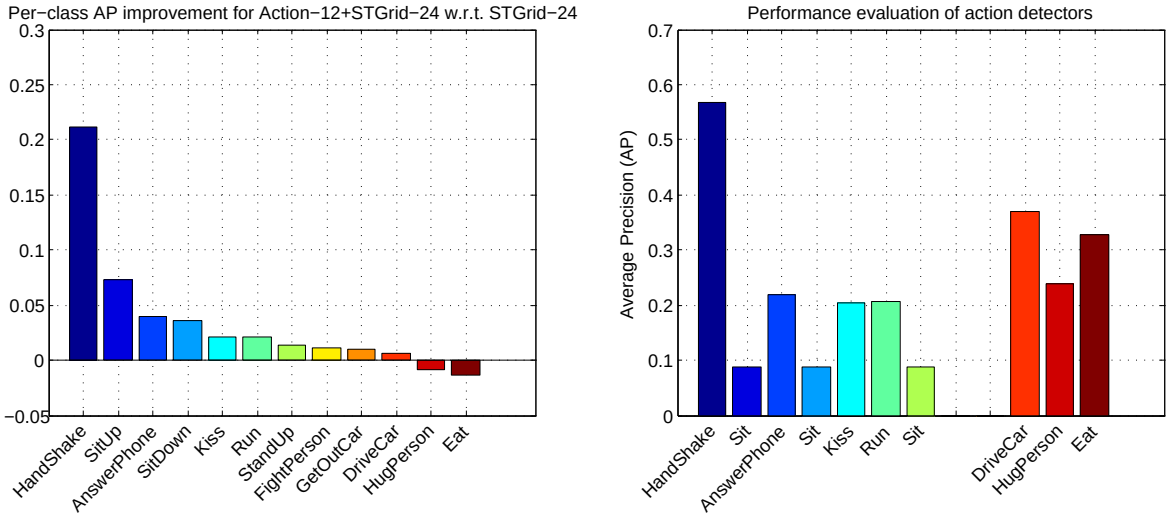


Figure 4.8: (Left) per-class AP improvement by STGrid-24+Action-12 channels compared to the baseline STGrid-24 channels, and (right) performance by the corresponding action detectors on a subset of Hollywood-2 test sequences.

The performance by Action-12 channels has been relatively impressive, individually (i.e., 0.526 mAP) as well as in combination with the baseline STGrid-24 channels (i.e., 0.560 mAP). This observation draws our attention to the combination STGrid-24+Action-12. Figure 4.8 (left) illustrates the relative improvement in each action class by the STGrid-24+Action-12 channels compared to the baseline STGrid-24 channels. We observe significant improvement for certain classes (e.g., HandShake and SitUp), whereas marginal or no improvement for others (e.g., HugPerson and Eat). This variation in per-class improvement requires further investigation of our proposed Action-12 channel. For this purpose, we evaluate our trained action detectors on a subset of Hollywood-2 test sequences. We annotate about 200-500 positive frames corresponding to each class with ground truth bounding boxes. For the negative subsets, we randomly select 1000 frames separately for each action class, without having any instance of the target action class. As per PASCAL VOC 2007 [36], we consider a detection to be a true positive if it overlaps at least 50% with the ground truth bounding box. Figure 4.7 presents the precision-recall (PR) curve for each action detector. Figure 4.8 (right) presents the average precision for each action detector (note that the 'Sit' detector is used for three action classes namely SitUp, SitDown, and StandUp). This evaluation sheds some light on the relative performance of our trained detectors and their effect on recognition performance by the Action-12 channel. For instance, the best performing 'HandShake' detector (Figure 4.8 (right)) achieves the highest performance gain for the action class HandShake (Figure 4.8 (left)).

4.4 Discussion

This chapter presents an extension to the standard BoF approach for classifying human actions in realistic videos. The main idea is to disambiguate local features that represent different events but cannot be distinguished based on local information alone. As we show experimentally, this separation helps to get significant improvement over the strong baseline. The proposed framework also enables introduction of additional supervision into BoF action classification in the form of region detectors that could be trained on related tasks. The method thus provides the flexibility to utilize additional training data (such as on-line images, PASCAL VOC images, etc.) to mitigate the problem of having limited training data, as is with the Hollywood-2 dataset.

Chapter 5

Attribute Bank for action recognition

Contents

In the previous Chapter 4, our focus has been to improve the discriminative power of local features by disambiguating them through region-level information in video. In particular, we employ pre-trained object and action detectors to segment video into spatial regions with different semantic meanings. Recently, Li *et al.* [98, 99] propose a somewhat different approach for scene classification in images. Their idea is to apply a large number of pre-trained generic object detectors (e.g., water, sky, boat, bear, etc.) on an input image at multiple scales. The response map for each object is max-pooled, and the corresponding maximum response values are concatenated into a vector representation. The proposed *Object Bank* representation has been shown to capture high-level semantics from scene images, and offers complementary information to low-level features.

Moreover, attribute-based representations have shown promising results for object as well as scene recognition in recent few years [43, 83, 85, 39, 165]. Their success is primarily owing to the notion of 'attribute', a high-level semantically meaningful representation.

In attribute-based methods for object recognition, an object is represented by using visual attributes. For instance, a zebra can be described as an object having texture of black/brown and white lines, and associated paws. Such visual attributes summarize the low-level features into object parts and other properties, and are then used as the building blocks for recognizing the object. In a parallel work to ours, Liu *et al.*[102] propose to represent video by visual attributes for human action recognition. Their framework employs manually specified attributes (such as *translation motion, arm pendulum-like motion, torso twist, having stick-like tool,* etc.), wherein attributes are discriminatively selected for each action class in a latent SVM [40] framework. Moreover, they augment their manual attributes with data-driven attributes, which are automatically inferred from the training data. Their method achieves promising results on Olympic-Sports [120] and UIUC [169] datasets. Likewise, we argue that a video representation based on characteristic visual attributes would be very useful in high-level action recognition task. The choice of a representative set of attributes depends on the target dataset, and may include any semantically meaningful concept in video.

In contrast to the work by Liu *et al.*[102], we in this chapter, propose a simple yet effective approach. We consider a diverse range of attributes which include objects (like car, chair, table, etc.), static actions, persons as well as discriminative poses. Our framework employs a pre-trained classifier for each attribute, trained on a large number of static images. Following the Object Bank approach, we apply all the classifiers on individual frames at multiple scales. For each attribute, we compute the maximum response value from the resulting space-time filter map. The final video representation is the concatenation of maximum response value of each attribute classifier. We refer to this as the *Attribute Bank* representation. We evaluate the Attribute Bank representation on the Hollywood-2 dataset, and demonstrate that it provides complementary information to that of the low-level features. Moreover, the Attribute Bank representation is vocabulary free and thus straightforward to compute.

The rest of the chapter is organized as follows. Section 5.1 describes the Attribute Bank representation in detail. Section 5.2 then employs the proposed high-level representation for human action recognition, and presents empirical results on the Hollywood-2 dataset. Finally, Section 5.3 concludes the chapter with a discussion.

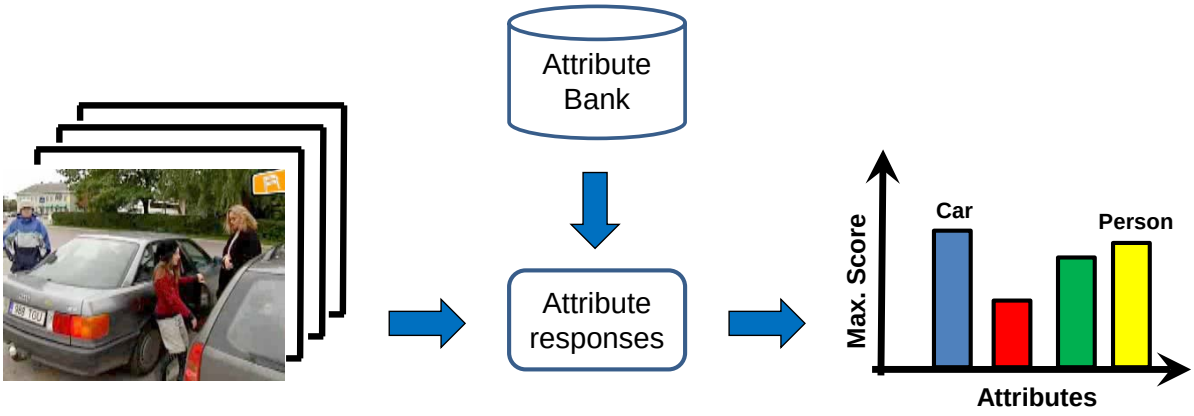


Figure 5.1: **Illustration of the Attribute Bank framework**. A range of attribute classifiers is applied on a video sequence, and the maximum response value corresponding to each attribute classifier is subsequently concatenated into a vector representation (refer to the text for further details).

5.1 The Attribute Bank representation

The Attribute Bank framework is comprised of a set of attribute classifiers. The considered attribute classifiers are trained to predict the presence of objects and people as well as characteristic static actions and poses. Section 5.1.1 explains the video encoding process, given a set of pre-trained attribute classifiers. Section 5.1.2 then details the set of attribute classifiers employed in the Attribute Bank framework.

5.1.1 Attribute filter based encoding

Given a video sequence v, an attribute filter response volume Ω_{a_k} is obtained by estimating the occurrence probability $p(a_k|v)$ for the attribute classifier a_k at multiple scales. Let n be the total number of attribute classifiers. We use maximum pooling on the resulting n response volumes, and concatenate the maximum score of each attribute classifier a_i into a vector representation:

$$\left[\max_{(x,y,t)} \Omega_{a_1}, \ldots, \max_{(x,y,t)} \Omega_{a_n}\right], \tag{5.1}$$

where (x, y, t) denotes the spatio-temporal volume for the max-poolilng, which in this case is the whole video, as illustrated in Figure 5.1. Moreover, we use the 24-level spatio-temporal grids [91], and divide each response volume Ω_{a_i} into 24 different types of grids. Each grid divides a response volume into a set of pre-defined grid-cells (see

Figure 5.2: **Illustration of poselets**. Poselets are part-based detectors which operate on novel body parts, and are invariant to distracting visual variations in images. The figure shows positive examples for some of the poselets: *frontal face*, *right arm crossing torso*, *pedestrian*, *right profile and shoulder*, and *legs frontal view* (figure reprinted from [17]).

Section 2.3.3 for the description about each grid type). For each grid with m cells, the corresponding video representation is the concatenation of attribute features in each grid cell c:

$$\left[\max_{(x,y,t)_c} \Omega_{a_1}, \ldots, \max_{(x,y,t)_c} \Omega_{a_n}\right]_{c=1}^{m}. \tag{5.2}$$

Consequently, a video sequence is encoded into 24 different grid channels, which are referred to as the Attribute Bank representation.

5.1.2 Attribute classifiers for the Attribute Bank

We have considered a diverse range of attributes for the proposed Attribute Bank representation. We use the latent SVM classifiers [40] (presented in Chapter 4) trained for the four object classes (*car*, *chair*, *table*, and *sofa*), and eight action classes (*answering phone*, *hugging*, *hand shaking*, *kissing*, *running*, *eating*, *driving*, and *sitting*). Additionally,

we use the Calvin upper-body detector[1] to detect the *person* attribute in videos. The Attribute Bank representation based on the aforementioned object, action, and person attributes is referred to as the *OAP-Bank* channels.

Furthermore, we use 150 different types of *poselet* as attributes (see Figure 5.2). Bourdev and Malik [17] have recently introduced a novel representation of a human body part, which they refer to as a *poselet*. Poselets are part based detectors and operate on novel body parts. These specialized detectors have been trained on a relatively large image dataset of manually annotated body parts, and invariant to distracting variations in visual appearance of images. Poselets have been demonstrated to be effective for detection, segmentation, pose estimation, as well as action recognition in still images [17, 16, 20, 61, 109]. Here, we propose to compute the Attribute Bank representation with 150 different types of poselets as attributes. We refer to these video channels as *Poselet-Bank*.

5.2 Action recognition with Attribute Banks

The Attribute Bank representation is an attempt to build high-level features for human action recognition in video. Thanks to the recent development of more robust object and body-part detectors [40, 17], we are able to describe video with high-level semantically meaningful features. Such features are not meant to replace low-level features. Instead, we observe that these features provide important complementary information from video sequences.

For action classification using the proposed OAP-Bank as well as the Poselet-Bank channels, we use a non-linear SVM with RBF kernel. As a strong baseline, we use the STGrid-24 channels (based on Harris3D and HOG/HOF features, and described in Section 4.2.1), and employ a non-linear SVM with χ^2 kernel for classification. Moreover, we combine the different video channels using the multi-channel kernel [193] (i.e., product of kernels, see Appendix A), and use *one-against-rest* approach for multi-class classification.

5.2.1 Experiments

We evaluate the performance of our Attribute Bank representation on the task of action recognition in the challenging Hollywood-2 dataset. Table 5.1 presents the results for the baseline STGrid-24 channels as well as for our proposed Attribute Bank based

[1] Available at: http://www.vision.ee.ethz.ch/calvin/calvin_upperbody_detector

Channels	STGrid-24 (Baseline)	OAP-Bank	OAP-Bank + STGrid-24	Poselet-Bank	Poselet-Bank + STGrid-24	OAP-Bank + Poselet-Bank + STGrid-24
mean AP	0.525	0.413	0.558	0.344	0.541	**0.571**
AnswerPhone	0.259	0.347	0.360	0.230	0.292	**0.366**
DriveCar	0.859	0.694	0.880	0.571	0.876	**0.881**
Eat	**0.607**	0.248	0.580	0.243	0.533	0.564
FightPerson	**0.749**	0.482	0.733	0.282	0.695	0.705
GetOutCar	0.447	0.307	0.426	0.303	0.438	**0.457**
HandShake	0.285	0.471	0.512	0.392	0.433	**0.523**
HugPerson	**0.461**	0.283	0.420	0.136	0.406	0.407
Kiss	0.569	0.521	**0.668**	0.398	0.600	0.665
Run	0.698	0.577	0.700	0.649	**0.767**	0.762
SitDown	**0.589**	0.366	0.556	0.381	0.573	0.566
SitUp	0.202	0.193	0.244	0.138	0.288	**0.334**
StandUp	0.574	0.473	**0.617**	0.404	0.596	0.616

Table 5.1: Per-class average precision (AP) performance of different channels/channel-combinations on the Hollywood-2 dataset.

channels and their combinations. We can observe that the individual performance of the OAP-Bank (i.e., 0.413 mAP) and Poselet-Bank (i.e., 0.344 mAP) channels are lower than that of the baseline STGrid-24 channels (i.e., 0.525 mAP). However, when the OAP-Bank and Poselet-Bank channels are combined with the baseline STGrid-24 channels, they yield a performance improvement of about 3% and 2% respectively, over the baseline. The superior performance by the OAP-Bank channels compared to the Poselet-Bank channels is probably due to the fact that the former encode the presence/absence of the specific actions (*answering phone*, *hugging*, *hand shaking*, *kissing*, *running*, *eating*, *driving*, and *sitting*), which are directly related to the action classes in the Hollywood-2 dataset. Moreover, the OAP-Bank channels capture the information about different objects (*car*, *chair*, *table*, and *sofa*), which also helps to discriminate among the action classes.

Furthermore, when the OAP-Bank and Poselet-Bank channels are both combined with the baseline STGrid-24 channels, we obtain an improvement of 4.6% over the baseline. We can see that our Attribute Bank based channels help to improve eight out of tweleve action classes (average precisions are marked in bold). It demonstrates that the proposed Attribute Bank representation which captures high-level information in video, is actually very discriminative. Moreover, the Attribute Bank features are shown to enrich the low-level features by combining the complementary high-level information in video.

5.3 Discussion

Our proposed Attribute Bank representation is based on a range of characteristic attributes (e.g., objects, specific actions, discriminative poses) in video and thus captures the high-level information therein. The Attribute Bank representation is composed from object and body-part classifiers, which have been trained on large number of images. Empirical results show that our Attribute Bank features are discriminative, and offer complementary high-level information to the low-level features.

The importance of context in visual recognition tasks has been demonstrated by several works (e.g., [139, 64, 111]), over the recent few years. Broadly speaking, context can be grouped into two categories: (a) co-occurrence context, and (b) geometric context. The former encodes the probability of co-occurrences of objects, actions, and scenes etc., whereas, the latter takes into account the layout of scenes and constraints of camera(s). The Attribute Bank representation implicitly encodes the co-occurrence context by concatenating the response maps of different attribute classifiers. Note that our Attribute Bank is comprised of a modest set (162 in total) of attributes, and yet achieves quite promising improvement (4.6% over the strong baseline). We intuitively argue that the performance could be further improved with the inclusion of more related attributes (e.g., based on more objects, color, texture, indoor/outdoor scene, etc.). Moreoever, we have included only weak geometrical information in the Attribute Bank representation through the coarse spatio-temporal grids [91]. More sophisticated and robust geometrical information, such as scene layout and depth information, may further improve the performance.

One important concern, however, is the computational complexity. Training thousands of attribute classifiers could be expensive. While obtaining an increasingly large number of detectors is becoming more and more viable with the emergence of large-scale datasets (e.g., LabelMe [151] and ImageNet [30]), there is still an *attribute filtering* step in the pipeline. The conventional naive scanning window approach hinders using a large number of attribute classifiers in video data. Efficient algorithms such as robust branch and bound scheme proposed by Lampert *et al.* [84] can be employed to speed up the computation time.

Chapter 6

Action-characteristic local motion descriptors

Contents

The recent success of local Bag-of-Features based methods is owing to their robustness to some variations in appearance and motion. Nevertheless, significant changes of view points and appearance affects local descriptors and, thus, introduces distraction to local representations. To address this problem, we in this chapter, propose a supervised approach to learn local motion descriptors from a large pool of annotated video data. The main motivation behind this method is to construct action-characteristic representations of body-joints undergoing specific motion patterns while learning invariance with respect to changes in camera views, lighting, human clothing, and other factors. In terms of the taxonomy proposed by Moeslund *et al.* [118] (and adopted in Section 1.1), such action-characteristic local motion descriptors represent *action primitives*.

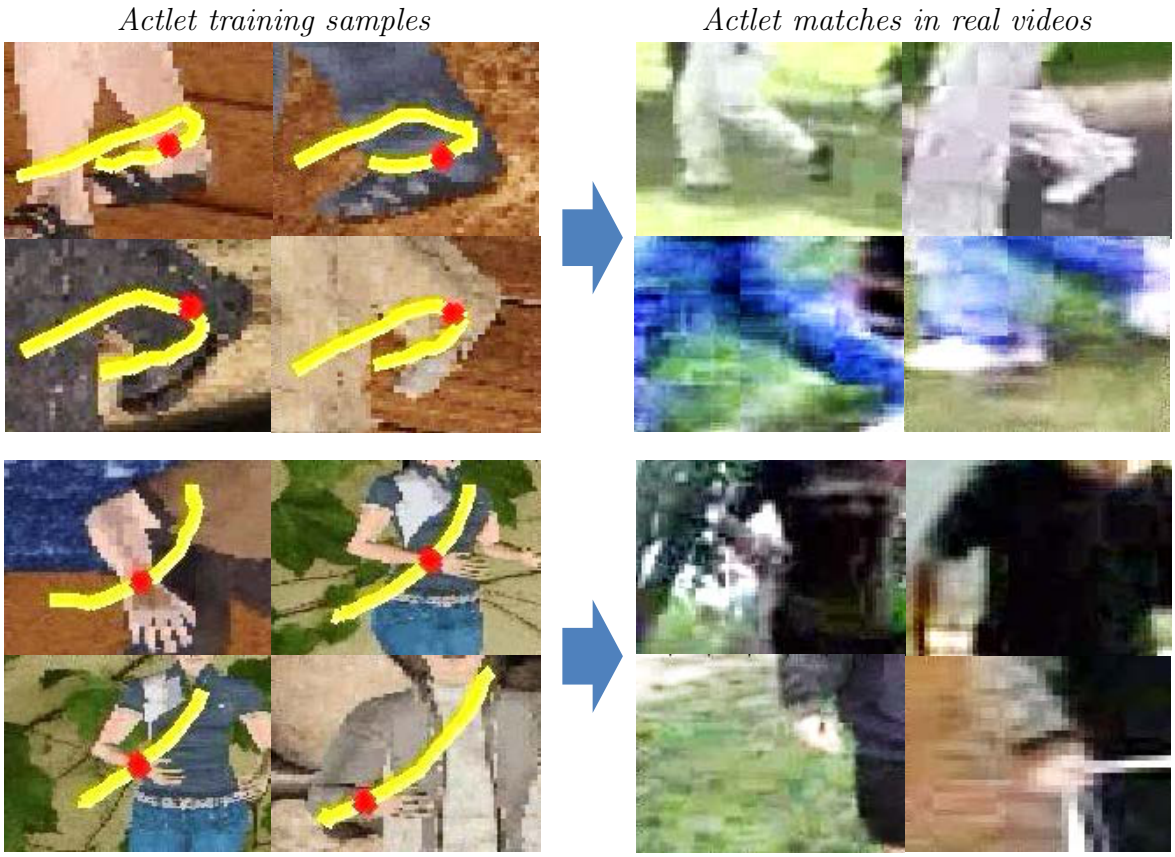


Figure 6.1: **Illustration of Actlets**. Actlets are specialized detectors which are trained on synthetic data (left) and localized on the real videos (right). The automatically annotated trajectories of body-joints are shown on the left.

Recently, Bourdev *et al.* [17] have proposed a supervised approach to learn appearance of body parts in static images. Body part detectors called *Poselets* are trained to be invariant to irrelevant appearance variations using manual annotation of body parts in training images. Inspired by this representation, we in this chapter, propose a supervised approach to learn *Actlets*, i.e., detectors of body parts undergoing specific patterns of motion. Learning Actlets requires a substantial amount of annotated training data. To collect such data, we propose to avoid the heavy burden of manual video annotation and generate annotated data automatically by synthesizing videos of avatars driven by the motion-capture data (see Figure 6.1). We next successfully employ Actlets for human action recognition in realistic video data. We evaluate our method and demonstrate its significant improvement as well as complementarity to BoF representation on the challenging UCF-Sports and YouTube-Actions datasets.

The rest of the chapter is organized as follows: Section 6.1 presents details of the synthetic

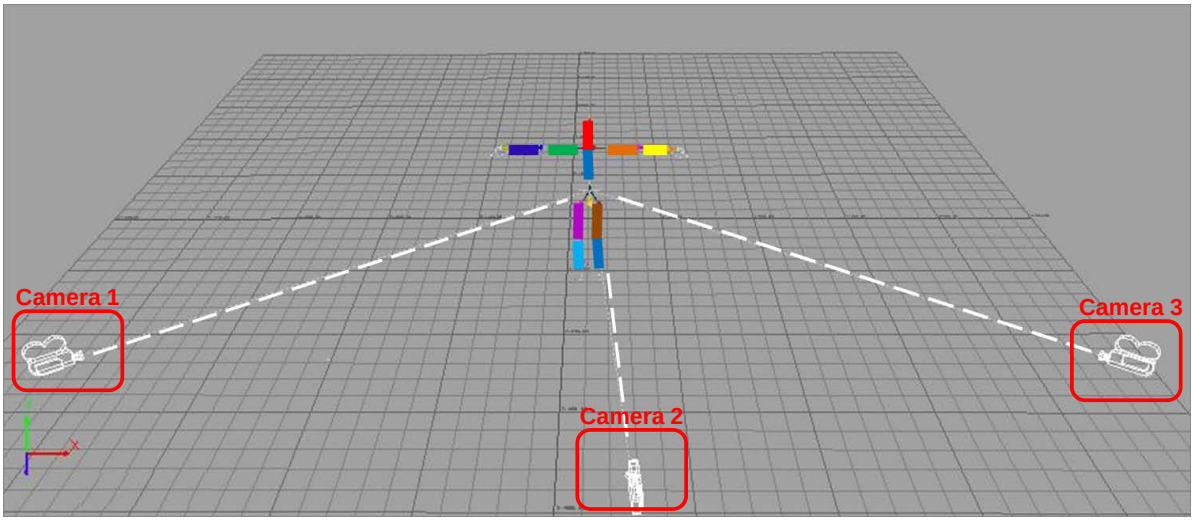


Figure 6.2: **Illustration of the camera setup in the scene**. A set of three camera view points (i.e., front, right, and left w.r.t. the character) is setup in the rendering scene.

dataset used to train Actlets in Section 6.2. Section 6.3 then describes application of Actlets in human action recognition. Finally, Section 6.4 concludes this chapter with a discussion.

6.1 Synthetic dataset of human motions

To train a representative set of Actlets, we need a relatively large amount of training data. The training data should cover a diverse range of human movements and should contain annotated positions of body-joints over time. Also, a significant amount of variation in terms of appearance (e.g., clothing and background), view-point, illumination, camera motion, and action styles, is required to span the expected variability in the test videos. While manual annotation of body-joints and their motion in video is highly time-consuming and therefore impractical, we resort to animation techniques and use motion capture data to build a synthetic dataset. The main advantage in this approach is the availability of the ground-truth positions of body-joints in each synthesized video, provided by the 2D projections of 3D body-joint positions in the motion-capture data. Furthermore, the approach allows to generate large amount of videos while inducing a diverse range of variations in view-points, camera motion, scale, illumination, clothing, physique, background, etc. The downside, on the other hand, is that the synthetic appearance may not match well with the real video. Nonetheless, we experimentally

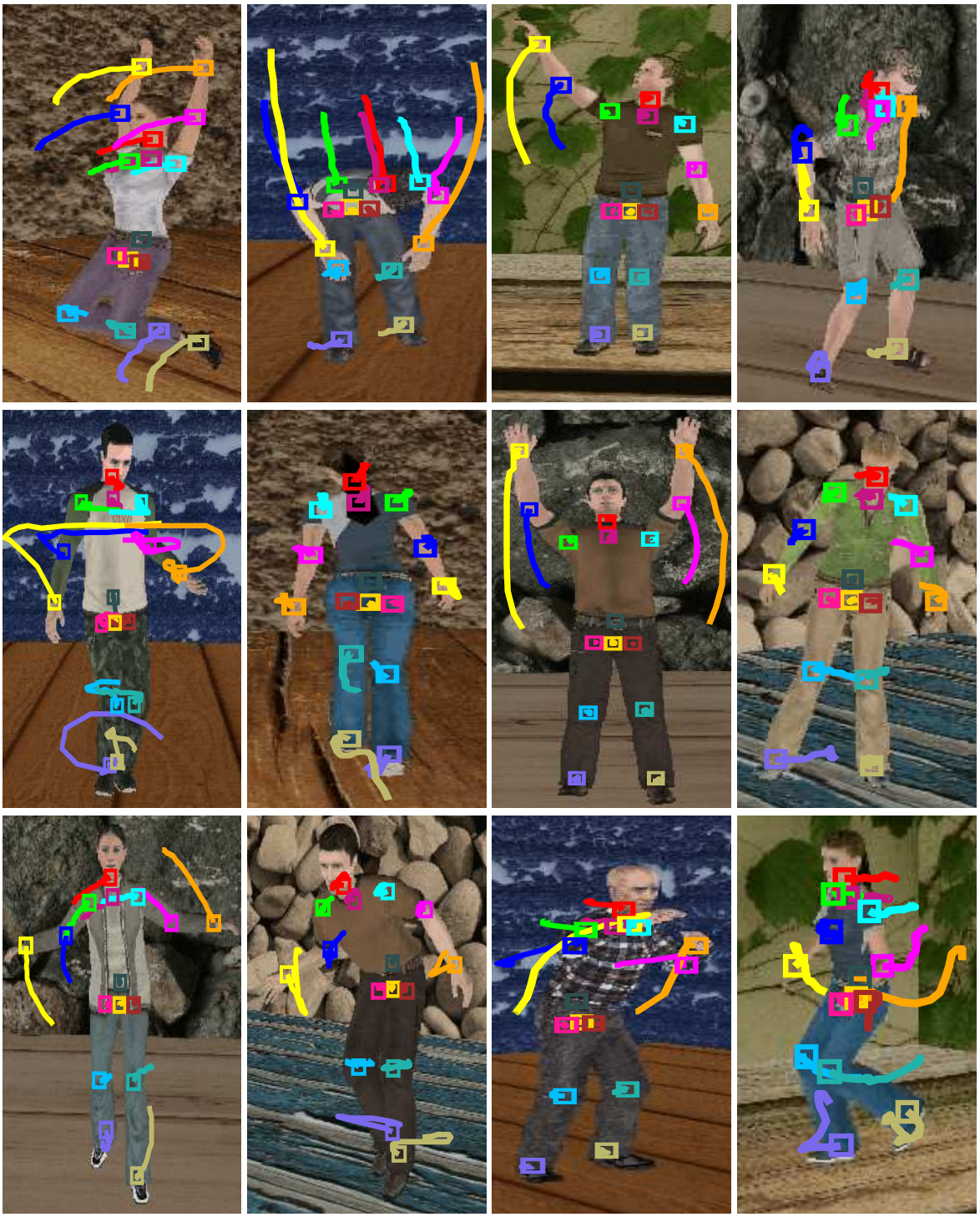


Figure 6.3: **Illustration of the synthetic dataset**. Sample frames from our synthetic dataset illustrating variability of generated videos in terms of view points, backgrounds, character physique, clothing, and motion. Color curves illustrate automatically annotated trajectories of body-joints.

demonstrate in Section 6.3, that we can leverage the synthetic data to learn informative Actlets, performing well on the real video data.

We use the CMU motion capture database[1], containing a large number of human motion sequences; from simple locomotions and physical activities to more complex movements involving human interactions. We perform motion re-targeting of the CMU motion capture sequences on 3D humanoid characters in Autodesk MotionBuilder 2011, and render videos from a set of fixed locations. We use ten 3D characters including males and females of different physiques, wearing different clothes. We render videos from a set of three different camera view points (front, right, and left, with respect to the character, see Figure 6.2) while using five different static backgrounds. Additionally, we simulate the panning of the camera which follows the motion of the character in each video. We render one video for each motion capture sequence in the CMU database while randomly choosing a character, background, and a view point. As a result, we get 2549 synthetic video sequences in total. All the synthetic videos are rendered at a resolution of 640×480 pixels and a frame rate of 24 FPS. Figure 6.3 illustrates a few example frames from our synthetic dataset together with the automatically annotated trajectories of body joints.

6.2 Training Actlets

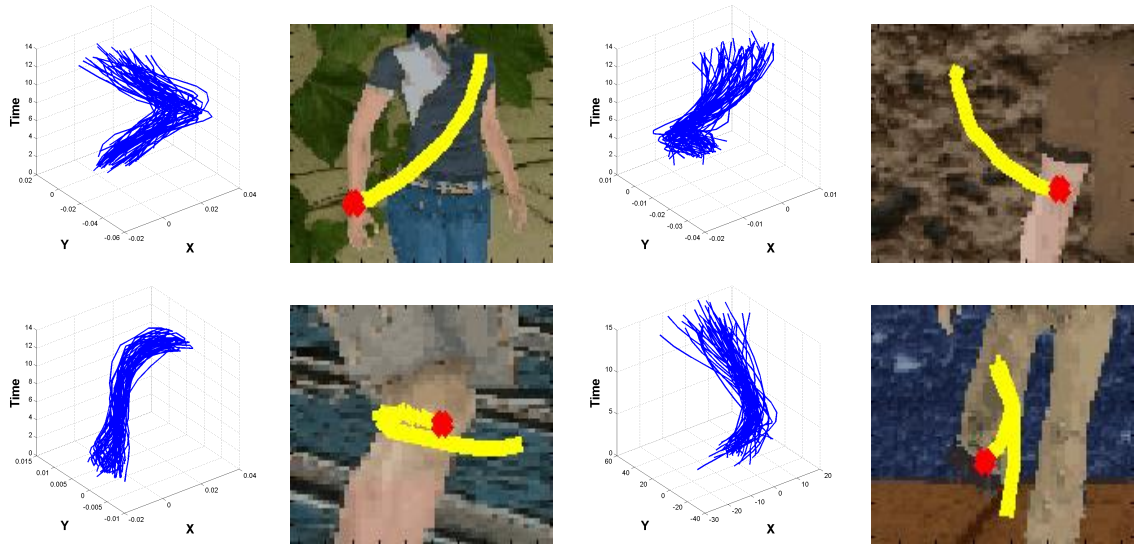
Here, we consider the motion of nine body-joints (head, left/right elbow, left/right wrist, left/right knee and left/right ankle), as these are expected to provide rich action description. These nine body-joints are treated in two ways: (a) grouping of similar motion patterns associated with each body-joint alone, and (b) grouping of similar motion patterns associated with two body-joints together. We perform clustering of 2D trajectories associated with the body-joints. We then extract video patches for each trajectory (or a pair of trajectories) and use them to train one Actlet classifier for each trajectory cluster. The details of the method are described below.

6.2.1 Trajectory representation

For each of the nine body-joints in a synthetic video, the associated 2D trajectory with spatial coordinates (x_t, y_t) over time $t \in 1...T$ is subdivided into overlapping sub-trajectories, each having a length of $L = 15$ frames. The shape of a sub-trajectory encodes the local motion pattern associated with the body-joint. Following [130], we represent

[1]Available at: `http://mocap.cs.cmu.edu`

(a) 1-joint-based trajectory clusters



(b) 2-joints-based trajectory clusters

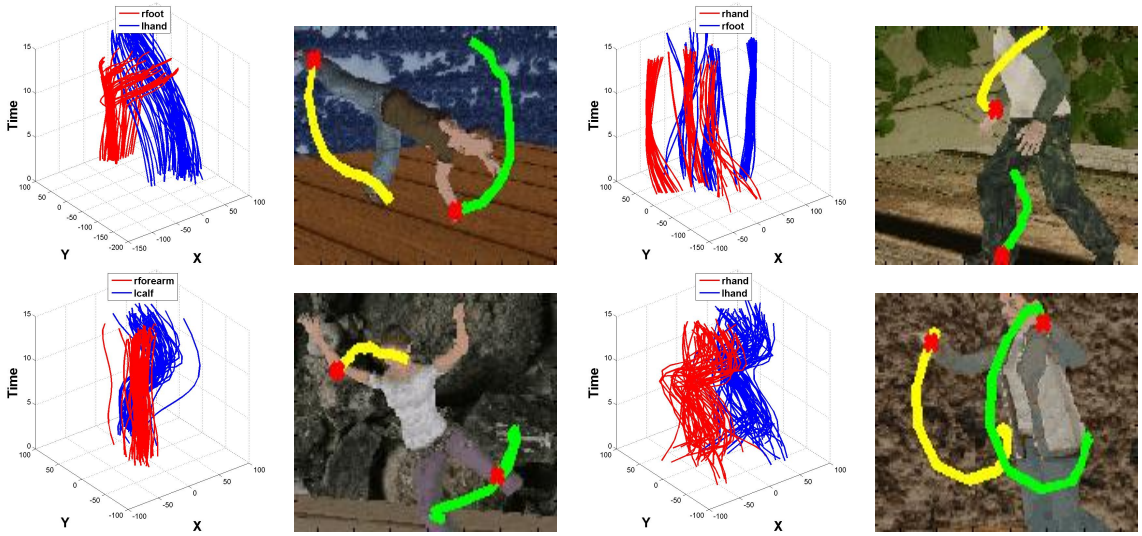


Figure 6.4: **Illustration of body-joint trajectory clusters**. Two types of Actlet clusters are: (a) based on motion patterns of only one body-joint, and (b) based on motion patterns of two body-joints together. All trajectories within a cluster are shown in separate plots by blue and red curves. An example video patch for each cluster is also shown.

the shape of a sub-trajectory with a velocity-based vector. Given a sub-trajectory of length L, we describe its shape by a sequence $S = (\Delta P_t, \ldots, \Delta P_{t+L-1})$ of displacement vectors $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. The resulting vector S is normalized by the height of the character in the rendered video. This normalization is required to discard the magnitude information, and is similar to that used in other techniques [130, 113, 114].

6.2.2 Clustering and training of Actlets

To group similar motion patterns associated with each body-joint (or a pair of body-joints), we perform k-means clustering (we set $k = 75$) on all the sub-trajectories associated with each of the nine body-joints (or a pair of body-joints) in all the 2549 synthetic videos. For pairs of body-joints, we consider all the 36 pairs of body-joints among the initial set of nine body-joints. We avoid occluded body joints by removing trajectories of right/left joints from the videos synthesized for the left/right views of the person respectively. For instance, consider the case of a video illustrated in Figure 6.3 (last row and last column), wherein, the right side of the character is occluded. In this case, we only consider the trajectories associated with the left side body-joints of the character, which are fully visible. Moreover, we perform both view-specific and view-independent clustering, where trajectories from the three different views are clustered either separately or jointly. To select distinct clusters, we sort clusters for each body-joint (or a pair of body-joints) according to the decreasing sum of distances to other clusters and keep the top $n = 50$ clusters from them. Figure 6.4 illustrates examples of the corresponding 1-joint-based and 2-joints-based clusters.

To train an Actlet for a given body-joint (or a pair of body-joints) and motion pattern(s), we extract video patches in the neighborhood of trajectories from one trajectory cluster. These video patches serve as positive training samples for an Actlet. For the negative training samples, we randomly extract 10,000 synthetic video patches, corresponding to trajectories from the remaining $n - 1$ clusters of the same body-joint (or pair of body-joints). We represent the extracted video patches by the histograms of oriented gradients (HOG), histograms of optical flow (HOF), and their combination, i.e., the HOGHOF descriptors [91]. We then train a linear Hellinger's SVM classifier on the respective descriptors. This way, for each descriptor type, we obtain a total of 1000 linear SVM classifiers for 1-joint-based Actlets[2], whereas, 1164 linear SVM classifiers for

[2]Front: 9 joints×50 clusters + left/right: 2×5 joints×50 clusters + view-independent: 9 joints×50

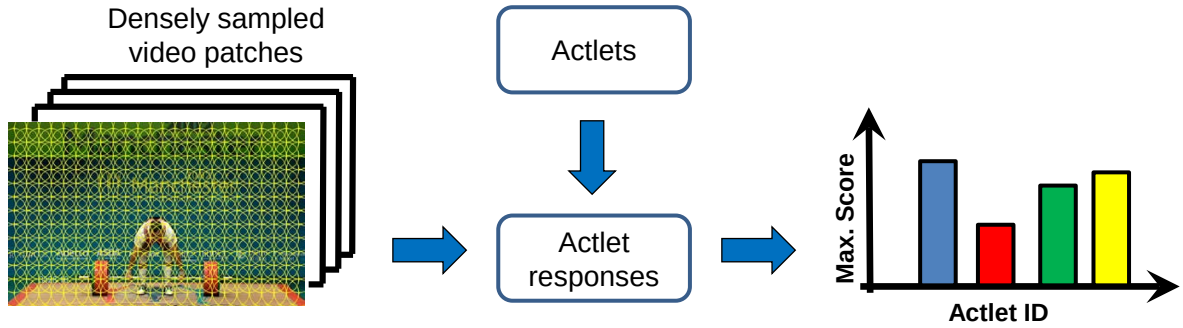


Figure 6.5: **Illustration of Actlets-based video representation**. Actlets are applied on a densely-sampled video sequence, and the maximum response corresponding to each Actlet classifier is subsequently concatenated into a vector representation (refer to the text for further details).

2-joints-based Actlets[3], corresponding to the view-specific and view-independent cases.

6.3 Actlets for action recognition

Actlets provide a means to detect specific motion patterns of body-joints in video disregarding irrelevant variations of the data in terms of backgrounds, clothing, view points and other factors. Our next goal is to deploy such descriptors for action recognition in real video. Given a video sequence v, we extract densely-sampled video patches and represent them by the HOG, HOF, and HOGHOF descriptors. For each descriptor, and each type of Actlets (i.e., 1-joint-based and 2-joints-based), we obtain a set of Actlet scores according to all the trained Actlet classifiers corresponding to the same type of descriptor. This way, we obtain an Actlet filter response volume Ω_{a_k} for the Actlet classifier a_k. Let n be the total number of Actlet classifiers. We use maximum pooling on the resulting n response volumes, and concatenate the maximum score of each Actlet classifier a_i into a vector representation:

$$\left[\max_{(x,y,t)} \Omega_{a_1}, \ldots, \max_{(x,y,t)} \Omega_{a_n}\right], \tag{6.1}$$

where (x, y, t) denotes the spatio-temporal volume for the max-pooling, which in this case is the whole video, as illustrated in Figure 6.5. Following the Attribute Bank representation, we use the 24-level spatio-temporal grids [91], and divide each response

clusters. We train Actlets for clusters with the minimum of 50 trajectories.

[3]Front: 36 2-joints×50 clusters + left/right: 2×10 2-joints×50 clusters + view-independent: 36 2-joints×50 clusters. We train Actlets for clusters with the minimum of 50 trajectories.

Channels	STGrid-24 (Baseline)	Actlets1 HOG	Actlets1 HOF	Actlets1 HOGHOF	Actlets2 HOG	Actlets2 HOF	Actlets2 HOGHOF
mean AP	**0.525**	0.353	0.455	0.456	0.333	0.455	0.468
AnswerPhone	0.259	0.189	0.251	0.269	0.250	0.249	**0.295**
DriveCar	**0.859**	0.653	0.803	0.827	0.556	0.793	0.835
Eat	**0.607**	0.182	0.572	0.489	0.077	0.547	0.480
FightPerson	**0.749**	0.375	0.532	0.469	0.409	0.538	0.526
GetOutCar	**0.447**	0.368	0.227	0.365	0.198	0.299	0.339
HandShake	0.285	0.294	0.317	0.297	0.253	**0.363**	0.304
HugPerson	**0.461**	0.274	0.287	0.304	0.256	0.265	0.333
Kiss	0.569	0.444	0.505	**0.571**	0.457	0.503	0.562
Run	**0.698**	0.534	0.614	0.600	0.560	0.575	0.638
SitDown	0.589	0.303	**0.611**	0.556	0.368	0.603	0.555
SitUp	**0.202**	0.196	0.149	0.141	0.168	0.128	0.130
StandUp	0.574	0.425	0.598	0.583	0.444	0.597	**0.621**

Table 6.1: Per-class average precision (AP) performance by different channels on the Hollywood-2 dataset.

Channels	STGrid-24 (Baseline)	Actlets1 HOG + STGrid-24	Actlets1 HOF + STGrid-24	Actlets1 HOGHOF + STGrid-24	Actlets2 HOG + STGrid-24	Actlets2 HOF + STGrid-24	Actlets2 HOGHOF + STGrid-24
mean AP	0.525	0.499	0.529	0.527	0.505	**0.531**	0.529
AnswerPhone	0.259	0.292	0.289	0.329	0.312	0.276	**0.332**
DriveCar	0.859	0.864	**0.884**	0.875	0.849	0.878	0.870
Eat	0.607	0.527	**0.649**	0.632	0.605	0.645	0.636
FightPerson	**0.749**	0.653	0.696	0.658	0.659	0.697	0.672
GetOutCar	0.447	0.408	0.422	0.433	0.374	**0.480**	0.446
HandShake	0.285	0.297	0.291	**0.316**	0.283	0.287	0.312
HugPerson	**0.461**	0.441	0.400	0.422	0.424	0.411	0.416
Kiss	0.569	0.573	0.594	**0.612**	0.590	0.593	0.609
Run	**0.698**	0.657	0.681	0.673	0.649	0.677	0.679
SitDown	0.589	0.552	**0.654**	0.605	0.535	0.645	0.595
SitUp	**0.202**	0.167	0.182	0.161	0.199	0.179	0.169
StandUp	0.574	0.561	**0.608**	0.603	0.579	0.602	0.614

Table 6.2: Per-class average precision (AP) performance by different channels/channel-combinations on the Hollywood-2 dataset.

volume Ω_{a_i} into 24 different types of grids. For each grid with m cells, the corresponding video representation is the concatenation of Actlet features in each grid cell c:

$$\left[\max_{(x,y,t)_c} \Omega_{a_1}, \ldots, \max_{(x,y,t)_c} \Omega_{a_n}\right]_{c=1}^{m}. \tag{6.2}$$

Consequently, the corresponding video representations (*Actlets1HOG*, *Actlets1HOF*, *Actlets1HOGHOF*, *Actlets2HOG*, *Actlets2HOF*, and *Actlets2HOGHOF*) are each comprised of 24 spatio-temporal grid channels.

For action classification based on the Actlet channels, we use a non-linear SVM with RBF kernel. We use Bag-of-Features (BoF) video representation as a baseline. Here, we follow

Chapter 3 and build the BoF video representation using the Harris3D feature points [88] in combination with the HOGHOF descriptors. We refer to this video representation as the *BoF* channel, and employ a non-linear SVM with χ^2 kernel for classification. Moreover, we integrate the Actlet channels with the BoF and STGrid-24 channels using the multi-channel kernel [193].

6.3.1 Experiments

Here, we evaluate performance of the Actlet channels for the task of action recognition on three challenging datasets: Hollywood-2, UCF-Sports and YouTube-Actions. The last two datasets mainly contain sports action classes (see Chapter 2 for detailed description). We report the results separately for each dataset.

Hollywood-2 results

Results on the Hollywood-2 dataset are presented in Table 6.1 and Table 6.2. Table 6.1 reports results for the baseline STGrid-24 channels as well as for all the individual Actlet channels. We can observe that the performance provided by all the Actlet channels is lower than the baseline STGrid-24 channels. Among the Actlets, the HOG-based Actlets perform the worst. Nevertheless, the HOGHOF-based Actlets seem to perform better than the HOF-based Actlets. This observation suggests that both appearance (i.e., HOG) and motion (i.e., HOF) information can be helpful in learning good Actlets.

Table 6.2 presents results for the combination of Actlet channels with the baseline STGrid-24 channels. It turns out that the HOG-based Actlets degrade the baseline performance by STGrid-24 channels. Whereas, HOF-based and HOGHOF-based Actlets result in no or only marginal improvement over the baseline. Overall, Actlets have not helped to improve the baseline performance on this dataset. Actually, the Hollywood-2 actions involve relatively few human kinematics compared to sports actions in the UCF-Sports and YouTube-Actions datasets. Since Actlets are trained to capture the dynamics of different moving body parts, their advantage can be better observed when recognizing kinematic actions, as we demonstrate on the UCF-Sports and YouTube-Actions datasets.

UCF-Sports results

Results on the UCF-Sports dataset are presented in Table 6.3 and Table 6.4. Table 6.3 now presents results for the baseline BoF channel as well as for all the Actlet channels. Here, the performance by all the Actlet channels is close to that by the baseline BoF channel. Among the Actlets, the performance by the HOG-based and HOF-based Actlets

Channels %	BoF (Baseline)	Actlets1 HOG	Actlets1 HOF	Actlets1 HOGHOF	Actlets2 HOG	Actlets2 HOF	Actlets2 HOGHOF
Average accuracy	077.25	075.02	074.46	**077.77**	075.63	076.07	076.82
Dive	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
GolfSwing	066.67	077.78	050.00	**088.89**	077.78	066.67	083.33
KickBall	085.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
WeightLift	**100.00**	083.33	083.33	083.33	083.33	083.33	083.33
HorseRide	**066.67**	058.33	050.00	050.00	058.33	041.67	041.67
Run	076.92	**084.62**	053.85	069.23	076.92	061.54	061.54
SwingPommel	085.00	085.00	**100.00**	**100.00**	095.00	**100.00**	**100.00**
Skateboard	016.67	008.33	**033.33**	016.67	016.67	**033.33**	**033.33**
Walk	**090.91**	068.18	081.82	077.27	063.64	081.82	072.73
SwingHighBar	084.62	084.62	**092.31**	**092.31**	084.62	**092.31**	**092.31**

Table 6.3: Performance accuracy by different channels on the UCF-Sports dataset.

Channels %	BoF (Baseline)	Actlets1 HOG + BoF	Actlets1 HOF + BoF	Actlets1 HOGHOF + BoF	Actlets2 HOG + BoF	Actlets2 HOF + BoF	Actlets2 HOGHOF + BoF	Kläser *et al.*[78]
Average accuracy	077.25	079.88	079.22	081.29	082.21	081.90	**083.24**	083.13
Dive	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
GolfSwing	066.67	083.33	072.22	083.33	**088.89**	**088.89**	**088.89**	079.60
KickBall	085.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	083.90
WeightLift	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	071.64
HorseRide	**066.67**	**066.67**	058.33	**066.67**	**066.67**	**066.67**	**066.67**	059.20
Run	**076.92**	**076.92**	061.54	**076.92**	**076.92**	**076.92**	**076.92**	076.00
SwingPommel	085.00	085.00	095.00	090.00	095.00	095.00	**100.00**	095.00
Skateboard	016.67	025.00	025.00	025.00	025.00	008.33	016.67	**083.30**
Walk	090.91	077.27	**095.46**	086.36	077.27	090.91	090.91	082.64
SwingHighBar	084.62	084.62	084.62	084.62	092.31	092.31	092.31	**100.00**

Table 6.4: Performance accuracy by different channels/channel-combinations on the UCF-Sports dataset.

is comparable; whereas, the HOGHOF-based Actlets again perform better. Moreover, performance by the Actlets1HOGHOF channels is slightly better than the baseline.

Table 6.4 reports results for the Actlet channels in combination with the baseline BoF channel. We can see that each Actlet channel has improved the baseline performance. We can also observe that performance by the individual Actlet channels (in Table 6.3) is reflected in their combination with the baseline BoF channel; HOG-based and HOF-based Actlets perform comparable, whereas, HOGHOF-based Actlets perform better. The best performance is by the Actlets2HOGHOF channels, i.e., 83.24%, which is an improvement of about 6% over the BoF baseline (i.e., 77.25%).

Moreover, we compare our results with those of Kläser *et al.*[78] in Table 6.4. Their method is based on a Bag-of-Features approach. The authors use regular dense sampling of feature points (similar to [176]), and compute the HOG3D descriptors [77]. For classification, they employ a non-linear SVM with χ^2 kernel. We can observe that the

Channels %	BoF (Baseline)	Actlets1 HOG	Actlets1 HOF	Actlets1 HOGHOF	Actlets2 HOG	Actlets2 HOF	Actlets2 HOGHOF
Average accuracy	62.95	56.06	64.57	65.66	58.87	63.27	**67.09**
Bike	71.51	81.08	71.24	**84.46**	81.29	69.24	80.85
Dive	85.00	59.00	**90.00**	80.00	74.00	84.00	81.00
Golf	73.00	88.00	76.00	86.00	**89.00**	77.00	**89.00**
SoccerJuggle	50.00	10.00	**51.00**	36.00	20.00	**51.00**	41.00
TrampolineJump	**74.00**	58.00	64.00	61.00	62.00	64.00	68.00
HorseRide	72.00	71.00	70.00	75.00	**76.00**	69.00	75.00
BasketballShoot	33.67	41.67	41.00	**46.00**	31.67	45.00	**46.00**
VolleyballSpike	73.00	72.00	80.00	82.00	72.00	79.00	**83.00**
Swing	71.00	62.00	**80.00**	76.00	60.00	76.00	**80.00**
TennisSwing	46.00	35.00	46.00	**56.00**	52.00	42.00	**56.00**
Walk	**43.30**	38.94	40.99	39.82	29.61	39.70	38.19

Table 6.5: Performance accuracy by different channels on the YouTube-Actions dataset.

Channels %	BoF (Baseline)	Actlets1 HOG + BoF	Actlets1 HOF + BoF	Actlets1 HOGHOF + BoF	Actlets2 HOG + BoF	Actlets2 HOF + BoF	Actlets2 HOGHOF + BoF	Liu *et al.*[101]
Average accuracy	62.95	67.03	69.89	70.99	66.52	68.56	70.07	**71.21**
Bike	71.51	82.76	81.42	**85.43**	77.40	78.75	82.07	73.00
Dive	85.00	82.00	**90.00**	89.00	86.00	**90.00**	88.00	81.00
Golf	73.00	87.00	87.00	86.00	**91.00**	87.00	89.00	86.00
SoccerJuggle	50.00	49.00	**59.00**	55.00	48.00	57.00	57.00	54.00
TrampolineJump	74.00	75.00	74.00	75.00	73.00	72.00	75.00	**79.00**
HorseRide	72.00	71.00	73.00	**75.00**	69.00	70.00	73.00	72.00
BasketballShoot	33.67	39.67	39.67	40.67	34.33	40.67	41.67	**53.00**
VolleyballSpike	73.00	84.00	85.00	**87.00**	79.00	82.00	85.00	73.30
Swing	71.00	72.00	**77.00**	**77.00**	76.00	**77.00**	**77.00**	57.00
TennisSwing	46.00	48.00	54.00	60.00	55.00	53.00	56.00	**80.00**
Walk	43.30	46.90	48.70	50.83	43.03	46.70	47.03	**75.00**

Table 6.6: Performance accuracy by different channels/channel-combinations on the YouTube-Actions dataset.

Actlet channels have helped to improve 7 out of 10 action classes (the best accuracies are marked in bold).

YouTube-Actions results

Results for the YouTube-Actions dataset are presented in Table 6.5 and Table 6.6. Table 6.5 presents results for the baseline BoF channel as well as for the individual Actlet channels. Here, the HOF-based and HOGHOF-based Actlets perform better than the baseline BoF channel. Among the Actlets, performance improves in the following order: HOG-based<HOF-based<HOGHOF-based. The best performance is by the Actlets2HOGHOF channels.

Table 6.6 then presents results for the combination of Actlet channels with the baseline BoF channel. We can note that each Actlet combination improves the baseline performance, again in the order: HOG-based<HOF-based<HOGHOF-based. The best

	Actlets1 HOG	Actlets1 HOF	Actlets1 HOGHOF	Actlets2 HOG	Actlets2 HOF	Actlets2 HOGHOF
FPS	0.7	0.7	0.6	0.7	0.7	0.6

Table 6.7: Computational cost in frames per second (FPS) for each type of Actlet channels.

performance is by the Actlets1HOGHOF channels, i.e., 70.99%, which is an improvement of about 8% over the BoF baseline (i.e., 62.95%).

We also compare our results with those of Liu *et al.*[101] (who actually published the dataset) in Table 6.6. The authors employ both motion and static features in a Bag-of-Features framework. They propose to use a divisive information-theoretic algorithm to learn compact yet discriminative visual vocabularies. Finally, they employ AdaBoost to integrate all the heterogeneous yet complementary features for recognition. In comparison, we can observe that the Actlet channels have helped to improve 7 out of 11 action classes (the best accuracies are marked in bold).

6.3.2 Discussion

Our empirical evaluation suggests that Actlets are suitable for sports-like actions, which involve substantial amount of human kinematics. This is owing to the fact that Actlets have been trained from the CMU motion capture sequences, which largely contain locomotions, physical and sports activities. Consequently, Actlets are trained to capture action-characteristic local motion patterns associated with different body-joints.

Results on the UCF-Sports and YouTube-Actions datasets have shown the effectiveness of Actlets. In particular, Actlets when combined with the baseline BoF, result in a significant improvement over the baseline. This performance gain indicates their complementarity. Actlets focus on characteristic local movements of people, whereas, BoF has a potential of capturing additional contextual information from the background.

6.3.3 Computational cost

Here, we evaluate the computational cost of Actlets. This cost measures the run-time in computing the Actlet scores on an input video sequence, which includes the run-time of dense feature extraction. However, we do not take into account computation of the 24 spatio-temporal grid channels for each type of Actlets. The evaluation is performed on a

set of videos from the Hollywood-2 dataset with spatial resolution up to 720×480 pixels (full resolution) and about 5550 frames of length in total. The run-time estimates for the C++ implementation of Actlets are obtained on an octa-core 64-bit Linux cluster node with 2.33 GHz processors and 16 GB RAM. However, the computational cost is estimated on a single core without any parallel processing. Table 6.7 presents results for each type of Actlets in terms of the average number of frames per second (FPS). On average, the computational cost of Actlets is 0.7 FPS. Note that the training run-time of Actlets is not included in this cost.

6.4 Discussion

Our synthetic dataset which is used to train Actlets is currently comprised of 2549 videos, i.e., only one video per CMU motion-capture sequence. As discussed before, our automatic approach gives us the freedom to introduce a variety of visual variations while rendering a synthetic video. That means, we can easily diversify the synthetic dataset in terms of person appearance, view-points, background, lighting, camera motion, etc., and generate a large amount of annotated video data. As a result, we can expect to obtain further improvements from retrained Actlets.

We train Actlets using the HOG, HOF, and HOGHOF descriptors [91]. Generally, HOGHOF-based Actlets give good results, which suggests that both appearance (i.e., HOG) and motion (i.e., HOF) information is useful in learning robust Actlets. It would be interesting to evaluate other types of descriptors with Actlets, such as the HOG3D descriptor [77].

Chapter 7

Summary and future perspectives

This thesis has targeted the problem of human action recognition in realistic kind of video data, such as movies and online videos. To this end, our approach has been to develop new supervised statistical representations, aiming to improve limitations of current local features based methods. We have first evaluated a range of available methods for local feature detection and description on the task of action classification. We have employed a common bag-of-features framework to evaluate and compare three interest point detectors and six descriptors, we have also introduced and evaluated densely sampled features. We have performed the experiments on three different datasets, of varying realistic variations, with 25 action classes in total. Among the main conclusions, we have observed that dense sampling of feature points outperforms interest point detectors on the realistic UCF-Sports and Hollywood-2 actions datasets. We have observed a rather similar performance by interest point detectors on each dataset. Across the datasets, the Harris3D detector has performed better on the KTH-Actions dataset, whereas, the Gabor detector has given better results for the UCF-Sports and Hollywood-2 datasets. Among the descriptors, HOG/HOF and HOG3D have shown good results.

Next, we have proposed to improve the standard bag-of-features representation by integrating non-local region-level information. The motivation behind this approach is that the inherently limited discriminative power of local features can be enhanced by disambiguating them through region-level cues. In particular, we have investigated both unsupervised and supervised video segmentation using (i) motion-based foreground separation, (ii) person detection, (iii) static action detection, and (iv) object detection. We have empirically shown that such region-level information provides complementary information to the local Harris3D features. Moreover, we have exploited the complementary nature of the resulting alternative video representations in a kernel combination

framework, and demonstrated promising results on the challenging Hollywood-2 dataset.

Moreover, we have investigated an attribute-based approach to integrate high-level information with the bag-of-features representation. The proposed Attribute Bank representation is capable of detecting characteristic attributes (e.g., objects, static actions, and poses) in video, and provides complementary high-level information to the low-level Harris3D features. The Attribute Bank representation is based on pre-trained detectors, which have been trained on large number of static images. Empirical evaluation on the Hollywood-2 dataset has demonstrated significant improvement (i.e., 4.6%) over a strong baseline.

Finally, we have proposed a novel approach to represent discriminative local motion patterns in video, which we refer to as *Actlets*. Actlets are body-part detectors, undergoing action-characteristic local motions. To train such specialized detectors, we have proposed to avoid the labor-extensive annotation in videos, and synthesized a large amount of videos by utilizing the motion-capture data. We have proposed a supervised approach to train Actlets, while learning invariance to distracting variations in video, such as person appearance and action styles, view-point changes, lighting conditions, and camera motions. We have then successfully employed Actlets in the Attribute Bank Framework for the task of human action classification in video. We have experimentally shown that Actlets capture discriminative local motion patterns in video, and provide complementary information to the bag-of-features representation. Quantitative results on the challenging UCF-Sports and YouTube-Actions datasets have shown the promise of the proposed approach.

All the methods proposed and developed in this thesis illustrate alternative ways of constructing supervised video representations. Empirical evaluation on several datasets demonstrates improvements of human action recognition in realistic settings. Moreover, the proposed supervised video representations are shown to be efficient, as many of them can be computed using readily available object, action, and pose detectors.

7.1 Future directions

The work presented in this thesis can be extended in several directions. Here, we have referred to a video representation as a *channel* [94]. We have investigated several supervised approaches to build new channels for human action recognition in realistic video. Moreover, we have employed a simple approach [193] (i.e., product of kernels) to

combine multiple channels. Our experimental evaluation on realistic datasets (such as the Hollywood-2 dataset) have shown that particular action classes benefit from specific channel(s). This observation highlights the need for (i) specialized features for particular action classes, and (ii) more sophisticated technique for channel combination/selection. Approaches based on early fusion (e.g., [76]) as well as late-fusion (e.g., Multiple Kernel Learning [141]) can be investigated to improve the channel combination.

The Attribute Bank representation is currently comprised of a modest set of attributes (4 objects + 8 actions + 150 poselets). Yet, we have empirically shown that such representation is highly discriminative, which captures high-level information in video, and provides complementary information to low-level features. As envisioned by Li *et al.* [98, 99], we argue that the performance gain can be further increased by including more attributes, based on more objects, scene context, and color, for instance. Moreover, we have introduced weak geometrical information within the Attribute Bank representation, in the form of coarse spatio-temporal grids [91]. More robust geometrical information, such as scene layout and depth information, could further improve the performance.

Our synthetic dataset of human motions, which has been used to learn Actlets, is currently limited to 2549 videos in total. As discussed in Chapter 6, our automated approach gives the full control to induce realistic variations in synthetic videos, in terms of person appearance and physique, background, illumination, view-point, and camera motion. We can, therefore, further extend the dataset to include more diversity in human motions. For instance, we can include more 3D characters of different appearance and physique, different view points, and lighting variations. A relatively large and diversified synthetic dataset is expected to help in learning even stronger Actlets.

As demonstrated in Chapter 6, Actlets have the potential to detect characteristic local motions in video, such as right foot forward, left hand swing, etc. Therefore, Actlets can be employed to perform automatic grouping of video clips (e.g., YouTube videos), based on the statistics of the detected action primitives.

Bibliography

[1] C. Achard, X.T. Qu, A. Mokhber, and M. Milgram. A novel approach for recognition of human actions with semi-global features. *Machine Vision and Applications*, 19(1):27–34, January 2008.

[2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, January 2006.

[3] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[4] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):Article No 16, April 2011.

[5] M.A.R. Ahad, T. Ogata, J.K. Tan, H.S. Kim, and S. Ishikawa. Motion recognition approach to solve overwriting in complex actions. In *8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–6, 2008.

[6] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):288–303, February 2010.

[7] C.D. Barclay, J.E. Cutting, and L.T Kozlowski. Temporal and spatial facotors in gait perception that influence gender recognition. *Perception and Psychophysics*, 23(2):145–152, 1978.

[8] H. Bay, T. Tuytelaars, and L.J. Van Gool. Surf: Speeded up robust features. In *Proc. European Conference on Computer Vision*, pages I: 404–417, 2006.

[9] P.R. Beaudet. Rotationally invariant image operators. In *Proc. International Conference on Pattern Recognition*, pages 579–583, 1978.

[10] S. Belongie, C. Fowlkes, F. Chung, and J. Malik. "Spectral Partitioning with Indefinite Kernels Using the Nyström Extension". In *Proc. European Conference on Computer Vision*, pages 531–542, London, UK, 2002. Springer-Verlag.

[11] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. International Conference on Computer Vision*, pages II: 1395–1402, 2005.

[12] A.F. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, 352:1257–1265, 1997.

[13] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001.

[14] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In *COLT'92: Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, New York, NY, USA, 1992. ACM Press.

[15] L. Bottou, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, L.D. Jackel, Y.L. Le Cun, U.A. Muller, E. Sackinger, P.Y. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwritten digit recognition. In *Proc. International Conference on Pattern Recognition*, pages B:77–82, 1994.

[16] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Proc. European Conference on Computer Vision*, pages VI: 168–181, 2010.

[17] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision (ICCV)*, 2009.

[18] M. Brand, N.M. Oliver, and A.P. Pentland. Coupled hidden markov models for complex action recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[19] M. Bregonzio, S.G. Gong, and T. Xiang. Recognizing action as clouds of space-time interest points. In *Proc. Computer Vision and Pattern Recognition*, pages 1948–1955, 2009.

[20] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *Proc. Computer Vision and Pattern Recognition*, pages 2225–2232, 2011.

[21] L.W. Campbell and A.F. Bobick. Recognition of human body motion using phase space constraints. In *Proc. International Conference on Computer Vision*, pages 624–630, 1995.

[22] C. Cedras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, March 1995.

[23] C. C. Chang and C. J. Lin. LIBSVM: a library for support vector machines, 2001.

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

[25] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *Proc. European Conference on Computer Vision*, pages IV: 158–171, 2008.

[26] N. Cristianini and J. Shawe Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[27] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop Statistical Learning in Computer Vision*, 2004.

[28] O.G. Cula and K.J. Dana. Compact representation of bidirectional texture functions. In *Proc. Computer Vision and Pattern Recognition*, pages I:1041–1047, 2001.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. Computer Vision and Pattern Recognition*, pages I: 886–893, 2005.

[30] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei Fei. Imagenet: A large-scale hierarchical image database. In *Proc. Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[31] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[32] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Proc. International Conference on Computer Vision*, pages 1491–1498, 2009.

[33] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. International Conference on Computer Vision*, pages 726–733, 2003.

[34] R.A. Elschlager and M.A. Fischler. The representation and matching of pictorial structures. In *IEEE Transactions on Computers*, pages 31–56, 1977.

[35] M. Enzweiler and D.M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, December 2009.

[36] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[38] R. Fablet and P. Bouthemy. Motion recognition using nonparametric image motion models estimated from temporal and multiscale co-occurrence statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1619–1624, December 2003.

[39] A. Farhadi, I. Endres, D. Hoiem, and D.A. Forsyth. Describing objects by their attributes. In *Proc. Computer Vision and Pattern Recognition*, pages 1778–1785, 2009.

[40] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[41] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005.

[42] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. Computer Vision and Pattern Recognition*, pages II: 264–271, 2003.

[43] V. Ferrari and A. Zisserman. Learning visual attributes. In *Advances in Neural Information Processing Systems*, December 2007.

[44] David A. Forsyth, Okan Arikan, and Deva Ramanan. D.: Computational studies of human motion: Part 1, tracking and motion synthesis. In *Foundations and Trends in Computer Graphics and Vision*, page 2006. Now Publishers Inc, 2006.

[45] B. Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *Proc. European Conference on Computer Vision*, pages I: 179–192, 2008.

[46] T. Gandhi and M.M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):413–430, September 2007.

[47] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.

[48] D.M. Gavrila and L.S. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *In International Workshop on Automatic Face- and Gesture-Recognition. IEEE Computer Society*, pages 272–277, 1995.

[49] P. Gehler and S. Nowozin. On feature combination methods for multiclass object classification. In *Proc. International Conference on Computer Vision*, 2009.

[50] D. Geronimo, A.M. Lopez, A.D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1239–1258, July 2010.

[51] A. Gilbert, J. Illingworth, and R. Bowden. Scale invariant action recognition using compound features mined from dense spatio-temporal corners. In *Proc. European Conference on Computer Vision*, pages I: 222–233, 2008.

[52] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Proc. International Conference on Computer Vision*, pages 925–931, 2009.

[53] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):883–897, January 2011.

[54] N.H. Goddard. The interpretation of visual motion: Recognizing moving light displays. In *Proceedings of Workshop on Visual Motion*, pages 212–220, 1989.

[55] N.H. Goddard. *The Perception of Articulated Motion: Recognizing Moving Light Displays*. PhD thesis, University of Rochester, Rochester, USA, 1992.

[56] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[57] R. D. Green and L. Guan. Quantifying and recognizing human movement patterns from monocular video images - part i: A new framework for modeling human motion. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:179–190, 2003.

[58] Y. Guo, G. Xu, and S. Tsuji. Understanding human motion patterns. In *Proc. International Conference on Pattern Recognition*, pages B:325–329, 1994.

[59] S. Gupta and R. Mooney. Using closed captions to train activity recognizers that improve video retrieval. In *Proceedings of the CVPR-09 Workshop on Visual and Contextual Learning from Annotated Images and Videos (VCL)*, Miami, FL, June 2009.

[60] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *Proc. International Conference on Computer Vision*, pages 1933–1940, 2009.

[61] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *Proc. International Conference on Computer Vision*, pages 991–998, 2011.

[62] C. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[63] S. Haykin. *"Neural Networks: A Comprehensive Foundation"*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[64] M. Hebert, A.A. Efros, and D. Hoiem. Putting objects in perspective. In *Proc. Computer Vision and Pattern Recognition*, pages II: 2137–2144, 2006.

[65] A. Hervieu, P. Bouthemy, and J.P. Le Cadre. A statistical video content recognition method using invariant features on object trajectories. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1533–1543, November 2008.

[66] D.C. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, February 1983.

[67] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34:334–352, 2004.

[68] N. Ikizler Cinbis, R.G. Cinbis, and S. Sclaroff. Learning actions from the web. In *Proc. International Conference on Computer Vision*, pages 995–1002, 2009.

[69] N. Ikizler Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *Proc. European Conference on Computer Vision*, pages I: 494–507, 2010.

[70] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *Proc. International Conference on Computer Vision*, pages 1–8, 2007.

[71] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 217–226, New York, NY, USA, 2006. ACM.

[72] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14(2):201–211, 1973.

[73] S. Johnson and M.R. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proc. Computer Vision and Pattern Recognition*, pages 1465–1472, 2011.

[74] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. International Conference on Computer Vision*, pages I: 604–610, 2005.

[75] T. Kadir and M. Brady. Scale saliency: a novel approach to salient feature and scale selection. In *International Conference on Visual Information Engineering*, pages 25–28, July 2003.

[76] F.S. Khan, J. van de Weijer, and M. Vanrell. Top-down color attention for object recognition. In *Proc. International Conference on Computer Vision*, pages 979–986, 2009.

[77] A. Klaeser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proc. British Machine Vision Conference*, 2008.

[78] A. Kläser, M. Marszałek, I. Laptev, and C. Schmid. Will person detection help bag-of-features action recognition? Technical Report RR-7373, INRIA Grenoble - Rhône-Alpes, Sept. 2010.

[79] A. Kokaram, N. Rea, R. Dahyot, M Tekalp, P. Bouthemy, P. Gros, and I. Sezan. Browsing sports video (trends in sports-related indexing and retrieval work). *IEEE Signal Processing Magazine*, 23(2):47–58, March 2006.

[80] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 2046–2053, 2010.

[81] V. Krüger, D. Kragic, A. Ude, and C. Geib. The meaning of action: A review on action recognitin and mapping. *Advanced Robotics*, 21(13):1473–1501, 2007.

[82] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[83] N. Kumar, A.C. Berg, P.N. Belhumeur, and S.K. Nayar. Attribute and simile classifiers for face verification. In *Proc. International Conference on Computer Vision*, pages 365–372, 2009.

[84] C.H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[85] C.H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. Computer Vision and Pattern Recognition*, pages 951–958, 2009.

[86] Gert Lanckriet, Nello Cristianini, Peter Bartlett, and Laurent El Ghaoui. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:2004, 2002.

[87] I. Laptev. *Local Spatio-Temporal Image Features for Motion Interpretation.* PhD thesis, Department of Numerical Analysis and Computer Science (NADA), KTH, 2004.

[88] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2/3):107–123, 2005.

[89] I. Laptev, B. Caputo, C. Schuldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108(3):207–229, December 2007.

[90] I. Laptev and T. Lindeberg. Space-time interest points. In *Proc. International Conference on Computer Vision*, 2003.

[91] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. Computer Vision and Pattern Recognition*, 2008.

[92] I. Laptev and P. Perez. Retrieving actions in movies. In *Proc. International Conference on Computer Vision*, pages 1–8, 2007.

[93] Ivan Laptev and Tony Lindeberg. Local descriptors for spatio-temporal recognition. In *In First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[94] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1265–1278, 2005.

[95] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. Computer Vision and Pattern Recognition*, pages II: 2169–2178, 2006.

[96] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proc. Computer Vision and Pattern Recognition*, 2005.

[97] F.F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. Computer Vision and Pattern Recognition*, pages II: 524–531, 2005.

[98] L-J. Li, H. Su, Y. Lim, and L. Fei-Fei. Objects as attributes for scene classification. In *ECCV Workshop Parts and Attributes*, 2010.

[99] Eric P. Xing Li-Jia Li, Hao Su and Li Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2010.

[100] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, November 1998.

[101] J. G. Liu, J. B. Luo, and M. Shah. Recognizing realistic actions from videos 'in the wild'. In *Proc. Computer Vision and Pattern Recognition*, 2009.

[102] J.G. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Proc. Computer Vision and Pattern Recognition*, pages 3337–3344, 2011.

[103] J.G. Liu and M. Shah. Learning human actions via information maximization. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[104] S.P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[105] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[106] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. International Conference on Computer Vision*, pages 1150–1157, 1999.

[107] W.L. Lu and J.J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. In *The 3rd Canadian Conference on Computer and Robot Vision*, pages 6–6, 2006.

[108] S. Maji, A.C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[109] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *Proc. Computer Vision and Pattern Recognition*, pages 3177–3184, 2011.

[110] D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London*, B-200:269–294, 1978.

[111] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *Proc. Computer Vision and Pattern Recognition*, 2009.

[112] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV Workshops*, pages 514–521, 2009.

[113] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *Proc. European Conference on Computer Vision*, pages I: 508–521, 2010.

[114] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *Proc. International Conference on Computer Vision*, pages 104–111, 2009.

[115] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. International Conference on Computer Vision*, pages I: 525–531, 2001.

[116] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[117] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, March 2001.

[118] T.B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 103(2-3):90–126, November 2006.

[119] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1632–1646, 2008.

[120] J.C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. European Conference on Computer Vision*, 2010.

[121] J.C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *Proc. British Machine Vision Conference*, 2006.

[122] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. Computer Vision and Pattern Recognition*, 2006.

[123] S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. In *Proc. Computer Vision and Pattern Recognition*, pages 469–474, 1994.

[124] J. M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Comm. and Image Representation*, 6(4):348–365, 1995.

[125] F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing*, 14(2):169–180, February 2005.

[126] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(3):710–719, June 2005.

[127] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. European Conference on Computer Vision*, pages Vol II: 71–84, 2004.

[128] A. Oreskovic. Youtube hits 4 billion daily video views. In *Reuters*, January 2012.

[129] O. Oshin, A. Gilbert, J. Illingworth, and R. Bowden. Spatio-temporal feature recognition using randomised ferns. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis*, 2008.

[130] R. Sukthankar P. Matikainen and M. Hebert. Feature seeding for action recognition. In *Proc. International Conference on Computer Vision*, 2011.

[131] A. Patron Perez, I.D. Reid, A. Patron, and I.D. Reid. A probabilistic framework for recognizing similar actions using spatio-temporal features. In *Proc. British Machine Vision Conference*, 2007.

[132] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. European Conference on Computer Vision*, pages IV: 143–156, 2010.

[133] P. Peursum, S. Venkatesh, and G.A.W. West. Tracking-as-recognition for articulated full-body human motion analysis. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[134] G. Piriou, P. Bouthemy, and J.F. Yao. Recognition of dynamic video contents with global probabilistic models of visual motion. *IEEE Transactions on Image Processing*, 15(11):3417–3430, November 2006.

[135] R. Polana and R.C. Nelson. Recognition of motion from temporal texture. In *Proc. Computer Vision and Pattern Recognition*, pages 129–134, 1992.

[136] R. Polana and R.C. Nelson. Low level recognition of human motion. In *Proceedings of the 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, 1994.

[137] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June 2010.

[138] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4–18, October 2007.

[139] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proc. International Conference on Computer Vision*, 2007.

[140] H. Ragheb, S.A. Velastin, P. Remagnino, and T. Ellis. Human action recognition using robust power spectrum features. In *Proc. International Conference on Image Processing*, pages 753–756, 2008.

[141] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[142] D. Ramanan, D.A. Forsyth, and A. Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proc. Computer Vision and Pattern Recognition*, pages I: 271–278, 2005.

[143] Deva Ramanan and David A. Forsyth. Automatic annotation of everyday movements. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[144] K. Rapantzikos, Y.S. Avrithis, and S.D. Kollias. Spatiotemporal saliency for event detection and representation in the 3d wavelet domain: potential in human action recognition. In *Proc. International Conference on Image and Video Retrieval*, pages 294–301, 2007.

[145] K. Rapantzikos, Y.S. Avrithis, and S.D. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 1454–1461, 2009.

[146] L. Råde and B. Westergren. *"Mathematics Handbook for Science and Engineering"*. Studentlitteratur, 5th edition, 2004.

[147] N.M. Robertson and I.D. Reid. Behaviour understanding in video: A combined method. In *Proc. International Conference on Computer Vision*, pages I: 808–815, 2005.

[148] M.D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[149] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics, and Image Processing*, 59(1):94–115, January 1994.

[150] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *Proc. Computer Vision and Pattern Recognition*, pages II: 721–727, 2000.

[151] B.C. Russell, A.B. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, May 2008.

[152] S. Sadanand and J.J. Corso. Action bank: A high-level representation of activity in video. In *Proc. Computer Vision and Pattern Recognition*, pages 1234–1241, 2012.

[153] G. Salton. *Automatic Information Organization and Retrieval*. McGraw Hill Text, 1968.

[154] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Proc. Computer Vision and Pattern Recognition*, pages 1281–1288, 2011.

[155] S. Savarese, A. DelPozo, J.C. Niebles, and L. Fei-Fei. Spatial-Temporal correlatons for unsupervised action classification. In *IEEE Workshop on Motion and video Computing, 2008. WMVC 2008*, pages 1–8, 2008.

[156] K. Schindler and L.J. Van Gool. Action snippets: How many frames does human action recognition require? In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[157] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proc. International Conference on Pattern Recognition*, 2004.

[158] 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*, 2007.

[159] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[160] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proc. International Conference on Computer Vision*, 2005.

[161] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. International Conference on Computer Vision*, 2003.

[162] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[163] C. Sminchisescu, A. Kanaujia, and D. Metaxas. Conditional models for contextual human motion recognition. *Computer Vision and Image Understanding*, 103(2-3):210–220, November 2006.

[164] T.E. Starner and A.P. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision*, pages 265–270, 1995.

[165] Y. Su and F. Jurie. Improving image classification using semantic attributes. *International Journal of Computer Vision*, 100(1):59–77, October 2012.

[166] S. Sumi. Upside-down presentation of the johansson moving light-spot pattern. *Perception*, 13(3):283–286, 1984.

[167] J. Sun, X. Wu, S.C. Yan, L.F. Cheong, T.S. Chua, and J.T. Li. Hierarchical spatio-temporal context modeling for action recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 2004–2011, 2009.

[168] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[169] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *Proc. European Conference on Computer Vision*, pages I: 548–561, 2008.

[170] P.K. Turaga, R. Chellappa, V.S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, November 2008.

[171] M. M. Ullah, S. N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *Proc. British Machine Vision Conference*, 2010.

[172] R. Urtasun, D.J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding*, 103(2-3):157–177, November 2006.

[173] J.C. van Gemert, J.M. Geusebroek, C.J. Veenman, and A.W.M. Smeulders. Kernel codebooks for scene categorization. In *Proc. European Conference on Computer Vision*, pages III: 696–709, 2008.

[174] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Computer Vision and Pattern Recognition*, 2001.

[175] H. Wang, A. Klaser, C. Schmid, and C.L. Liu. Action recognition by dense trajectories. In *Proc. Computer Vision and Pattern Recognition*, pages 3169–3176, 2011.

[176] H. Wang, M. M. Ullah, A. Klaša̋er, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. British Machine Vision Conference*, 2009.

[177] J.J. Wang, J.C. Yang, K. Yu, F.J. Lv, T.S. Huang, and Y.H. Gong. Locality-constrained linear coding for image classification. In *Proc. Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[178] L. Wang, W.M. Hu, and T.N. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, March 2003.

[179] Y. Wang, P. Sabzmeydani, and G. Mori. Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In *Proceedings of the 2nd conference on human motion: understanding, modeling, capture and animation*, pages 240–254, 2007.

[180] Y. Wang, D. Tran, and Z.C. Liao. Learning hierarchical poselets for human parsing. In *Proc. Computer Vision and Pattern Recognition*, pages 1705–1712, 2011.

[181] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proc. Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[182] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, February 2011.

[183] G. Willems, J.H. Becker, T. Tuytelaars, and L.J. Van Gool. Exemplar-based action recognition in video. In *Proc. British Machine Vision Conference*, 2009.

[184] G. Willems, T. Tuytelaars, and L.J. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. European Conference on Computer Vision*, pages II: 650–663, 2008.

[185] S.F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *Proc. International Conference on Computer Vision*, pages 1–8, 2007.

[186] X.X. Wu, D. Xu, L.X. Duan, and J.B. Luo. Action recognition using context and appearance distribution features. In *Proc. Computer Vision and Pattern Recognition*, pages 489–496, 2011.

[187] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Proc. Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[188] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. Computer Vision and Pattern Recognition*, pages 1385–1392, 2011.

[189] A. Yilmaz and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Proc. International Conference on Computer Vision*, pages I: 150–157, 2005.

[190] J.S. Yuan, Z.C. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *Proc. Computer Vision and Pattern Recognition*, pages 2442–2449, 2009.

[191] L. Zelnik Manor, M. Irani, D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 103(2-3):249–257, November 2006.

[192] Lihi Zelnik-Manor and Michal Irani. Event-based analysis of video. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:123, 2001.

[193] J. Zhang, M. Marszałek, M. Lazebnik, and C. Schmid. Local features and kernel for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73:213–238, 2007.

[194] Z.M. Zhang, Y.Q. Hu, S. Chan, and L.T. Chia. Motion context: A new representation for human action recognition. In *Proc. European Conference on Computer Vision*, pages IV: 817–829, 2008.

[195] X. Zhou, K. Yu, T. Zhang, and T.S. Huang. Image classification using super-vector coding of local image descriptors. In *Proc. European Conference on Computer Vision*, pages V: 141–154, 2010.

[196] A. Zisserman and A. Vedaldi. Efficient additive kernels via explicit feature maps. In *Proc. Computer Vision and Pattern Recognition*, pages 3539–3546, 2010.

List of Figures

List of Tables

Appendix A

Classification

In statistical terms, classification is typically comprised of the following two main steps: (a) classifier training, and (b) classification. In the first step, the parameters of the model are learned resulting in a classification function, or a classifier. In the second step, the trained classifier is employed to assign labels to previously unseen samples. The training samples are typically labeled, i.e., their true assignment to classes is known in advance. Such an approach is referred to as *supervised learning*. Let $x_i \in \Re^N$ be a feature vector representation of an image or a video sequence. The training set can be represented as $\{(x_i, y_i)\}_{i=1}^n$, where $y_i \in \{\omega_1, \ldots, \omega_l\}$ determines the assignment of the vector x_i to one or several classes l. The classification problem can then be formulated as finding the value of the function $f : \Re^N \to \Re$, given the test feature vector $x \in \Re^N$ as an input argument. The value of the function $y \in \Re$ determines the membership of the test feature vector x to one of the classes $\omega_1, \ldots, \omega_l$, i.e., if $y = i$, then the test feature vector belongs to the class ω_i.

A wide variety of classification techniques is available, which can be roughly categorized into discriminative models, probabilistic models, and combination of both. Given a set of classes $\{\omega_1, \ldots, \omega_l\}$, a discriminative model estimates one or more discriminative functions separating the different classes, whereas, a probabilistic model estimates a probability function for each class. Here, we focus on discriminative models, particularly the Support Vector Machines (SVM) classifier.

SVM, introduced by Vapnik *et al.* [14], is a modern and very powerful learning technique. Since its inception, the technique has been very successfully employed in a wide variety of applications, e.g., bioinformatics, computer vision, text categorization, financial analysis, etc. Suppose the case of two-class problem, wherein the training set can be represented

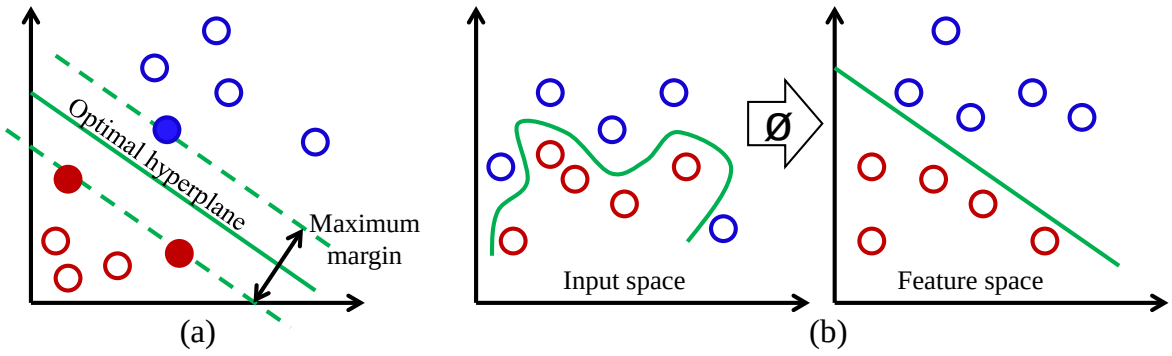


Figure A.1: **Illustration of SVM classification**. (a) Optimal hyperplane linearly separating sample points from two classes; (b) projection of non-linear input space $\mathcal{X}$ to a high dimensional feature space $\mathcal{H}$ using the non-linear function ϕ, allows the linear separability of sample points in that space.

as $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \Re^N$ is a feature vector, and $y_i \in \{-1, +1\}$ determines the membership of the feature vector to one of the two classes. Every feature vector can be considered as a point in $N-$dimensional *feature space*. Thus, the aim in SVM classification is to find a *discriminant function* $f : \Re^N \to \Re$, that distinguishes between points belonging to the different classes in the feature space. If $f(x) > 0$, then the point x is classified to the class $+1$, and if $f(x) < 0$, it is classified to the class -1. The linear discriminant function, then, is given by $f(x) = w^T x + b$, where w is the *weight vector* and b is the *bias*. The function divides the feature space into two half-spaces by a *hyperplane* given by $f(x) = w^T x + b = 0$. This is called a *linear SVM classifier*. Moreover, linear SVM classifier is often referred to as the *maximum-margin classifier*. This is due to the fact that in the linearly separable case, SVM generate a *separating hyperplane* by maximizing the distance to the samples from both classes. The distance is referred to as the *margin*, whereas, sample points which lie on the margin are called the *support vectors* (see Figure A.1(a)).

We can formulate the following optimization problem [63]: Given the labeled training instances $\{(x_i, y_i)\}_{i=1}^{n}$, find the optimal values of the wight vector w_o and the bias b_o such that they satisfy the constraint:

$$y_i \left(w_o^T x_i + b_o\right) \geq 1 \quad \text{for} \quad i = 1, 2, ..., n. \tag{A.1}$$

and the weight vector w_o minimizes:

$$\phi(w) = \frac{1}{2} \|w\|^2 = \frac{1}{2} w^T w \tag{A.2}$$

This is a constrained optimization problem called the *primal problem*. It may be solved by constructing the Lagrangian function [146]:

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i \left\{ y_i \left(w^T x_i + b \right) - 1 \right\}, \tag{A.3}$$

where α_i are called *Lagrange multipliers*. The solution of the optimization problem corresponds to the saddle point of the Lagrangian function, that has to be minimized with respect to w and b and maximized with respect to α_i. In this way, the following conditions can be defined:

$$\frac{\partial J(w, b, \alpha)}{\partial w} = 0, \tag{A.4}$$

and

$$\frac{\partial J(w, b, \alpha)}{\partial b} = 0. \tag{A.5}$$

Differentiating the Lagrangian function yields:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i, \tag{A.6}$$

and

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{A.7}$$

According to the *Karush-Kuhn-Tucker theorem* [63, 146], the following equation is satisfied at the saddle point of the Lagrangian function:

$$\alpha_{i,o} \left\{ y_i \left(w_o^T x_i + b_o \right) - 1 \right\} = 0 \quad \text{for} \quad i = 1, 2, ..., n. \tag{A.8}$$

It is therefore concluded that $\alpha_{i,o} \neq 0$ only for those feature vectors x_i for which $y_i \left(w_o^T x_i + b_o \right) = 1$, that is for the support vectors.

The primal optimization problem can be transformed into the *dual problem*. This can be done by substituting Eq. A.6 and Eq. A.7 into the Lagrangian function (Eq. A.3). The resulting equation becomes:

$$Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j. \tag{A.9}$$

Now dual optimization problem can be formulated similarly to that of the primal problem [63]: given the labeled training samples $\{(x_i, y_i)\}_{i=1}^{n}$, find the Lagrange multipliers $\{\alpha_{i,o}\}_{i=1}^{n}$, that maximize the objective function (Eq. A.9) provided that the following constraints are met:

1. $\sum_{i=1}^{n} \alpha_{i,o} y_i = 0$
2. $\alpha_{i,o} \geq 0 \quad$ for $\; i = 1, 2, ..., n.$

The Lagrange multipliers determined as a result of the optimization process can be used to compute the optimal weight vector:

$$w_o = \sum_{i=1}^{n} \alpha_{i,o} y_i x_i. \tag{A.10}$$

And the optimal bias can be computed using any support vector x_s according to Eq. A.1, i.e.,

$$\begin{aligned} y_s \left(w_o^T x_s + b_o \right) &= 1 \\ b_o = 1 - w_o^T x_s \quad &\text{for} \;\; y_s = 1. \end{aligned} \tag{A.11}$$

Finally, the optimal parameters w_o and b_o can be used to formulate the discriminant function that defines the optimal separating hyperplane. Since the term $\alpha_{i,o}$ equals to 0 for non-support vectors, the discriminant function can then be expressed only in terms of the support vectors, i.e.,

$$f(x) = \sum_{i=1}^{m} \alpha_{i,o} y_i x_i^T x + b_o, \tag{A.12}$$

where $x_1, x_2, \ldots, x_m$ are the support vectors and α_i are the corresponding Lagrange multipliers. It is important to note that the only operation that is performed on the feature vectors, in the computation of the discriminant function, is the inner product $x_i \; . \; x$. Furthermore, it is interesting to know that the internal model of the classifier is represented in the form of a subset of the training samples, the corresponding Lagrange multipliers, and the bias.

The initially proposed linear SVM classifier (Eq. A.12) is very efficient as well as effective in applications where the data is linearly separable in the feature space. However, many complex real-world applications (e.g., image categorization) require more expressive

hypothesis spaces than linear functions. Therefore, Boser *et al.* [14] propose a way to construct non-linear classifiers by applying the *kernel trick* to maximum-margin hyperplanes. Non-linear SVM projects the input space $\mathcal{X} \subseteq \Re^N$ into a high dimensional feature space $\mathcal{H}$ via $x \mapsto \phi(x)$, where the data is linearly separable, and a linear classifier can be used in that space (see Figure A.1(b)). This can be done efficiently by exploiting the fact that it is possible to compute the inner product of the feature vectors in the feature space using the so-called *kernel functions* (or kernels), without explicitly determining the high dimensional representations of the feature vectors. This concept is referred to as *kernel-induced feature space*. Now the discriminant function of the SVM classifier (Eq. A.12) can be defined in terms of the feature space $\mathcal{H}$ as:

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i \phi(x_i)^T \phi(x) + b \tag{A.13}$$

Let $K(x_i, x) = \phi(x_i)^T . \phi(x) \quad \forall x_i, x \in \mathcal{X}$ be the kernel function, which allows to compute the dot product in the high dimensional feature space without being explicitly mapped into it. Then, a non-linear representation of SVM can be obtained by replacing the dot product $x_i . x$ of Eq. A.13 with $K(x_i, x)$:

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b \tag{A.14}$$

The kernel function $K(x_i, x)$ can be seen as a similarity measure between the feature vectors x_i and x. Moreover, it is important to note that only those kernel functions which satisfy *Mercer's theorem* [26], can induce feature spaces of inner products. According to Mercer's theorem, $K(x_i, x)$ should be a *positive-definite* and *symmetric* matrix. This class of kernels is known as *Mercer kernels*. Nevertheless, it is important to mention here that Mercer kernels are not the only kernels which can be used with SVM. Many practical applications demonstrate that it is still possible to use kernels that do not obey the Mercer law. In that case, it is not guaranteed that there exists a feature space in which the kernel function is an inner product. However, the classifiers that employ such kernels may still perform very well. Several specialized kernels have been proposed in the literature for classifying various kinds of data (e.g., [10, 193, 24]). We can broadly classify these kernels into the Mercer and non-Mercer kernels. Some commonly used kernel functions include:

- Polynomial kernel

$$K(x,y) = \left(x^T y + p\right)^d \tag{A.15}$$

A special case $K(x,y) = x^T y$ is referred to as the *linear kernel*.

- Gaussian kernel (RBF)

$$K(x,y) = \exp\left\{-\gamma \sum_i \|x_i^a - y_i^a\|^b\right\}, \quad a \in \Re^+, b \in [0,2] \tag{A.16}$$

- Sigmoid kernel

$$K(x,y) = \tanh\left(\kappa x^T y + \theta\right) \tag{A.17}$$

The Mercer's theorem for sigmoid kernel is satisfied only for some values of κ and θ.

- Intersection kernel

$$K(x,y) = \sum_{i=1}^{N} \min(x_i, y_i) \tag{A.18}$$

- Chi-square kernel

$$K(x,y) = \exp\left\{-\gamma \chi^2(x,y)\right\}, \quad \text{where} \quad \chi^2 = \sum_{i=1}^{N} \frac{(x_i - y_i)^2}{x_i + y_i} \tag{A.19}$$

All the above mentioned kernels are proved to be Mercer kernels [125, 10].

Initially, SVM were designed to handle only binary classification problems, i.e., two-class problems. However, many practical applications include more than two classes. This limitation inspires extending the binary SVM to multi-class SVM. One common way to achieve multi-class SVM is the *one-against-all* approach (e.g., [15]), which is based on *winner-takes-all* strategy. In this approach, an SVM classifier is constructed for each of the l classes. The ith SVM is trained on all the instances of the ith class being positive, whereas, all the remaining $l-1$ class instances being negative. During the test phase, the test instance x is scored by all the l SVM classifiers, and the final decision is made on the basis of the values of the l discriminant functions:

$$l = \arg\max_{i=1,\ldots,l} f_i(x) \tag{A.20}$$

As described earlier, kernels are essentially related to similarity (or distance) measures. Such information is actually available in many data analysis problems. Therefore, what makes kernels to be a choice in most of the cases is the fact that the learning algorithms and theory can largely be decoupled from the specifics of the application area, which must simply be encoded into the design of an appropriate kernel function. For instance, working with kernels avoids the need to explicitly work with Euclidean coordinates. This is particularly useful for data sets involving strings, trees, micro arrays, text, etc. Nonetheless, using a single kernel may not be enough to solve accurately the problem under consideration. This happens, for instance, when dealing with image classification problems, where results may vary depending on the similarity measure chosen (e.g., color, shape, texture, etc.). As a result, information provided by a single similarity measure (kernel) may not be enough for classification purposes, and the combination of kernels appears as an interesting alternative to the choice of the 'best' kernel. A natural approach is to consider linear combinations of kernels [86]. In this thesis, we follow [193] and use the *multi-channel kernel*; wherein each channel c corresponds to a kernel, obtained using a specific similarity measure:

$$K\left(x,y\right)=exp\left(-\sum_{c}\frac{1}{\Omega_{c}}D\left(x^{c},y^{c}\right)\right)\quad, \tag{A.21}$$

where $D(x^c, y^c)$ is the distance computed using channel c , and Ω_c is the normalization factor computed as an average channel distance [193].

As discussed earlier, linear kernels are very efficient to train [71]. On the other side, non-linear kernels tend to yield better classification accuracy [193], but are computationally expensive. A class of kernels that are almost as efficient as the linear ones but usually much more accurate are the *additive homogeneous kernels* [196, 108]. These can be represented as: $K\left(x,y\right)=\sum_{i=1}^{N}k\left(x_{i},y_{i}\right)$, where k is itself a kernel function on the non-negative reals. Examples of k include the Hellinger's (Bhattacharya's) kernel $k\left(x,y\right)=\sqrt{xy}$ and the χ^2 kernel $k\left(x,y\right)=2xy/\left(x+y\right)$. While these kernels are usually defined for non-negative feature vectors (e.g., histograms), one can extend them to arbitrary vectors by setting $k^{'}\left(x,y\right)=\text{sign}\left(xy\right)k\left(\left|x\right|,\left|y\right|\right)$. Moreover, the computational advantage of using additive kernels is that they can be represented as linear kernels, with the computation of an efficient *feature map*. For instance, in case of the Hellinger's kernel, it suffices to consider the feature map defined by $\left[\Psi\left(x\right)\right]_{i}=\sqrt{x_{i}}$, as in fact $K\left(x,y\right)=\Psi\left(x\right)\ .\ \Psi\left(y\right)=\sum_{i=1}^{N}\sqrt{x_{i}}\sqrt{y_{i}}=\sum_{i=1}^{N}\sqrt{x_{i}y_{i}}$. For the χ^2 and other kernels, one can use the approximated feature maps introduced in [24], which are nearly as efficient.

Appendix B

Signatures

VU:

Le Directeur de Thèse:

PÉREZ Patrick

VU:

Le Responsable de l'École Doctorale:

VU pour autorisation de soutenance:

Rennes, le

Le Président de l'Université de Rennes 1:

CATHELINEAU Guy

VU après soutenance pour autorisation de publication:

Le Président de Jury:

BOUTHEMY Patrick